Input/output logic with a consistency check - the case of permission

Maya Olszewski [1] Xavier Parent [2] Leendert van der Torre [3]

University of Luxembourg
Computer Science and Communiations
Esch-sur-Alzette, Luxembourg

Abstract

In Input/output (I/O) logic, one makes a distinction between three kinds of permission, called negative, positive static and positive dynamic permission. They have been studied semantically and axiomatically by Makinson and van der Torre in the particular case where the underlying I/O operation for obligation is one of the standard systems. In this paper, we investigate what happens when the underlying I/O operation is one of the constrained I/O operations recently introduced by Parent and van der Torre. Their distinctive feature is two-fold. First, they are not closed under logical consequence. Second they have a built-in consistency check, which filters out excess outputs and allows them to properly deal with contrary-to-duty reasoning. The main contribution of this paper is the characterization of the positive static permission with a set of rules, called subverse rules. Due to the fact that the studied logics are different from the original framework, although the proof of the characterization result is similar to the original one, it still includes novel arguments. This is the definition of a first positive permission proof system for constrained output.

Keywords: Deontic logic, input/output logic, permission, normative system.

1 Introduction

The aim of this paper is to analyse three kinds of permission operations, derived from the Input/output (I/O) logics O_1 and O_3, introduced by Parent and van der Torre [17]. The analysis looks at negative permission, positive static permission and positive dynamic permission, such as Makinson and van der Torre have done in 2003 for the unconstrained I/O logics $out_1 - out_4$ [13]. [4] There are two main differences between the older and the newer systems. The newer systems are augmented with a consistency check. It is this property

[1] maya.olszewski.001@student.uni.lu

[2] x.parent.xavier@gmail.com

[3] leon.vandertorre@uni.lu

[4] Hansson [10] provides an enlightening overview of the major issues in deontic logic that are specific for permission.

which puts these logics at the level of constrained output, which can deal with contrary-to-duty (CTD) reasoning. They also lack the weakening of the output (WO) rule. (WO allows to infer (a, y) from (a, x) and $x \vdash y$, where (a, x) represents the norm that if a, then x ought to be the case, and $x \vdash y$ means that y logically follows from x). We analyse the differences that these changes cause to the different kinds of permissions and try to get rule-sets that fully characterize the permission operations. This leads us to introduce the first proof systems for positive permission in terms of constrained output.

With permission being far less studied than obligation, we see it as important to give it its fair share of spotlight. In practice, normative codes such as traffic rules often include both obligatory and permissive norms, and so it is vital when modeling such rules to have a good understanding of the choice of permission at hand as well as of the underlying (input/output) logic. As they lack WO, we argue that O_1 and O_3 can be better suited for modelling normative reasoning compared to the *out* logics. We briefly recall below the argument given in [14,15] against WO.

WO yields as a special case the principle of conjunction elimination, warranting the move from $(a, x \wedge y)$ to (a, x). As suggested for example by Hamblin [7], Goble [6] and Hansen [8, p. 91], such a principle is counter-intuitive in those cases where x and y are not separable, so that (to quote Hansen) "failing a part [of the order] means that satisfying the remainder no longer makes sense. E.g. if I am to satisfy the imperative 'buy apples and walnuts', and the walnuts [...] and the apples [are meant to] land in a Waldorf salad, then it might be unwanted and a waste of money to buy the walnuts if I cannot get the apples" [8, p. 91].

WO is also undesirable with respect to the issue of deontic detachment. Deontic detachment (DD) is the law : from $(\top, x)$ and (x, y) infer $(\top, y)$, where $\top$ denotes a tautology. It is a special case of the law known as cumulative transitivity (CT): from (a, x) and $(a \wedge x, y)$ infer (a, y). Counter-examples have been given to deontic detachment (see, e.g. [11,3]). They can be blocked by replacing CT with the following variant rule− we call it "aggregative cumulative transitivity" (ACT): from (a, x) and $(a \wedge x, y)$ infer $(a, x \wedge y)$. This substitute rule makes sense only in a system without WO. Here is an example. The Luxembourgish traffic laws [1] say that if one wants to park one's car at a parking spot having a park meter during the times specified on the street sign, then one should buy a ticket. They also say that, if a parking ticket is purchased, then it should be put on display inside the vehicle. The obligation to put the ticket on display no longer holds, if the obligation to pay is violated (for instance the ticket has been forged). Thus, the correct conclusion is: one should pay-and-display the ticket.

We believe that the permission operations defined in terms of O_1 and O_3 are worth studying as well. Similarly to the logics O_1 and O_3, the permission operations underlying those new logics also lack the WO rule. They also have a consistency proviso restraining the application of two rules, one of them being AND and the other being ACT. This is needed to block the well-known pragmatic oddity from [18] among other things.

Regarding WO, the same situation can be expected to arise with "may" as with "must". And indeed it does. For illustration's sake, consider the following example. Restaurants often have a lunch-menu (l), and typically they have the option to order a starter (s), a main course (m) and a dessert (d), a starter and a main course or a main course and a desert. However it is not generally allowed to order a starter and a desert, without there being a formal prohibition, but there is a lack of a positive permission. Let $(a, x)^p$ denote the conditional permission to do x given a. We have $(l, s \wedge m \wedge d)^p$ but not $(l, s \wedge d)^p$. As a second case, consider a modified version of Feldman's medication example [5, p. 87]. Let a and b be two medicines such that medicine a needs medicine b in order to be safe for use in the treatment of disease d. In that case we have that $(d, a \wedge b)^p$ but not $(d, a)^p$.

There is another class of I/O logics to compare to O_1 and O_3, namely constrained I/O logics [12]. They are better suited for normative reasoning than unconstrained I/O logics, as they are capable of handling CTD reasoning. We could, in principle, define the three kinds of permission using constrained I/O logic as the underlying logic for obligation, similarly as they are defined for unconstrained I/O logic. The main downside to this approach is that (to our knowledge) there is no axiomatic characterization of constrained I/O logic that is "intrinsic". Straßer *et al.* [22] provide a dynamic proof theory of constrained I/O logics−it is that of the adaptive logic (AL) framework (see e.g. [21] for a general introduction). First, unconstrained and constrained I/O logics are embedded within some suitable modal logics. Next, the adaptive counterparts of all the constrained I/O operations are given. Representation results are provided for the modal characterizations in both the unconstrained setting and the constrained setting. It would be interesting to investigate the relationship between their account and ours. We leave this issue as a topic for future research. One would need to go beyond their framework in its current form, which does not cover the new I/O logics from [14,16,17] yet, and has no apparatus for handling positive permissions.

