

# A Theory of Permission Based on the Notion of Derogation

Audun Stolpe

*Department of Informatics, Informatikkbygget, Gaustadallen 23, N-0371 Oslo, Norway*

---

## Abstract

This paper presents a unifying theory of permission that integrates the concept of negative permission with three concepts of positive permission, namely explicit permission, exemption and antithetic permission. The concepts are defined and logically related by paying particular attention to the system of which they form part. A simple procedure for calculating the permitted actions that can be said to be implicit in a code of norms or a policy specification is then given.

*Key words:* Permission, policy specification, information accountability.

---

## 1. Introduction

The concept, or rather concepts, of permission is a topic of much debate in the theory of normative systems and the philosophy of law.<sup>1</sup> It is also, for reasons I shall have more to say about shortly, potentially of crucial importance for information management in open distributed systems, where the need for principled ways of specifying allowable uses of information is increasing—rapidly. There are several problems related to permission, but from a logico-philosophical point of view they can all be lumped together into the question: What is the common denominator of all permitted actions, and how does it connect the many facets of the concept of permission?

It is generally agreed that there are two main categories of permission, although there is no consensus yet on how they are related. The first, usually called *negative permission*, is fairly simple: An action is negatively permitted

---

*Email address:* [audus@ifi.uio.no](mailto:audus@ifi.uio.no) ()

<sup>1</sup>See e.g. [4, 5, 6, 8, 9, 10, 16, 17, 24, 25, 26, 27, 31, 33, 36].

by a code if and only if it is not prohibited by that code. This is nonetheless a very important concept that figures prominently in law. In criminal law, it is known as the principle *nullum crimen sine lege*—there is no crime where there is no law—where it admits of a number of interpretations depending, among other things, on whether it is applied to the legislature or to the adjudicating authorities. In the latter case it expresses the presumption of the innocence of the defendant, that is, it coincides with the principle *in dubio pro reo* stating that whenever there is doubt, one should rule in favour of the accused. When the principle is applied to the legislature, on the other hand, it says that the existence of a crime depends on there being a previous legal provision declaring the action to be a penal offense.<sup>2</sup> In societies in which new regulations are continually enacted the general assumption that the non-existence of a prohibitive rule gives rise to a permission is an important one to defend [10]. The proliferation of novel technology, the emergence of multi-culturalism, professionalization of areas of expertise etc. all push new legislation into existence. Not to infringe on people’s freedom, therefore, it is important to acknowledge that where there is no law there is leeway, rather than, say, to rule by the spirit of law.

The concept of *positive permission*, on the other hand, is more elusive. As a first approximation one may say that something is positively permitted if and only if a code explicitly presents it as such [21, p. 391]. According to Norwegian law, for instance, if a temporary representation of a work that is ordinarily protected by intellectual property rights is essential to a process whose sole purpose is to facilitate the legitimate use of the work, then it is permitted to make copies of it. This is explicitly stated in the law, and is in that sense a *positive* permission. But this leaves a central logical question unanswered. As well as the actions that a code explicitly pronounces to be permitted, there are others that in some sense follow from the explicit ones. The problem is to clarify the inference from one to the other [ibid.].

In computer science the concept of permission has figured prominently in theoretical research for many years—although often under different names such as privilege or access right.<sup>3</sup> The need for a rigorous definition of the concept arose in connection with the problem of how to design systems that

---

<sup>2</sup>See [7] and [36] for a fuller discussion of this issues.

<sup>3</sup>See e.g. [22, 23, 30] and [32]

would prevent information from “escaping” beyond appropriate boundaries, usually by storing information in a set of files associated with an access policy formulated in terms of roles and privileges.<sup>4</sup>

Not disputing the obvious merits of this line of research, Weitzner et al. argue in a recent paper [35] that the access control paradigm is not very well equipped to handle the rapidly evolving Web-based information ecosystem as we now know it. Due to the proliferation of personal information on the Web and the increasing analytical power available to large institutions through Web search engines and other facilities, access control over a single instance of personal data, they argue, is insufficient to guarantee the protection of privacy when either the same information is publicly available elsewhere or it is possible to infer it with a high degree of accuracy from other information that is itself public [ibid., 84]. In response, Weitzner et al. propose to shift the emphasis from data protection onto *information accountability*, understood as the design of architectures that make the *use* of information transparent and traceable. Rather than to hide it from view—which is increasingly difficult anyway—we should aim to make it possible to determine whether a particular use of information is appropriate under a given set of rules, and hence to determine when individuals and institutions can be held accountable for *misuse*, so the argument goes. Information accountability, one might say, is a take on information management that, unlike the effort to ensure compliance through access control, is modelled on the actual relation between the law and its subjects [ibid. 86].

In the abstract at least, this idea makes good sense. As the information economy grows more complex and organic, the boundary between the ‘digital’ and the ‘analog’ world is being erased. There is in general no clean separation between an electronic market, say, and the rest of society. An action taken in a digital environment may have ramifications within the environment itself—usually in terms of the availability of information and services—or outside of it, in terms of judicial liabilities, contractual obligations and so forth. But, when the virtual and the real fuses, the protection of liberties becomes partly an algorithmic problem, and that forces us to think through the concepts involved more rigorously than we would otherwise have to.

Weitzner et al. list three architectural features that an accountable information infrastructure would need to have: It would need to have a *policy*

---

<sup>4</sup>See e.g. [1, 2, 12], and [28]

*aware transaction log* that records the information pertinent to the assessment of accountability. It would also need to have a *policy language* against which compliance is checked, and finally it would have to provide *policy-reasoning tools* to assist users in answering such questions as: Is this data allowed to be used for a given purpose? But clearly, if we want computers to answer such questions, then we need adequate methods for calculating the permissions that can reasonably be said to be implied by the policy. This problem cries out for a principled solution.

In what follows, the phrase “positive permission” will be used as a collective term covering both explicitly declared permission and anything that must be reckoned permitted by implication from what has thus been explicitly stated (by some as yet unspecified notion of implicature). I shall distinguish between two kinds of *implied* positive permission, namely *exemption* and *antithetic permission*. An action of the former kind may tentatively be characterised as one which is exempted from a covering prohibition by a permissive provision. Stated differently, an exemption is an action that (vagueness intended) falls under a provision that has been declared by law to constitute an exception to a general prohibition. Consider the following example from the Norwegian police act § 11 : ‘It is forbidden for participants in any public arrangement to wear a mask, *unless* participating in a play or masquerade or the like’ (my emphasis). We may infer that it is permitted to wear a mask during a public performance of, say, *The Tempest*, since *The Tempest* is indeed a play. In other words, if you are an actor in a public performance of *The Tempest* then you’re exempted from the prohibition against wearing a mask in public.

The second form of implied permission, may be called *antithetic permission*, since such a permission (the thesis) overrules any prohibition (the antithesis) that is incompatible with it. Unlike exemption, the primary function of antithetic permissions is not to limit or suspend an existing prohibition, but rather to prevent one from being passed or practiced in the first place. As a first attempt, we may say that an action is antithetically permitted if it cannot be prohibited by a code without making that code contradict an explicit or implicit permission which is already in force. The paradigm example—but by no means the only one—is an action protected by constitutional law. Freedom of expression, for instance, is recognized as a human right under Article 19 of the Universal Declaration of Human Rights and is recognized in international human rights law in the International Covenant

on Civil and Political Rights. Although it is not absolute—limitations may follow the "harm principle" or the "offense principle", for example in the case of public nudity or "hate speech"—it is a provision that guarantees certain rights to people, in the sense that it pre-empts any conceivable prohibition against those rights. An example that comes to mind is the Jyllands-Posten incident of 2005, when Muslim organizations filed a complaint with the Danish police, following the publication of twelve cartoons depicting the Islamic prophet Mohammad. The investigation was discontinued by the Regional Prosecutor in Viborg, who concluded that Jyllands-Posten must be reckoned protected by the freedom of expression. The Director of Public Prosecutors in Denmark later agreed. One may say, therefore, that the printing of the cartoons was deemed antithetically permitted by the Danish authorities.

Taking stock, positively permitted is anything that is either i) explicitly declared to be so, or ii) permitted by implication from something that falls under (i) and that may (exemption) or may not (antithetic permission) constitute an exception to an already existing general prohibition. Clearly, these concepts are quite closely related.

The aim of the present paper is to exhibit the logical interrelations between these variants of positive permission as well as the logical interrelations between positive and negative permission. I shall do so in a way that heeds the slogan "no logic of norms without attention to the system of which they form part", that is, permissive norms will be analysed in the larger context of a *system*. Finally, by way of illustration of the utility of the framework, I give a simple procedure for calculating the positive permission that can be said to be implicit in a code or policy specification.

## 2. An informal analysis of permissive structures

An explicitly granted permission would be completely idle were it not set out against a backdrop of prohibitions, because telling me what I am permitted to do does not in any way alter the range of choices open to me if nothing is denied me anyway. The point may be stated in terms of reasons: When some authority issues a directive, that authority *purports* to give its subjects a reason to act accordingly. The extent to which it actually does so, one might say, reflects the extent to which its claim to authority is accepted. Thus the statement "according to institution  $s$ ,  $b$  is mandatory given  $a$ ", can, if the authority of  $s$  is presumed unchallenged, be understood as " $s$  has given

you a reason to do  $b$  given  $a$ ". An explicit permission, on the other hand, is a declaration to the effect that you have *no* (institutional) reason *not* to do  $b$  given  $a$ , or stated differently; it is a legislative act whereby you are *released* from an obligation or (by pre-emption) shielded from the imposition of one. In other words, a permission has no positive regulatory content, meaning that it does not *require* that something be done. It only serves to inform you what you are *not* under a duty to do, which, of course, is entirely superfluous if there is no duty. In other words, permission is essentially a *negative* concept.

This is not a novel insight, of course, but has been stressed time and again by philosophers of law and logicians alike.<sup>5</sup> Nevertheless, a look at the sources strongly indicates that the implications of this simple observation have not been fully appreciated. Its significance consists in the fact that the purpose of a positive permission can only be to *restrict* the scope and influence of an already existing prohibition or to pre-empt one that could possibly be passed. Declaring an action permitted does not add to the requirements imposed on people by a mandatory norm. From a logical point of view, therefore, positive permission is essentially *derogation*: A positive permission suspends, annuls or obstructs a covering prohibition, thereby generating a corresponding set of liberties.

In what I shall consider the principal case of positive permission, an implied positive permission suspends a general prohibition that is already in force—it is, one might say, an *exemption* from an operative ban. Consider the following example from §8 of the Norwegian personal information act:

§8. Personal information may only be processed by the consent of the registered person, or if processing is statutorily warranted, or such processing is required in order to

- (a) honour an agreement with the registered person, or to perform a task that accords with the registered person's wishes before such an agreement was entered into,
- (b) fulfill a legal obligation on the part of the person responsible for handling the information,
- (c) attend to the registered person's vital interests,
- (d) perform a task in the interest of the general public,
- (e) exercise public authority, or

---

<sup>5</sup>See for instance [4, 25, 27, 33] and [34]

- (f) attend to a justifiable interest that is not outweighed by the regard for the registered person's right to privacy.

As indicated by the word “only” in the opening sentence, accessing someone's personal information is in general prohibited. The statute then goes on to list a set of particular cases for which the prohibition is pronounced null or void. These cases are in effect exempted from the ban, and therefore constitute permissions.