As mentioned above, there are important differences between the classical I/O logics and the operations O_1 and O_3. Because of these differences, the proofs given by Makinson and van der Torre [13] do not always go through. The formal challenge thus consists in finding alternative proofs to the ones Makinson and van der Torre give, taking into consideration the nature of the new logics. We prove the characterization of the positive static permission operation by its subverse rule-set by showing that a result called the non-repetition property holds, such as Makinson and van der Torre did. However since intermediary results do not hold, the proof of the non-repetition property for O_1 is different. For O_3, the result is somewhat similar, as it also uses phasing of the derivation. In the present case the whole derivation cannot be phased, and one can phase only certain sub-parts of derivations, which is enough to prove the non-repetition property.

This paper is structured as follows. Section 2 gives the required background on I/O logic, section 3 outlines the differences between the classical and the new I/O logics, sections 4, 5 and 6 present respectively the negative permission, the static positive permission and the static dynamic permission. Finally, in section 7 we outline a few directions for future research.

2 Background

This section gives a brief review of the basic notions of I/O logic that are used throughout this work.

2.1 Semantics

I/O logic uses *conditional norms*, which are pairs of the form (a, x), where a is called the *body* of the norm and x the *head* of the norm. The norm (a, x) can be read as *if a, then x is obligatory.* For a set of norms G, $h(G)$ is the set of all heads of elements of G and $b(G)$ the set of all bodies of elements of G. $G(A)$ is defined as $\{x : (a, x) \in G\ for\ some\ a \in A\}$.

The four unconstrained output operations of I/O logic that have first been introduced are the following, where G is a set of norms, A a set of formulae of a propositional language, Cn the consequence operation of classical propositional logic and $\mathcal{L}$ the set of all boolean formulae:

Definition 2.1 (Classical unconstrained I/O operations [12])

- *Simple-minded output*: $out_1(G, A) = Cn(G(Cn(A)))$
- *Basic output*: $out_2(G, A) = \cap\{Cn(G(V)) : A \subseteq V, V\ complete\}$
  $= \cap\{out_1(V) : A \subseteq V, V\ complete\}$
  A set V is *complete* iff $V = \mathcal{L}$ or $V \subseteq \mathcal{L}$ is maximally consistent.
- *Reusable simple-minded output*:
  $$out_3(G, A) = \cap\{Cn(G(B)) : A \subseteq B = Cn(B) \supseteq G(B)\}$$
- *Reusable basic output*:
  $$out_4(G, A) = \cap\{Cn(G(V)) : A \subseteq V \supseteq G(V), V\ complete\}$$

Parent and van der Torre have introduced new logics O_1 and O_3 corresponding to out_1 and out_3 with an additional consistency check and without the rule WO [17]. They solve a problem that was present in the earlier systems: How to prevent the pragmatic oddity and the drowning problem? The pragmatic oddity [18] arises from the possibility of detaching a CTD obligation in a violation context, and aggregating it with its associated primary obligation. The following is a typical example: "you should keep your promise and apologize for not keeping it" can be derived from "you should keep your promise", "if you do not keep your promise you should apologize" and "you do not keep your promise" [16]. The drowning problem arises when a primary obligation no longer holds after a violation has occurred. [5]

[5] Other approaches are possible. It is often thought that the CTD scenarios involve two kinds of obligations, prima facie (ideal, etc) obligations vs. all-things-considered (actual, etc) obligations. [4,20] are two examples of a formal setting articulating such a distinction. Our take is different. We are interested in obligations which still hold even if violated, as opposed

Let $x \dashv\vdash y$ stand for $(x \vdash y)$ and $(y \vdash x)$. Then the systems Parent and van der Torre present are defined in the following way:

Definition 2.2 (New I/O logics [17])

- *Single-step detachment*: $x \in O_1(G, A)$ iff there exists some finite $M \subseteq G$ and a set $B \subseteq Cn(A)$ such that $M \neq \emptyset$, $B = b(M)$, $x \dashv\vdash \wedge h(M)$ and $\{x\} \cup B$ is consistent. $O_1(G) = \{(A, x) : x \in O_1(G, A)\}$.
- *Iterated detachment*: $x \in O_3(G, A)$ iff there exists some finite $M \subseteq G$ and a set $B \subseteq Cn(A)$ such that $M(B) \neq \emptyset$, $x \dashv\vdash \wedge h(M)$ and
  · $\forall B'(B \subseteq B' = Cn(B') \supseteq M(B') \Rightarrow b(M) \subseteq B')$
  · $\{x\} \cup B$ is consistent.
  $O_3(G) = \{(A, x) : x \in O_3(G, A)\}$.

M is called the *witness* of (A, x).

2.2 Proof Theory

Each of the previously defined output operations have their associated proof system, called $deriv_i$, for $i \in \{1, ..., 4\}$ for the classical I/O logics and D_i for $i \in \{1, 3\}$ for the new ones, each of which consists of the following sets of rules:

- $deriv_1 = \{$TAUT, SI, WO, AND$\}$
- $deriv_2 = \{$TAUT, SI, WO, AND, OR$\}$
- $deriv_3 = \{$TAUT, SI, WO, AND, CT$\}$
- $deriv_4 = \{$TAUT, SI, WO, AND, OR, CT$\}$
- $D_1 = \{$EQ, SI, R-AND$\}$
- $D_3 = \{$EQ, SI, R-ACT$\}$

Where the rule names have the following meaning:

- TAUT - tautology
- SI - strengthening of the input
- WO - weakening of the output
- AND - conjunction of the output
- OR - disjunction of the input
- CT - cumulative transitivity
- EQ - equivalence
- R-AND - restricted AND
- R-ACT - restricted aggregative cumulative transitivity

Those rules are the following:

$$\frac{-}{(\top, \top)}\ \text{TAUT} \qquad \frac{(a, x) \qquad b \vdash a}{(b, x)}\ \text{SI}$$

$$\frac{(a, x) \qquad x \vdash y}{(a, y)}\ \text{WO} \qquad \frac{(a, x) \qquad (a, y)}{(a, x \wedge y)}\ \text{AND}$$

$$\frac{(a, x) \qquad (b, x)}{(a \vee b, x)}\ \text{OR} \qquad \frac{(a, x) \qquad (a \wedge x, y)}{(a, y)}\ \text{CT}$$