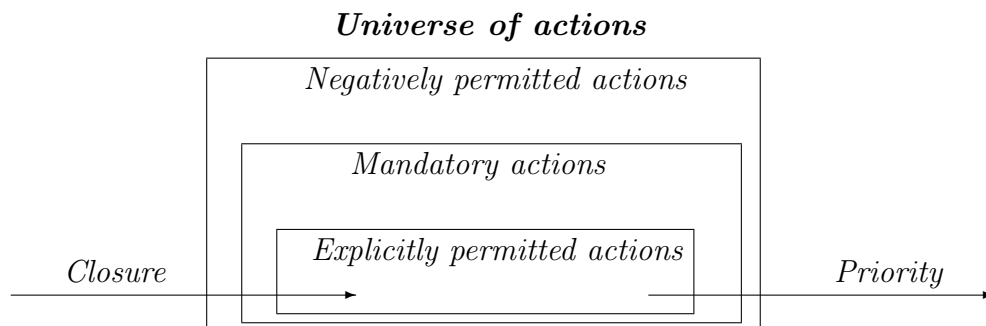
Alf Ross, for one, was very clear on this (although, as I shall argue later, he failed to draw the right conclusions): “Norms of permission have the normative function only of indicating, within some system, what are the exceptions from the norms of obligation of the system” [27, p. 120]. The Norwegian education act § 2-4 provides another example: “Christian teachings and ethical education is an ordinary school subject that shall normally be attended by all pupils”. The relevant sense of normality is specified by the attendant clause: “On the basis of written notification from parents, pupils shall be exempted from attending those parts of the teaching at the individual school that they on the basis of their own religion or philosophy of life, perceive as being the practice of another religion or adherence to another philosophy of life”. Here the mandatory norm is formulated as a duty, but nothing of importance turns on that since requiring attendance is the same as prohibiting absence.

Taking stock, the principal permissive unit, one might say, is a structure consisting of a *general mandatory norm + specific exemptions*. This structure bears instructive similarities to closed world databases, although the parallel should not be overburdened: Just as a closed world database operates under the assumption that what is not currently known to be true is false, a permissive structure operates under the presumption that what is not positively permitted is prohibited. In a closed world database everything is assumed to be false, as far as that is compatible with the information that is explicitly stated in the database. If information to the contrary is available, however, the presumption of falsity yields. A permissive structure relates prohibitions and exemptions in the same way. In the personal information act, the general prohibition against processing personal information works logically like a closed-world assumption. If a case is not listed as an exemption (or implied by one), then it isn't.

The interesting thing to note here is that the presumption of falsity in

both cases establishes a priority relation between two sources of information; a default assumption, on the one hand, and explicitly stated information on the other. If the explicitly given information is incompatible with the default assumption, then the latter is suspended. This may all seem obvious, and is certainly well-known from database theory. Nevertheless, this simple priority structure seems often to be lost out of sight when it comes to the concept of permission. As it is a *conditio sine qua non* for an adequate theory, it is important to understand its significance, and to keep it fixed in view.

Now, the same priority relationship in turn holds between negatively permitted actions and mandatory ones. The principle of negative permission, we recall, states that whatever is not prohibited by a code is permitted according to that code. It follows that if an action is prohibited, then it is not negatively permitted. In other words prohibitions assume priority over negative permissions, and negative permissions always yield to the greater force of a mandatory norm. Assembling all the parts of this emerging picture, we can illustrate a permissive structure as follows:



A few explanatory remarks are perhaps in order: A normative problem may be regarded as a question concerning the deontic status of *actions*. In the majority of cases, the act referred to in a norm is the production of a certain effect or change. Since there are infinitely many ways to specify the effect of an action, and infinitely many ways of bringing it about, the universe of actions must be assumed to be infinite. It follows that no norm rules out all freedom of choice, although one may say, following Alf Ross, that according to how precisely the action is specified, the norm is *more or less rigorous or discretionary*. Returning to the illustration, the outermost box must thus be understood as comprising an *infinite* set of actions. Now, the function of the



principle of negative permission is to cover this entire set *thereby closing the set of prohibitions and duties*, that is, nothing is obligatory if it is not stated to be so. The set of mandatory actions in turn closes the set of explicitly permitted ones, that is, nothing is positively permitted if *it* is not stated to be so. Hence the relationship between the three classes of actions may tentatively be illustrated by a three-level nesting of boxes where the closure of the respective domains goes from the outermost to the innermost box, whereas the order of priority<sup>6</sup> goes in the converse direction.<sup>7</sup> We can read off from this diagram a list of desiderata that any theory of permission should meet. Such a theory should;

1. Do justice to the negative character of explicit permission by casting them as exemptions,
2. Recognise the distinctness of explicit from negative permission,
3. Show that a permissive structure is a *unit* where
  - (a) Explicitly permitted actions are given priority over mandatory actions, and
  - (b) Mandatory actions are given priority over negatively permitted ones.

I shall use these desiderata as methodological guide-lines in what follows.

### 3. Input/output logic as a model of normative systems

The reader should be warned that the term “positive permission” is not quite apt, since positive permissions comprise exemptions and antithetic permissions. That is, the set of positive permissions includes cases for which

---

<sup>6</sup>This priority ordering, it should be stressed, is only valid within a permissive structure consisting of a mandatory norm and its particular exemptions. It is certainly not the case that permissive provisions always take priority over obligating norms. As was remarked by one of the referees, one may also find that general permissions have exemptions: A law may e. g. permit public demonstrations in general, but state that they are forbidden if some of the people carry weapons, say, or damage property.

<sup>7</sup>Note that antithetic permissions do not figure in this picture. An antithetic permission will turn out to be either a negative permission or a positive permission and there is no simple way to represent the determining factor graphically. However, we shall see that once the structure above has been given an adequate formal expression, a definition of antithetic permission naturally suggests itself.

a prohibition is deemed *inapplicable*—that is, *absent*. In a sense therefore, both negative and positive permissions can be negative insofar as they may arise in the wake of a prohibition. I shall nevertheless continue to talk about positive permission, since this terminology is already established, but the reader should bear the negative character of this concept in mind. Its significance lies in the fact that positive permissions, as well as negative permissions, are *systemic* or *holistic* properties. In general there are as many ways to permit an action as there are ways of blocking or reducing the consequences of a set of mandatory norms. This means that permissions, negative or positive, are most naturally evaluated against the total output of a *code*. A permission is not a separate modality or norm-character, it is a way of constraining a set of decrees—an *operation* on a code. Particularly when it comes to permissions therefore, it seems wise to heed the warning no logic of norms without attention to a system of which they form part [17, p. 29].

Just as the theoretical paradigm of a *theory* is a logically closed set of sentences (i.e. a set of sentences closed under entailment), the theoretical paradigm of a normative system may be taken to be a set of mandatory norms that contains all norms it entails (by some as yet unspecified notion of entailment). This is abstract, true, but will do well for purposes of conceptual analysis. One of the few such accounts on offer is the theory of input/output logic as set out in a series of papers by Makinson and van der Torre ([18, 19] and [20]). Input/output logic is deliberately designed to serve as an abstract model of normative *systems*, and I shall take it as my idiom of choice in the following.

In input/output logic a norm is simply a pair  $(a, b)$  correlating an *applicability condition*, *trigger* or *input*  $a$  with a *duty*, *optimality condition* or *output*  $b$ . I shall sometimes denote them neutrally as the antecedent and consequent of a norm respectively. For the purposes of the present paper the base language in which the antecedent and consequent of a norm are formulated is just propositional logic, but this is not a requirement. The base language needs to be closed under the boolean connectives, but may in addition contain other constructs, such as for instance deontic modalities. Hence, the  $b$  in  $(a, b)$  could very well be a normative proposition containing an *ought* or *must*. However, the choice of base language is of no consequence for the developments that follow, so long as it is a boolean language, so simplicity favours propositional logic.

A norm in the present paper therefore contains only declarative sentences.

Accordingly, we think of the antecedent as a description of the states of affairs to which the norm applies, and of the consequent as a description of a state of affairs that is considered mandatory whenever the antecedent is satisfied. I shall distinguish between mandatory *norms*, on the one hand, and the *obligations* or *requirements* they impose on the other. Norms have applicability conditions, requirements do not. Stated differently, requirements are the consequents of mandatory norms. In general, I shall use the term “norm” as a generic term for all relevant pairs  $(a, b)$  of formulae figuring in a system either as mandates or permissions. Thus, there will be *permissive norms* as well as mandatory ones.<sup>8</sup>

To be sure any really adequate representation of norms, requires much more than this. It needs to represent human, agency, the passage of time, bearers and counterparties of obligations and so on. Nevertheless, it seems wise to reserve more complex machinery until we have obtained a clear picture of the abstract structure, and until we have confirmed that the essential ideas are sound.

Note that a norm  $(a, b)$  in input/output logic is construed as a *logically arbitrary* stipulation connecting an *input*  $a$  with an *output*  $b$ —it is logically arbitrary in the sense that a pair is not a formula, so there is nothing to the norm  $(a, b)$  over and above the fact that some authority requires that  $b$  be done given  $a$ . One could see this as an expression of a kind of anti-naturalism, or conventionalism, wrt. to norms. The validity of a norm  $(a, b)$  need not have any ontological or epistemological status beyond that of being decreed to hold. As Kelsen says: “Norms posited by human acts of will are arbitrary in the genuine signification of the word: that is, they can decree any behaviour whatsoever to be obligatory” [15, p. 4]. Notably, the pair as such has no logic, the contrapositive of a norm is not necessarily a norm, nor is any pair of the form  $(a, a)$ . Hence, norms do not behave as material conditionals, and they do not satisfy the reflexivity property. This is important insofar as reflexivity would collapse the distinction between the actual and the ideal, making every fact optimal according to the norms.

A *code* of norms in input/output logic is simply a set  $G$  of such pairs, from

---

<sup>8</sup>There is ample precedence in the literature for such a generic use of the term “norm”: Hart [14] for instance speaks of power-conferring norms, whereas Raz [26] and Alchourron and Bulygin [4] distinguish, as I do, between permissive and mandatory norms

whence it follows that the explicitly declared requirements, in any situation  $a$  (or alternatively, on any *input*  $a$ ) according to  $G$ , can be obtained by taking the image of  $a$  under  $G$ . The basic notion of normative implicature in turn allows implicit norms to be derived from the explicit ones—i.e. from the ones contained in  $G$ —e. g. by recognizing that which implies (logically) the trigger of a norm as itself a trigger of a norm, and that which follows (logically) from an explicitly declared requirement as itself mandated by a norm. To be more precise, the basic model of a normative system is an operation  $out$  of type  $2^{L^2} \times 2^L \mapsto 2^L$  (where  $L$  is the language of propositional logic) defined as follows:

**Definition 3.1.**  $out(G, a) = Cn(G(Cn(a)))$ , where  $Cn(a)$  is the closure under logical entailment of the formula  $a$ .

This definition of the  $out$ -operator is, in a broad sense of the term, semantic. The task of logic is seen as a modest one. It is not to create or determine a distinguished set of norms, but rather to prepare information before it goes in as input to such a set  $G$ , to unpack output as it emerges and, if needed, coordinate the two in certain ways. A set  $G$  of conditional norms is thus seen as a transformation device, and the task of logic is to act as its “secretarial assistant” [18, p. 2].