---

to obligations satisfying ought-implies-can.

$$\frac{(a,x) \qquad (a,y) \qquad a\wedge x\wedge y \not\vdash \bot}{(a, x\wedge y)}\text{R-AND} \qquad \frac{(a,x) \qquad x \dashv\vdash y}{(a,y)}\text{EQ}$$

$$\frac{(a,x) \qquad (a\wedge x,y) \qquad a\wedge x\wedge y \not\vdash \bot}{(a, x\wedge y)}\text{R-ACT}$$

We say that $(a,x) \in deriv_i(G)$ (or $D_i(G)$) iff (a,x) is derivable from G using the rules of $deriv_i$ (or D_i). We say that $(A,x) \in deriv_i(G)$ (or $D_i(G)$) iff $(a,x) \in deriv_i(G)$ (or $D_i(G)$), where a is a conjunction of formulas in A. Equivalently, we say that $x \in deriv_i(G,A)$ (or $D_i(G,A)$).

Looking at the proof systems another difference between the classical and the new systems becomes apparent: the latter lack WO, whereas it is present in the former ones.

For simplifying derivation representations, let us define a generalized version of R-AND:

$$\frac{(a,x_1) \qquad \dots \qquad (a,x_n) \qquad a\wedge x_1\wedge \dots \wedge x_n \not\vdash \bot}{(a, x_1\wedge \dots \wedge x_n)}\text{G-R-AND}$$

which is a short version of n consecutive R-AND applications.

D_1 and D_3 are sound and complete w.r.t. the semantics [17], i.e. $(A,x) \in O_i(G)$ iff $(A,x) \in D_i(G)$ and so O_i and D_i can be interchanged when needed for $i \in \{1,3\}$.

We use the notation of O and D when we talk about the output operations with the consistency check O_1 and O_3, and *out* and *deriv* for the classical output operations out_1-out_4.

Parent *et al.* [14] define the notion of derivation as follows.

Definition 2.3 (Derivation)
Let D be a proof system. A *derivation* of (a,x) from a set of norms G is a finite sequence of pairs ending with (a,x), each of which is either an element of G or follows from earlier pairs in the sequence using the rules of D. The elements of G being used in a derivation are called the *leaves* of the derivation, and it is required that all leaves have a consistent fulfilment, i.e. for all leaves (a,x), $a\wedge x$ is consistent. The length of a derivation is the length of the sequence.

In this work we mostly represent derivations graphically using proof trees.

3 O versus *out*

Already at this point there is one significant difference when it comes to O versus *out*: whereas *out* is a closure operation [13], O is not, as it does not satisfy inclusion: take $G = \{(x,\neg x)\}$, $\neg x \notin O(G,x)$ so $G \not\subseteq O(G)$. However monotony ($G \subseteq H \Rightarrow O(G) \subseteq O(H)$) and idempotence ($O(O(G)) = O(G)$) both hold, as shown below. (Note that one half of idempotence is established for O_1 only.)

Proposition 3.1 *(Monotony)*
Let $O = O_1, O_3$ be an output operation, G, H be sets of norms with $G \subseteq H$ and let $(a,x) \in O(G)$. Then $(a,x) \in O(H)$.

Proof. Assume $(a,x) \in O(G)$. By the definitions of O_1 and O_3, there exists a witness M for (a,x), with $M \subseteq G$. As $G \subseteq H$, one can take the same M as witness to get that $(a,x) \in O(H)$. □

The following sequence of results leads to showing that the left-in-right direction of idempotence holds for O_1 :

Lemma 3.2 *Let $O = O_1$ be an output operation, G be a set of norms. Let M be the witness for (a,x). Then M does not contain a pair of the form (a_i, x_i) with $a_i \wedge x_i \vdash \bot$.*

Proof. Suppose M contains a pair of the form (a_i, x_i) with $a_i \wedge x_i \vdash \bot$. We know, by definition of O_1 that $x \dashv\vdash \wedge h(M)$, so $x \vdash x_i$, thus $a_i \wedge x \vdash \bot$. But $a_i \in b(M)$, so $b(M) \cup \{x\} \vdash \bot$ by monotony for $\vdash$, which contradicts the definition of the witness M. □

Lemma 3.3 *Let $O = O_1$ be an output operation, G be a set of norms. Let $(a,x) \in O(G)$ and M be the witness for (a,x). Then $M \subseteq O(G)$.*

Proof. Let $O = O_1$, $(a_i, x_i) \in M$. $\{(a_i, x_i)\}$ is finite and non-empty, $a_i \vdash a_i$, $x_i \vdash x_i$ and $\{x_i, a_i\} \nvdash \bot$ by Lemma 3.2. So $(a_i, x_i) \in O(G)$. □

Proposition 3.4 *(Idempotence, left-to-right)*
Let $O = O_1$ be an output operation, G be a set of norms. Then $(a,x) \in O(G) \Rightarrow (a,x) \in O(O(G))$.

Proof. Let $(a,x) \in O(G)$ and $M = \{(a_1,x_1), ..., (a_n,x_n)\}$ be the witness for (a,x). By Lemma 3.3, $M \subseteq O(G)$. So $(a,x) \in O(O(G))$. □

Proposition 3.5 *(Idempotence, right-to-left)*
Let $O = O_1, O_3$ be an output operation, G be a set of norms.
Then $(a,x) \in O(O(G)) \Rightarrow (a,x) \in O(G)$.

Proof. Take $(a,x) \in O(O(G))$. By completeness, there exists a derivation of (a,x) from $O(G)$ in the corresponding proof system D. We have that every leaf $(a_i, x_i) \in O(G)$. Let $\{(a_1,x_1), ..., (a_n,x_n)\} \subseteq O(G)$ be the enumeration of the leaves of that derivation. Then there also exists a derivation of (a_i, x_i) from G in the corresponding proof system D. Let $\{(a_{i_1}, x_{i_1}), ..., (a_{i_m}, x_{i_m})\} \subseteq G$ be the enumeration of the leaves of that derivation. We have that every leaf $(a_{i_j}, x_{i_j}) \in G$ for j such that $1 \leq j \leq m$. Putting those derivations together, we can get a derivation of (a,x) from G where the leaves are $\{(a_{1_1}, x_{1_1}), ..., (a_{n_m}, x_{n_m})\} \subseteq G$. By soundness, $(a,x) \in O(G)$. □

4 Negative Permission

Negative permission is the most straightforward permission of the three kinds we are going to discuss. Something is said to negatively permitted if it is not prohibited.

Definition 4.1 (Negative permission [13])
Let G be a set of norms and O an output operation. Then $(a,x) \in negperm(G)$ iff $(a, \neg x) \notin O(G)$.

We will now discuss if the results on negative permission from Makinson and van der Torre's work [13] still hold in this new setting. Let us first look at what Horn rules the negative permission operation satisfies. In Makinson and van der Torres's fashion let us call the premises of the rules of the form $(\alpha, \varphi) \in O(G)$ a *substantive premise* and the premises of the form $\theta \in Cn(\gamma)$ and $\bigwedge(\alpha \wedge \varphi) \nvdash \bot$ *auxiliary premise*. The idea behind the inverse of a Horn rule is the following: having one or more substantive premises, one takes one of them, negates its head and puts it as permitted in the conclusion. In retribution one takes the conclusion, negates its head and puts it as permitted in the premises. The other premises are left unchanged. Intuitively it says that if a group of conditional obligations imply some conclusion, which is also a conditional obligation, then taking all the premises in this group with the exception of one and combining it with the permission to not do the conclusion, then this implies that we also have the permission to not do what the excluded obligation stated (otherwise we would have the obligation of the conclusion). The updated Horn rules fit rules such as R-AND and R-ACT and their inverses. A Horn rule has the form:

$$\begin{aligned}\text{(HR): } &(\alpha_i, \varphi_i) \in O(G)\ (i \leq n)\ \&\ \theta_j \in Cn(\gamma_j)\ (j \leq m)\\ &\&\ \bigwedge_{k=0}^{n} (\alpha_k \wedge \varphi_k) \nvdash \bot\ \Rightarrow (\beta, \psi) \in O(G)\end{aligned}$$

Its inverse has the form:

$$\begin{aligned}\text{(HR)}^{-1}\text{: } &(\alpha_i, \varphi_i) \in O(G)\ (i < n)\ \&\ (\beta, \neg\psi) \in negperm(G)\\ &\&\ \theta_j \in Cn(\gamma_j)\ (j \leq m)\ \&\ \bigwedge_{k=0}^{n} (\alpha_k \wedge \varphi_k) \nvdash \bot\\ &\Rightarrow (\alpha_n, \neg\varphi_n) \in negperm(G)\end{aligned}$$

The inverses of each rule are given in Table 1.

Proposition 4.2 *Let $O = O_1, O_3$ be an output operation. If O satisfies a rule of the form* (HR)*, then the corresponding negperm operation satisfies the inverse(s)* $(\text{HR})^{-1}$.

Proof. The proof of EQ is trivial and the proof of SI is similar to the original paper by Makinson and van der Torre, so we omit them here.
Let G be a set of norms.

- Let O satisfy R-AND, $(a, x) \in O(G)$, $(a, \neg(x \wedge y)) \in negperm(G)$ and $a \wedge x \wedge y \nvdash \bot$. Then $(a, x \wedge y) \notin O(G)$ by definition of *negperm*. As $(a, x) \in O(G)$ and $a \wedge x \wedge y \nvdash \bot$, by R-AND for O we have $(a, y) \notin O(G)$. So $(a, \neg y) \in negperm(G)$.
- (i) Let O satisfy R-ACT, $(a, x) \in O(G)$, $(a, \neg(x \wedge y)) \in negperm(G)$ and $a \wedge x \wedge y \nvdash \bot$. Then $(a, x \wedge y) \notin O(G)$ by definition of *negperm*.

Table 1: An enumeration of all the rules satisfied by the proof systems corresponding to O_1 and O_3 as well as their inverse and subverse rules. Superscript o indicates an obligatory norm, superscript p a permissive one.

Rule	(HR)	$(\text{HR})^{-1}$	$(\text{HR})^{\downarrow}$
EQ	$\dfrac{(a,x) \quad x \dashv\vdash y}{(a,y)}$	$\dfrac{(a,x)^p \quad x \dashv\vdash y}{(a,y)^p}$	$\dfrac{(a,x)^p \quad x \dashv\vdash y}{(a,y)^p}$
SI	$\dfrac{(a,x) \quad b \vdash a}{(b,x)}$	$\dfrac{(a,x)^p \quad a \vdash b}{(b,x)^p}$	$\dfrac{(a,x)^p \quad b \vdash a}{(b,x)^p}$
R-AND	$\dfrac{(a,x) \quad (a,y) \quad a \wedge x \wedge y \nvdash \bot}{(a,x \wedge y)}$	$\dfrac{(a,x)^o \quad (a,\neg(x \wedge y))^p \quad a \wedge x \wedge y \nvdash \bot}{(a,\neg y)^p}$	$\dfrac{(a,x)^o \quad (a,y)^p \quad a \wedge x \wedge y \nvdash \bot}{(a,x \wedge y)^p}$
R-ACT	$\dfrac{(a,x) \quad (a \wedge x,y) \quad a \wedge x \wedge y \nvdash \bot}{(a,x \wedge y)}$	$\dfrac{(a,x)^o \quad (a,\neg(x \wedge y))^p \quad a \wedge x \wedge y \nvdash \bot}{(a \wedge x,\neg y)^p}$ $\dfrac{(a \wedge x,y)^o \quad (a,\neg(x \wedge y))^p \quad a \wedge x \wedge y \nvdash \bot}{(a,\neg x)^p}$	$\dfrac{(a,x)^o \quad (a \wedge x,y))^p \quad a \wedge x \wedge y \nvdash \bot}{(a,x \wedge y)^p}$ $\dfrac{(a \wedge x,y)^o \quad (a,x)^p \quad a \wedge x \wedge y \nvdash \bot}{(a,x \wedge y)^p}$

As $(a, x) \in O(G)$ and $a \wedge x \wedge y \nvdash \perp$, by R-ACT for O we have that $(a \wedge x, y) \notin O(G)$, so $(a \wedge x, \neg y) \in negperm(G)$.

(ii) Let O satisfy R-ACT, $(a \wedge x, y) \in O(G)$, $(a, \neg(x \wedge y)) \in negperm(G)$ and $a \wedge x \wedge y \nvdash \perp$. Then $(a, x \wedge y) \notin O(G)$ by definition of *negperm*. As $(a \wedge x, y) \in O(G)$ and $a \wedge x \wedge y \nvdash \perp$, by R-ACT for O we have that $(a, x) \notin O(G)$, so $(a, \neg x) \in negperm(G)$.