However, a syntactic representation of the  $out$ -operator, can be given by defining  $(a, b) \in out(G)$  iff  $b \in out(G, a)$ .<sup>9</sup> This projection of the  $out$ -operation onto its left argument, although it can be regarded merely as a stylistic variant, entails a change of gestalt: We are now construing  $out$  as an operator mapping relations to relations, whence a normative system becomes a relation that is the value of (the monadic)  $out$  for some argument  $G$ . As it turns out, this latter relation, taking  $G$  as given, may be represented by the system that contains the following axioms

**Tautology:**  $(t, t)$  for arbitrary tautologies  $t$

**Inclusion:**  $(a, b)$  for  $(a, b) \in G$ .

and the following rules of inference;

---

<sup>9</sup>Overloading the terminology a bit, I shall refer to both the dyadic and the monadic  $out$ -operator, as simply an  $out$ -operator

**Input strengthening (SI):** From  $(a, b)$  to  $(c, b)$  whenever  $c \vdash a$ .

**Output weakening (WO):** From  $(a, b)$  to  $(a, c)$  whenever  $b \vdash c$ .

**Conjunctive conclusions (AND):** From  $(a, b)$  and  $(a, c)$  to  $(a, b \wedge c)$

In other words membership in  $out(G)$  corresponds to derivability from  $G$ , where derivability is understood in terms of membership in the least superset of  $G$  that contains  $(t, t)$  and is closed under *AND*, *SI* and *WO*. In [20] the latter set is denoted  $deriv(G)$ . Observation 1 of the same paper then shows that  $out(G) = deriv(G)$ . Note that these operators are closure operators. That is, they satisfy

**Idempotence:**  $out(out(G)) = out(G)$ ,

**Inclusion:**  $G \subseteq out(G)$ , and

**Monotony:**  $out(G) \subseteq out(G')$  whenever  $G \subseteq G'$

I shall have occasion to appeal to these later.

By modifying definition 3.1 in certain ways, other operators can be defined that satisfy more rules, such as cumulative transitivity, that are certainly interesting candidate principles for reasoning with norms. However, definition 3.1 will do fine for the purposes of this paper (i. e. introducing additional principles would not change anything), so I refer the reader to the sources.

Henceforth, I shall work with the monadic version of the *out* operator (definition 3.1 is still important by way of intuitive motivation, though). The remainder of this section records a few properties that will be important for the subsequent developments:

**Lemma 3.2 (Easy half of conditionalization).** *If  $(a \wedge c, b \rightarrow d) \in out(G)$  then  $(c, d) \in out(G \cup \{(a, b)\})$  whenever  $c \vdash a$ .*

**Proof.** Suppose that  $c \vdash a$  and  $(a \wedge c, b \rightarrow d) \in out(G)$ . Then  $(c, b \rightarrow d) \in out(G)$  by *SI*. Moreover  $(c, b) \in out(G \cup \{(a, b)\})$ , again by *SI*, so  $(c, d) \in out(G \cup \{(a, b)\})$ , by *AND*.

□

**Lemma 3.3 (Hard half of conditionalization).** *If  $(c, d) \in out(G \cup \{(a, b)\})$  then  $(a \wedge c, b \rightarrow d) \in out(G)$ .*

**Proof.** Argument proceeds by induction on the length of a derivation. For the basis of the induction suppose first that  $(c, d) = (t, t)$ . Since we have  $(t \wedge a, \neg b \vee t) = (t \wedge a, b \rightarrow t) \in out(G)$  for any  $a$  and  $b$ , by input strengthening and output weakening, we are done. Next, suppose  $(c, d) \in G$ . Then  $(c, b \rightarrow d) \in out(G)$  by output weakening, and  $(a \wedge c, b \rightarrow d) \in out(G)$  by input strengthening. For the induction step, suppose the property holds for shorter proofs, and

1. Suppose that  $(c, d) \in out(G \cup \{(a, b)\})$  and that  $(c, d)$  is derivable from  $(g, h) \in out(G \cup \{(a, b)\})$  by *SI*. Then  $c \vdash g$  and  $d = h$ . Since  $(g, h) \in out(G \cup \{(a, b)\})$ , it follows by the induction hypothesis that  $(a \wedge g, b \rightarrow h) \in out(G)$ . Since  $c \vdash g$  we thus have  $(a \wedge c, b \rightarrow h) \in out(G)$ , by *SI*, and since  $d = h$  we have  $(a \wedge c, b \rightarrow d) \in out(G)$  as desired.
2. Suppose that  $(c, d) \in out(G \cup \{(a, b)\})$  and that  $(c, d)$  is derivable from  $(g, h) \in out(G \cup \{(a, b)\})$  by *WO*. Then  $g = c$  and  $h \vdash d$ . Since  $(g, h) \in out(G \cup \{(a, b)\})$ , it follows by the induction hypothesis that  $(a \wedge g, b \rightarrow h) \in out(G)$ . Since  $g = c$  we thus have  $(a \wedge c, b \rightarrow h) \in out(G)$ , and since  $h \vdash d$  we have  $(a \wedge c, b \rightarrow d) \in out(G)$ , by *WO* as desired.
3. Suppose that  $(c, d) \in out(G \cup \{(a, b)\})$  and that  $(c, d)$  is derivable from  $(g, h), (g', h') \in out(G \cup \{(a, b)\})$  by *AND*. Then  $c = g = g'$  and  $d = h \wedge h'$ . By the induction hypothesis we have that  $(a \wedge g, b \rightarrow h), (a \wedge g', b \rightarrow h') \in out(G)$ , whence  $(a \wedge g \wedge g', b \rightarrow h \wedge h') \in out(G)$ , by *AND*. Since  $c = g = g'$  and  $d = h \wedge h'$ , it follows that  $(a \wedge c, b \rightarrow d) \in out(G)$ , by *SI* as desired.