□

5 Static Positive Permission

The static positive permission takes into account two explicit sets of norms. A set G of explicit obligations and a set P of explicit permissions. Something is said to be statically permitted if one can get it as output from the obligation set together with a single permission.

Definition 5.1 (Static positive permission [13])
Let G be a set of explicit obligations and P a set of explicit permissions and O an output operation. Then $(a, x) \in statperm(P, G)$ iff $(a, x) \in O(G \cup Q)$ for some $Q = \{(c, z)\} \subseteq P$ or $Q = \emptyset$.

For static permission, the definition yields that $O(G) \subseteq statperm(P, G)$ as O is monotone. What is different with O than with *out* is that *statperm* is no longer a closure operation in its argument P as inclusion does not hold: take $P = \{(x, \neg x)\}, G = \emptyset$. Then $(x, \neg x) \in P$ but $(x, \neg x) \notin statperm(P, G)$, so $P \not\subseteq statperm(P, G)$. However, monotony holds ($P \subseteq Q$ implies $statperm(P, G) \subseteq statperm(Q, G)$) as O is monotonous and idempotence ($statperm(P, G) = statperm(statperm(P, G), G)$) also holds.

Proposition 5.2 *(Idempotence)*
Let $O = O_1, O_3$ be an output operation.
Then $statperm(P, G) = statperm(statperm(P, G), G)$.

Proof. To show the inclusion from right to left, one can take the same approach as for Proposition 3.4, using proof theory.

For the other way, assume $(a, x) \in statperm(P, G)$, let M be the witness for (a, x), and $B = b(M)$. By definition, since $\{x\} \cup B$ is consistent, M is also a witness for (B, x) and so $(B, x) \in statperm(P, G)$. Now one can take $M' = \{(B, x)\}$ to be the witness for (a, x) in $statperm(statperm(P, G), G)$, and thus $(a, x) \in statperm(statperm(P, G), G)$. □

statperm also is not a closure operation in its argument G, as inclusion does not hold: take $G = \{(x, \neg x)\}, P = \emptyset$. Then $(x, \neg x) \in G$ but $(x, \neg x) \notin statperm(P, G)$, so $G \not\subseteq statperm(P, G)$. Monotony holds as O is monotonous, but here idempotence fails: take $G = \emptyset, P = \{(a, x), (a, y)\}$ such that $a \wedge x \wedge y \nvdash \perp$. Then $(a, x \wedge y) \notin statperm(P, G) = O(\{(a, x)\}) \cup O(\{(b, y)\})$ but $(a, x \wedge y) \in statperm(P, statperm(P, G)) = O(\{(a, x), (a, y)\})$.

Let us define the subverse of Horn rules, which are the rules satisfied by *statperm*. Here, one of the substantive premises as well as the conclusion of the Horn rule are changed from being an obligatory norm to being a permissive norm. This simply says that if we have a set of obligations that imply another obligation, then having the same set of obligation with the exception of one premise, which now is a permission, will change the conclusion from an obligation into a permission:

$$\begin{aligned}(\mathrm{HR})^{\downarrow}\colon\ & (\alpha_i, \varphi_i) \in O(G)\ (i < n)\ \&\ (\alpha_n, \varphi_n) \in \mathit{statperm}(P, G) \\ & \&\ \theta_j \in Cn(\gamma_j)\ (j \leq m)\ \&\ \bigwedge_{k=0}^{n} (\alpha_k \wedge \varphi_k) \nvdash \bot \\ & \Rightarrow (\beta, \psi) \in \mathit{statperm}(P, G)\end{aligned}$$

The subverses of each Horn rule for O_1 and O_3 are given in Table 1. We will now prove a series of results leading up to the proof that the subverse set is sufficient to characterize the static permission operation *statperm*. The way to get there mimics the way Makinon and van der Torre took in 2003 [13].

Proposition 5.3 *Let O be O_1 or O_3. If O satisfies a rule of the form* (HR)*, then the corresponding statperm operation satisfies the subverse(s)* $(\mathrm{HR})^{\downarrow}$.

We omit the proof, as it is virtually the same as the original one [13].

Makinson and van der Torre have shown that for O the subverse set of a Horn rule is sufficient to characterize the corresponding static permission operation [13]. They have established that the problem reduces to showing that the non-repetition property holds. The *non-repetition property* is satisfied if for any $(b, y) \in O(G \cup \{(c, z)\})$ there exists a derivation of (b, y) from $G \cup \{(c, z)\}$ using the rules of the corresponding proof system, such that (c, z) is attached to at most one leaf node.

Proposition 5.4 *Consider O_1 and D_1. Let D be a derivation of (b, y) with a leaf-set L, in which some leaves are used more than once. Then there exists a derivation D' of (b, y) from a leaf-set $L' \subseteq L$ where every leaf is used at most once.*

The proof given by Makinson and van der Torre [13] in the original framework does not work in the new setting, because of the consistency proviso restraining the application of AND. We provide an alternative proof, which also would have worked for the original framework.

Proof. D is a derivation from L to (b, y), so by soundness and completeness, it holds that $y \in O_1(N, b)$ for N consisting of the norms present in the leaf-set L. By definition of O_1, $\exists M \subseteq N$ and $B \subseteq Cn(b)$ with $B = b(M)$ and $M \neq \emptyset$, $y \dashv\vdash \wedge h(M)$ and $\{y\} \cup B \nvdash \bot$.

Let $\{(a_1, x_1), ..., (a_n, x_n)\} = M$.
Then $a_1 \wedge ... \wedge a_n \wedge x_1 \wedge ... \wedge x_n \dashv\vdash \bigwedge B \wedge y \nvdash \bot$.

As $B \subseteq Cn(b)$, $b \vdash a_1 \wedge ... \wedge a_n = \bigwedge B$. We can thus build the following derivation. For visual effect we omit the auxiliary premises in the proof tree.

$$\dfrac{\dfrac{\dfrac{\dfrac{(a_1,x_1)}{(a_1\wedge ...\wedge a_n,x_1)}\text{SI} \quad ... \quad \dfrac{(a_n,x_n)}{(a_1\wedge ...\wedge a_n,x_n)}\text{SI}}{(a_1\wedge ...\wedge a_n, x_1\wedge ...\wedge x_n)}\text{G-R-AND}}{(b,x_1,...x_n)}\text{SI}}{(b,y)}\text{EQ}$$

Put $L' = M \subseteq N$. This derivation uses all elements of L' only once, so all norms of the initial leaf-set L are used at most once. □

Corollary 5.5 *O_1 satisfies the non-repetition property.*

Corollary 5.6 *The subverse set of EQ, SI, R-AND suffices to characterize the static permission operation based on O_1.*

Let us look at O_3 now. The following result has been adapted from the original framework [13] to fit O_3. This proof is inspired by the work of Makinson and van der Torre. Similarly to their proof, we are phasing the derivation a certain way. The difference is that, given the nature of the output operations we are considering, we are not able to phase the full derivation, and restrict ourselves to certain sub-derivations.

Lemma 5.7 *Let D be a derivation using the rules EQ, SI, R-ACT. Then at any line $l : (a,x)$ of the derivation:*

- *the head of l, x, classically implies the head of any line above it in D that l is based on*
- *the conjunction of body and head of l, $a \wedge x$, both classically implies the body and the head of any line above it in D that l is based on*

The proof is a straightforward proof by induction on the length of the derivation, and is omitted here.

Lemma 5.8 *Let $D = \{l_1, ..., l_n\}$ be a derivation of (b,y) from leaf-set L using the rules EQ, SI, R-ACT, and let l_i be a line where R-ACT is applied. Then there exists a derivation D' of (b,y) from leaf-set L, which is alike D, except for the fact that the two sub-derivation above line l_i follow the order SI, R-ACT, EQ.*

Proof. Consider the derivation D.

$$\cdots\ (a_i,x_i)\ \cdots \qquad \cdots\ (a_j,x_j)\ \cdots$$

$$\dfrac{(a,x') \qquad (a\wedge x', x'') \qquad a\wedge x'\wedge x'' \nvdash \bot}{l_i : (a, x'\wedge x'')}\text{R-ACT}$$

$$\cdots \qquad\qquad \cdots$$

$$(b,y)$$

Let l_i be a line of the conclusion of an R-ACT rule. Let d_1 be the left sub-derivation, and d_2 the right sub-derivation, with (a, x') and $(a \wedge x', x'')$ as their respective roots and $L(d_1),\ L(d_2)$ as leaf-sets. The rule EQ is invertible both with SI and R-ACT, it can be applied at any point in the derivation. Without loss of generality assume that EQ is applied at the bottom of the derivations d_1 and d_2. This leaves rules SI and R-ACT above in the upper parts of d_1 and d_2. By Lemma 5.7, it holds that $a \wedge x' \wedge x'' \vdash a_k$ and $a \wedge x' \wedge x'' \vdash x_k$ for every norm (a_k, x_k) from which $(a, x' \wedge x'')$ follows in the derivation D, so for all the lines in d_1 and d_2. This gives that in d_1 and d_2, R-ACT followed by SI can be inverted to SI followed by R-ACT; the following derivation

$$\dfrac{\dfrac{(b, y_1) \qquad (b \wedge y_1, y_2) \qquad b \wedge y_1 \wedge y_2 \nvdash \bot}{(b, y_1 \wedge y_2)}\,\text{R-ACT} \qquad c \vdash b}{(c, y_1 \wedge y_2)}\,\text{SI}$$

can be transformed into:

$$\dfrac{\dfrac{(b, y_1) \qquad c \vdash b}{(c, y_1)}\,\text{SI} \qquad \dfrac{(b \wedge y_1, y_2) \qquad c \wedge y_1 \vdash b \wedge y_1}{(c \wedge y_1, y_2)}\,\text{SI} \qquad c \wedge y_1 \wedge y_2 \nvdash \bot}{(c, y_1 \wedge y_2)}$$

The fact that $c \wedge y_1 \wedge y_2 \nvdash \bot$ follows from

- $a \wedge x' \wedge x'' \vdash c$
- $a \wedge x' \wedge x'' \vdash y_1$
- $a \wedge x' \wedge x'' \vdash y_2$
- $a \wedge x' \wedge x'' \nvdash \bot$

So the derivations d_1 and d_2 can be phased to SI, R-ACT, EQ. □

Theorem 5.9 *In a derivation using at most the rules EQ, SI, R-ACT and having two leaves (a, x) and root (b, y), one of those leaves can be eliminated.*

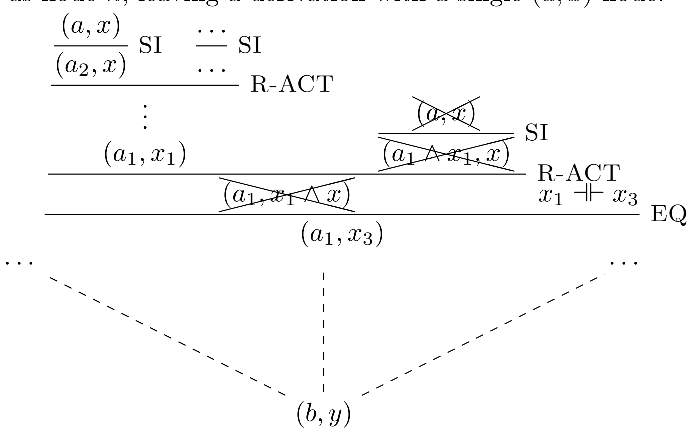
Proof. Looking at derivations having two leaves labelled with (a, x), we are in the following scenario, with the depicted R-ACT node n being the meeting point of two sub-derivations both containing (a, x):

$$\begin{array}{ccc} \cdots\ (a,x)\ \cdots & \cdots\ (a,x)\ \cdots & \\ \vdots & \vdots & \\ p : (a_1, x_1) & q : (a_1 \wedge x_1, x_2) & a_1 \wedge x_1 \wedge x_2 \nvdash \bot \end{array}$$
$$\dfrac{p : (a_1, x_1) \qquad q : (a_1 \wedge x_1, x_2) \qquad a_1 \wedge x_1 \wedge x_2 \nvdash \bot}{n : (a_1, x_1 \wedge x_2)}\,\text{R-ACT}$$
$$\cdots \qquad \vdots \qquad \cdots$$
$$(b, y)$$

By Lemma 5.8, we know that those two sub-derivations can be replaced by derivations where the order of the rules is SI, R-ACT, EQ. Call n the meeting node of the two sub-trees d_1 and d_2 containing (a, x), and such that the sub-tree with n as root is phased SI, R-ACT, EQ.