This completes the proof. □

Thus, we have the following simple corollary:

**Corollary 3.4.** *If  $c \vdash a$  then  $(c, d) \in out(G \cup \{(a, b)\})$  iff  $(a \wedge c, b \rightarrow d) \in out(G)$ .*

Given the equality  $out(G) = deriv(G)$ , this corollary is analogous to the deduction theorem for classical logic, in the sense that it establishes a link between derivability and material implication. The differences should be obvious though; special care is required wrt. applicability conditions.

#### 4. Positive permission according to Makinson and van der Torre

The desiderata listed in section 2 rule out most accounts of permission on offer. Ross is on the right track when he says “I know of no permissive legal rule which is not logically an exemption modifying some prohibition”, but then he continues “and interpretable as the negation of a prohibition” [27, p. 122]. Of course explicit permission *is* interpretable as the negation of an obligation/prohibition insofar as what a permissive provision does is to render a prohibition null and void for a particular case. It is not thereby to be *identified*, as Ross seems to think, with a negative permission, however, because their respective positions in the priority ordering differ (cmp. the illustration): Permissive provisions override mandatory norms, and are therefore to a certain extent *protected*—in cases of conflict the permission prevails. Negative permissions, on the other hand, are overridden by mandatory norms and are therefore *exposed*—in cases of conflict the permission yields. In other words the negation of an obligation is not necessarily the same as a negative permission as the latter concept is here understood. Ross takes the negative character of explicit permission as evidence that the two can be identified, and his account consequently violates desideratum 2.

An inventory and evaluation of existing approaches is outside the scope of this thesis. One account that is of particular interest here, however, is that of [21], since it takes the same general point of view as that adopted here—that is, of treating positive permission holistically as a systemic property. It is also the first analysis of permission in an input/output idiom, and many of its central insights have influenced the present paper.<sup>10</sup> I shall argue, however, that it too violates the aforementioned desiderata.

A few notational conventions first: Since we wish to analyse permissions in the larger context of a system, the proper unit of analysis is a *code*  $\langle G, P \rangle$

---

<sup>10</sup>To my knowledge, there is only one other study of permission in input/output logic so far, and that is [31].

consisting of a set of *explicitly stated mandatory norms*  $G$  and a set of *explicitly stated permissive norms*  $P$ . I shall sometimes, for brevity, refer to  $P$  simply as the set of explicit permissions, although, strictly speaking, it is a set of permissive *norms*. In the general case where  $(a, b)$  is an implied norm, I shall say that it is a mandatory or a permissive, norm (as the case may be) *according to* or *in* such a code, in which case it means that  $b$  is required or permitted, respectively, by that code whenever  $a$  is *true*. Reformulated accordingly, Makinson and van der Torres' concept of negative permission becomes:

**Definition 4.1 (Negative permission).**  $(a, b)$  is negatively permitted according to  $\langle G, P \rangle$  iff  $(a, \neg b) \notin out(G)$ .

The interpretation of this definition is straightforward:  $G$  is the set of mandatory norms, so  $b$  is negatively permitted given  $a$  iff it is not prohibited under the same condition. Note that the set of explicit permissions  $P$  does not come into play. In other words the negatively permitted actions are taken to be those for which a contrary prohibition is absent, *not counting exemptions*. Note also that negative permission so construed trivially satisfies desideratum 3b, that is, it gives priority to prohibitive norms over negatively permitted actions, since  $(a, \neg b) \in out(G \cup \{(a, \neg b)\})$  for any  $G$ , so a negative permission  $(a, b)$  always yields to  $(a, \neg b)$  were the latter to be added to the code.

As regards the concept of antithetic permission (aka. *dynamic positive permission*), Makinson and van der Torre's definition translates:

**Definition 4.2 (Dynamic positive permission).**  $(a, b)$  is a dynamic positive permission according to  $\langle G, P \rangle$  iff  $(c, \neg d) \in out(G \cup \{(a, \neg b)\})$  for some positively permitted  $(c, d)$  in the same code.

I shall postpone a detailed discussion of this concept until later. Suffice it to say that I am not going to change it very much, since the general idea seems sound enough. In Makinson and van der Torres's words the idea is to see  $(a, b)$  as permitted whenever, given the obligations already present in  $G$ , we can't forbid  $b$  under the condition  $a$ , without thereby committing ourselves to forbid, under a condition  $c$ , something  $d$  that is implicit in what has been explicitly permitted [21, p. 398]. However, a more detailed understanding of its inner workings obviously requires an understanding of the kind of *positive*



permission that is appealed to here, and since I shall plug in a different concept of positive permission than Makinson and van der Torre, I'd rather return to it then. The concept they intend is one they call *static positive permission*:

**Definition 4.3 (Static positive permission).**  $(a, b)$  is a static positive permission according to  $\langle G, P \rangle$  iff  $(a, b) \in out(G \cup \{(c, d)\})$  for some  $(c, d) \in P$ .

As the observant reader will have noticed, definition 4.3 has the effect of including all norms in  $out(G)$  among the static positive permissions, implementing the principle that *ought* entails *may*. The idea behind the definition is to treat  $(a, b)$  as permitted iff there is some explicitly given permission  $(c, d)$  such that when we join it with the obligations in  $G$  and apply the output operation to the union, then we get  $(a, b)$  among the consequences [21, p. 397]. Static positive permissions are thus treated like *weak* obligations, the basic difference being that while obligations proper may be used jointly, permissions may only be applied one by one. This restriction is intended to capture the fact that two actions may be permitted under a common condition without being jointly so. For instance, whereas it is usually the case that drinking is permitted and that driving is permitted according to the same legal system, it is usually *not* the case that drinking *and* driving is permitted, so permissions cannot in general be conjoined.