The rest of the proof is very similar to the proof Observation 3 (c) done by Makinson and van der Torre [13]. The rule R-ACT goes from (a, x), $(a \wedge x, y)$ with $a \wedge x \wedge y \nvdash \bot$ to $(a, x \wedge y)$. We call (a, x) the minor premise and $(a \wedge x, y)$ the major premise. The succession of R-ACT can be written in a way where no major premise of an application of R-ACT is the conclusion of another application of R-ACT. This has been shown for ACT [13] (from (a, x) and $(a \wedge x, y)$ to $(a, x \wedge y)$), and it still holds for its restricted version.

As node q is a major premise of R-ACT, it is not the conclusion of another R-ACT application, which means that the sub-tree d_2 has as only leaf (a, x) and root q and uses only one SI. The sub-tree d_1 which has p as root uses SI and R-ACT. By this and by Lemma 5.7, it holds that $x_2 \dashv\vdash x$ and $x_1 \vdash x$. So $x_1 \dashv\vdash x_1 \wedge x$, and one can delete from the tree the sub-tree d_2 with root q as well as node n, leaving a derivation with a single (a, x) node:



□

Corollary 5.10 *In a derivation using at most the rules EQ, SI, R-ACT and having multiple leaves (a, x) and root (b, y), all but one of those leaves can be eliminated.*

Corollary 5.11 *O_3 satisfies the non-repetition property.*

Corollary 5.12 *The subverse set of EQ, SI, R-ACT suffices to characterize the static permission operation based on O_3.*

6 Dynamic Positive Permission

Similarly to the static positive permission, the dynamic positive permission takes into account a set of obligations G and a set of permissions P. However, the static positive permission is not a straightforward result of an output operation. The main idea is that (a, x) is dynamically permitted if adding $(a, \neg x)$ to the set of obligations causes a conditional prohibition of something that is permitted under that same condition for the set G. This can be understood as a form of conflict resolution; one allows something, if allowing the opposite causes conflicts with the already existing permissions. The definition as found in the original framework [13] had to be adapted slightly, as there, exact opposites are used in order to detect conflicts ((a, x) and $(a, \neg x)$). In the old systems $out_1 - out_4$ this was not a problem, as WO was always present. The WO rule allowed to include any norm derivable from the exact opposite in this conflict resolution. In the new systems this can no longer be used as is, because of the lack of WO. Instead we have to work with collectively inconsistent pairs so that all those norms that are no longer derivable via WO are still considered explicitly.

Definition 6.1 (Dynamic positive permission)
Let G be a set of explicit obligations and P a set of explicit permissions and O an output operation. Then $(a, x) \in dynperm(P, G)$ iff $\exists c, u, v$ s.t. $(c, u) \in O(G \cup \{(a, \neg x)\})$ and a pair $(c, v) \in statperm(P, G)$ with $u \wedge v$ inconsistent and c consistent.

To get a better understanding of how the dynamic permission works and how it detects conflicts, let us look at the following example.

Example 6.2 Let f denote *eating with fingers*. c denote *clean* and e denote *eat*. Let us assume that it is always permitted to eat something, but that if something is not clean, then we should not eat it. So let G and P be such that $(\neg c, \neg e) \in O(G)$ and $(\top, e) \in statperm(P, G)$. Then it is dynamically permitted to not eat with fingers, because adding $(\top, f)$ to the obligation set would allow us to derive $(\neg c, f)$ which is in conflict with $(\neg c, \neg e)$:
$(\neg c, f) \in O(G \cup \{(\top, f)\})$ for $(\neg c, \neg e) \in statperm(P, G)$ with $f \wedge \neg e \vdash \bot$.
So $(\top, \neg f) \in dynperm(P, G)$.

Makinson and van der Torre give a general proof that if an output operation satisfies a Horn rule, then the dynamic permission operation satisfies its inverse. We cannot follow the same path as they did, as they use certain properties that do not hold, such as inclusion. We do not give a general proof, but we show that for the systems O_1 and O_3 specifically this holds.

Proposition 6.3 *Let $O = O_1, O_3$ be an output operation. If O satisfies a Horn rule of the form (HR), then the corresponding dynamic permission satisfies the inverse of the Horn rule (HR)$^{-1}$.*

Proof. This proof makes use of the proof theory, as D_1 and D_3 are sound and complete w.r.t. O_1 and O_3 respectively. The relevant Horn rules and their inverses are given in Table 1.

- EQ is straightforward. Details are omitted.
- Suppose O satisfies SI. Let P and G be such that $(a, x) \in dynperm(P, G)$ and $a \vdash b$. We have $(c, v) \in statperm(P, G)$ and $(c, u) \in O(G \cup \{(a, \neg x)\})$ for some u, v and c such that c consistent but $u \wedge v \vdash \bot$. By completeness, (c, u) is derivable from $G \cup \{(a, \neg x)\}$ in the corresponding proof system D. Given SI, (c, u) is derivable from $G \cup \{(b, \neg x)\}$. So by soundness $(c, u) \in O(G \cup \{(b, \neg x)\})$, and hence $(b, x) \in statperm(P, G)$. Hence *dynperm*

satisfies $(SI)^{-1}$.