Although this restriction to singleton applications of explicit permissions causes the operation of deriving a static positive permission from one or more static positive permissions to differ from an ordinary input/output operation such as *out* (the reader is referred to the cited paper for details), there is clearly an intimate relationship between them. Makinson and van der Torre are able to establish the following connection: Every rule of inference, such as e.g. *AND*, that is satisfied by an input/output operation (in this paper we consider only one; the operation *out* given by definition 3.1) is an instance of the general Horn form

$$HR : \frac{(a_1, b_1), \dots, (a_n, b_n)}{(g, h)} c \vdash d$$

where  $c$  and  $d$  may be tautologies (again in the case of *AND*). Now, for any given input/output operation, satisfaction of such a Horn rule is reflected

by a derivation rule for positively permitted norms that takes the form of a *subverse rule*: Label each norm  $(a, b)$  either  $(a, b)^o$  or  $(a, b)^p$ , depending on whether  $(a, b)$  is in  $out(G)$  or  $(a, b)$  is positively permitted according to definition 4.3. Then the subverse of a Horn rule is;

$$HR^\downarrow : \frac{(a_1, b_1)^o, \dots, (a_{n-1}, b_{n-1})^o \quad (a_n, b_n)^p}{(g, h)^p} \quad c \vdash d$$

In other words, the subverse rule is obtained by downgrading to permission status one of the premises and also the conclusion of the corresponding rule for mandatory norms [op. cit., p. 401]. For instance, the subverse of the *AND* rule (call it *P-AND* for convenience) is:

$$P-AND \frac{(a, b)^o \quad (a, c)^p}{(a, b \wedge c)^p}$$

Thus, even though two distinct permissions need not be jointly permitted, the conjunction of a positive permission and an obligation always is, since all input/output operations satisfy *AND* (this is only meant as an example, not a criticism). Similarly, positive permission also satisfies

$$P-SI \frac{(a, b)^p}{(c, b)^p} \quad c \vdash a$$

which, as I shall argue in the next section, is problematic, as well as

$$P-WO \frac{(a, b)^p}{(a, c)^p} \quad b \vdash c$$

Now, return to the example from the Norwegian personal information act: Let  $(t, \neg p)^o$  stand for “processing of personal information is prohibited” and  $(c, p)^p$  for “processing of personal information is permitted if the registered person gives his consent”. We have the following derivation:

$$P-AND \frac{SI \frac{(t, \neg p)^o}{(c, \neg p)^p} \quad (c, p)^p}{(c, p \wedge \neg p)^p} \\ P-WO \frac{(c, p \wedge \neg p)^p}{(c, q)^p}$$

So if the registered person agrees to let anyone access his information, then everything is permitted! Clearly something goes wrong here, and it is not too difficult to see what it is. The problem is that the concept of static positive permission does not establish a relation of priority between a permissive norm and the general prohibition to which it relates. Positive permissions and mandatory norms are treated as i/o pairs on the same logical level, so conflicts are not resolved. A direct application of the definition should make this even clearer: Put  $G := \{(t, \neg p)\}$  and  $P := \{(c, p)\}$ . Then, obviously  $(c, p \wedge \neg p) \in out(G \cup \{(c, p)\})$  by *SI* and *AND*. In other words  $(c, p)$  is not treated as an exemption, but just as another norm. But according to desideratum 3b that's wrong— $(c, p)$  is an exemption and should take priority. In a footnote Makinson and van der Torre comment: “We do not consider here the contractions or revisions that one might wish to make to the code when  $A$  [that is, the set of norms] is inconsistent with some  $z \in Z$  [that is, the set of permissions]. This is a separate matter, and forms part of the logic of normative change” [op. cit., p. 415]. But that, I am convinced, is plainly false. A positive permission is, in the principal case, an exemption, and exemptions *always* conflict with a background prohibition—that's the whole point. More generally, if a permissive provision does not conflict with a directive that is or could possibly be passed, then it simply has no purpose—there is nothing for it to do. In other words, the contractions or revisions that one might wish to make to the code when a mandatory norm conflicts with a permissive norm is decidedly not a separate matter. Indeed permissive norms simply *are* a kind of specification saying how this is to be done.

## 5. Permission as derogation

Summing up so far, I have argued that any theory of positive permission should take the negative character of the concept as fundamental. The purpose of a positive permission can only be to restrict the scope and influence of an already existing prohibition or to pre-empt one that could possibly be passed. In the former case the prohibition acts as an exception to a general prohibition that is already in force. In the latter case it serves as a shield against prohibitive laws that could conceivably be passed. Notwithstanding the (as yet unclarified) differences, if either kind of permission is to have a point, a prohibition must be assumed. Thus, it is the purpose of a positive permission to act as a constraint on valid or applicable law, and to suspend a lower-ranking prohibition in cases where that prohibition contradicts the

permissive provision.

This informal analysis suggests that one way of looking at positive permission is in terms of *derogation*, where “derogation” is used as a term of art to denote the elimination (temporary or not) of a norm from a normative system. In other words, derogation is taken to be the norm-theoretic analogue of *contraction*, and the working hypothesis is that the concept of positive permission can be fruitfully analysed in terms of it.

In classical revision-theory, a contraction on a set is carried out by intersecting maximally non-implying subsets, aka. remainders. That is, to remove an element  $a$  from a theory  $A$  one considers subsets of  $A$  that are such that they do not entail  $a$  whereas all proper supersets do. The outcome of the operation is then taken to be that which all selected remainders agree on. This has proved to be a robust and sustainable mathematical idiom, and I see no reason to deviate from it. Hence:

**Definition 5.1 (Remainders).**  $out(G) \perp (a, b)$  is the set of  $H$  such that

1.  $H \subseteq out(G)$
2.  $(a, b) \notin out(H)$ , and
3. If  $H \subset I \subseteq G$  then  $(a, b) \in out(I)$

Note that we are taking remainders of the *closure*  $out(G)$  rather than of the base  $G$ , so we are aiming for a version of *theory contraction* rather than, in the terminology of [13], *base-contraction*. Generalising the analysis to base-contraction should be straightforward, once a skeleton theory is in place.