- Suppose O satisfies R-AND. Take P and G such that $(a,x) \in O(G)$, $(a, \neg(x \wedge y)) \in dynperm(P,G)$ and $a \wedge x \wedge y \nvdash \bot$. We have $(c,v) \in statperm(P,G)$ and $(c,u) \in O(G \cup \{(a, x \wedge y)\})$ for u, v and c such that $u \wedge v \vdash \bot$ and c consistent. By completeness, (c,u) is derivable from $G \cup \{(a, x \wedge y)\}$ and (a,x) is derivable from G in the corresponding proof system D. D has R-AND and $a \wedge x \wedge y \nvdash \bot$. So one can combine the two derivations to obtain a derivation of (c,u) from $G \cup \{(a,y)\}$. By soundness $(c,u) \in O(G \cup \{(a,y)\})$. This implies $(a, \neg y) \in statperm(P,G)$, and shows that *dynperm* satisfies $(\text{R-AND})^{-1}$.
- Suppose O satisfies R-ACT.
  · Take P and G such that $(a,x) \in O(G)$, $(a, \neg(x \wedge y)) \in dynperm(P,G)$ and $a \wedge x \wedge y \nvdash \bot$. Then $(c,v) \in statperm(P,G)$ and $(c,u) \in O(G \cup \{(a, x \wedge y)\})$ for c, v and u such that c consistent and $u \wedge v \vdash \bot$. By completeness, (a,x) is derivable from G, and (c,u) is derivable from $G \cup \{(a, x \wedge y)\}$ in the corresponding proof system D. D has R-ACT as rule and $a \wedge x \wedge y \nvdash \bot$. One can combine the two derivations to obtain a derivation of (c,u) from $G \cup \{(a \wedge x, y)\}$. By soundness $(c,u) \in O(G \cup \{(a \wedge x, y)\})$. It follows that $(a \wedge x, \neg y) \in dynperm(P,G)$. Hence *dynperm* satisfies the first version of $(\text{R-ACT})^{-1}$.
  · Take P and G such that $(a \wedge x, y) \in O(G)$, $(a, \neg(x \wedge y)) \in dynperm(P,G)$ and $a \wedge x \wedge y \nvdash \bot$. Then $(c,v) \in statperm(P,G)$ and $(c,u) \in O(G \cup \{(a, x \wedge y)\}$ for such that there is some c, v and u such c consistent and $u \wedge v \vdash \bot$. By completeness, $(a \wedge x, y)$ is derivable from G and (c,u) is derivable from $G \cup \{(a, x \wedge y)\}$ in the corresponding proof system D. D has R-ACT and $a \wedge x \wedge y \nvdash \bot$. So one can combine the two derivations to obtain a derivation of (c,u) from $G \cup \{(a,x)\}$. By soundness, $(c,u) \in O(G \cup \{a,x\}), andhence(a, \neg x) \in dynperm(P,G)$. This shows that *dynperm* satisfies the second version of $(\text{R-ACT})^{-1}$.

□

7 Conclusion and Future Work

In this paper we introduce the first proof systems for permission in terms of constrained output. We use the two logics of constrained output with a consistency check. The proofs are generalizations of the proofs of Makinson and van der Torre for unconstrained output [13]. Only constrained output can handle CTD reasoning, so O_1/O_3 together with the permissive norms defined in this paper is the first approach satisfying the following minimal requirements:

- detachment semantics for obligation and permissive norms (negative, static, dynamic) which can reason about CTD and dilemmas in a consistent way
- proof systems for these semantics both for obligation and one kind of permission (static)

As topics for future research, we firstly would like to find out whether the inverse rule-set is enough to fully characterize the negative permission and the positive dynamic permission operations. Furthermore, it would be desirable to find general proofs, as some of the proofs we provided are tailored to O_1 and O_3 specifically. This would allow to include any future systems in the analysis.

There are several papers about permission as exception/derogation [2,9,19]. We leave it as a topic for future research to investigate if the account studied in this paper yield any new insight on this notion.

Finally, we only consider two operations O_1 and O_3, whereas there are four classical I/O operations out_1-out_4. Indeed, only two operations with a consistency check have been defined so far. The definition of O_2/O_4 such that they satisfy all the desired properties remains an open problem.

References

[1] *Recueil de Législation Routière*, `http://legilux.public.lu/eli/etat/leg/code/route/20190531`, accessed: 2019-06-12.

[2] Boella, G. and L. van der Torre, *Permissions and Obligations in Hierarchical Normative Systems*, in: *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, ICAIL '03 (2003), p. 109–118.

[3] Cariani, F., *Deontic logic and natural language*, in: D. Gabbay, J. Horty, X. Parent, R. van der Meyden and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, College Publications, London, 2020 Volume 2. To appear.

[4] Carmo, J. and A. J. I. Jones, *Deontic logic and contrary-to-duties*, in: D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume 8* (2002), pp. 265–343.

[5] Feldman, F., "Doing the Best We Can: An Essay in Informal Deontic Logic," D. Reidel Publishing Company, 1986.

[6] Goble, L., *A logic of good, should, and would: Part I*, Journal of Philosophical Logic **19** (1990), pp. 169–199.

[7] Hamblin, C., "Imperatives," Oxford, 1987.

[8] Hansen, J., "Imperatives and Deontic logic," Ph.D. thesis, University of Leipzig (2008).

[9] Hansen, J., *Reasoning about permission and obligation*, in: S. O. Hansson, editor, *David Makinson on Classical Methods for Non-Classical Problems*, Springer Netherlands, Dordrecht, 2014 pp. 287–333.

[10] Hansson, S. O., *The varieties of permission*, in: D. Gabbay, J. Horty, X. Parent, R. van der Meyden and L. van der Torre, editors, *Handbook of Deontic Logic and Normative Systems*, College Publications, London, 2013 pp. 195–240, volume 1.

[11] Lassiter, D., "Graded Modality," Oxford University Press, 2017.

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

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

[14] Parent, X., D. Gabbay and L. an der Torre, *Intuitionistic basis for input/output logic*, in: S. O. Hansson, editor, *David Makinson on Classical Methods for Non-Classical Problems*, Springer Netherlands, Dordrecht, 2014 pp. 263–286.

[15] Parent, X. and L. van der Torre, *I/O logics without weakening*, To appear in *Filosofiska Notiser*.

[16] Parent, X. and L. van der Torre, *The pragmatic oddity in norm-based deontic logics*, in: G. Governatori, editor, *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17 (2017), p. 169–178.

[17] Parent, X. and L. van der Torre, *I/O logics with a consistency check*, in: J. Broersen, C. Condoravdi, S. Nair and G. Pigozzi, editors, *Deontic Logic and Normative Systems* (2018), pp. 285–299.

[18] Prakken, H. and M. Sergot, *Dyadic deontic logic and contrary-to-duty obligations*, in: D. Nute, editor, *Defeasible Deontic Logic*, Kluwer Academic Publishers, Dordrecht, 1997 pp. 223–262.

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

[20] Strasser, C., *A deontic logic framework allowing for factual detachment*, Journal of Applied Logic **9** (2010), pp. 61–80.

[21] Straßer, C., "Adaptive Logics for Defeasible Reasoning. Applications in Argumentation, Normative Reasoning and Default Reasoning," Trends in logic **38**, Springer, Berlin, 2014.

[22] Straßer, C., M. Beirlaen and F. V. D. Putte, *Adaptive logic characterizations of input/output logic*, Studia Logica **104** (2016), pp. 869–916.