An important property of remainders, as so defined, is that they are closed under the *out*-operation:

**Lemma 5.2.** *If  $G = out(G)$  and  $(a, b) \in G$ . Then  $H = out(H)$  for every subset  $H$  of  $G$  which is maximally such that  $(a, b) \notin out(H)$ .*

**Proof.** By inclusion for *out* (recall that *out* is a closure operator), it suffices to show that  $(c, d) \in H$  whenever  $(c, d) \in out(H)$ . Suppose therefore that  $(c, d) \in out(H)$ . By monotony for *out* we have that  $out(H) \subseteq out(G)$ , hence  $(c, d) \in G$  using the supposition that  $G = out(G)$ . Now, suppose for *reductio ad absurdum* that  $(c, d) \notin H$ . Then by the maximality of  $H$  and

the fact that  $(c, d) \in G$  we know that  $(a, b) \in \text{out}(H \cup \{(c, d)\})$ . But since  $(c, d) \in \text{out}(H)$ , by assumption, we have that  $\text{out}(H) = \text{out}(H \cup \{(c, d)\})$  so that  $(a, b) \in \text{out}(H)$ , contrary to hypothesis.

□

Let, for the time being, the derogation operation be defined by the *full meet* of the remainder set:

**Definition 5.3.**  $\text{out}(G) - (a, b) := \bigcap(\text{out}(G) \perp (a, b))$ <sup>11</sup>

Full meet contraction on sets of *formulae* is known to be a bit too heavy-handed, as it discards an unnecessarily large amount of information. It is reasonable to expect—one should at least anticipate the possibility—that some of these problems may carry over into input/output logic. A *partial meet* strategy, in the terminology of [3], should be expected to perform better.<sup>12</sup> For the sake of conceptual clarity, however, I have chosen to keep the framework as simple as possible, and to leave the appropriate generalisations for later. It should be said, though, that some of the properties established in what follows depend on the underlying derogation operator's being a full meet operator. These properties would thus have to be enforced by a condition on the selection function in a partial meet framework. It would be an interesting exercise to formulate these conditions and to see what they have to say about the structure of a normative system, but I leave this for future research. For the time being I shall be content to wave a hand wherever I find that appropriate and illuminating.

Full meet derogation satisfies the following properties (they should look familiar to anyone coming from revision theory):

**Lemma 5.4.** *Full meet derogation satisfies;*

$$\textbf{Closure: } \text{out}(G) - (a, b) = \text{out}(\text{out}(G) - (a, b))$$

---

<sup>11</sup>It is important not to confuse the *derogation* operation  $\text{out}(G) - (a, b)$  with set-theoretic difference. I shall use a backslash to denote the latter, as in  $\text{out}(G) \setminus (a, b)$ .

<sup>12</sup>A complete characterisation of partial meet derogation on input/output systems can be found in [29].

**Vacuity:**  $out(G) \subseteq out(G) - (a, b)$  whenever  $(a, b) \notin out(G)$

**Failure:** If  $\vdash b$  then  $out(G) \subseteq out(G) - (a, b)$

**Inclusion:**  $out(G) - (a, b) \subseteq out(G)$

**Success:** If  $(a, b) \in out(G)$  and  $\nvdash b$  then  $(a, b) \notin out(G) - (a, b)$

**Local Recovery:** If  $(a, b) \in out(G)$  then  $(a, b) \in out((out(G) - (a, c)) \cup \{(a, c)\})$

**Proof.** Closure is an easy consequence of lemma 5.2. **Vacuity**, **Inclusion**, **Failure** and **Success** all follow immediately from the definitions 5.1 and 5.3. For local recovery, suppose that  $(a, b) \in out(G)$ . We want to show that  $(a, b) \in out((out(G) - (a, c)) \cup \{(a, c)\})$ . By *AND* it suffices to show that  $(a, c \rightarrow b) \in out(G) - (a, c)$ . Note that  $(a, c \rightarrow b) \in out(G)$ , by *WO*, since  $(a, b) \in out(G)$ . Suppose for *reductio ad absurdum* that  $(a, c \rightarrow b) \notin H$  for some  $H \in out(G) \perp (a, c)$ . Then by the maximality of  $H$  it follows that  $(a, c) \in out(H \cup \{(a, c \rightarrow b)\})$ , so lemma 3.3 yields  $(a, (c \rightarrow b) \rightarrow c) \in H$ . Now,

$$\begin{aligned}
(c \rightarrow b) \rightarrow c &\vdash \neg(c \rightarrow b) \vee c \\
&\vdash \neg(\neg c \vee b) \vee c \\
&\vdash (c \wedge \neg b) \vee c \\
&\vdash c
\end{aligned}$$

Hence  $(a, c) \in H$ , by lemma 5.2, and one application of *WO*, contradicting  $H \in out(G) \perp (a, c)$ .

□

The reader should be aware that the framework of theory contraction generates certain anomalies when applied to sets of norms. I. e. using theory contraction, rather than base-contraction has disadvantages when what is discussed are not theories or beliefs but normative systems. For instance, if  $G = (t, a \wedge b)$  and  $P = (t, b)$  then  $(t, a)$  is still in  $out(G) - (t, b)$ . This result is often counterintuitive. For instance, filling in and returning a tax-form is not an obligation that can be partially satisfied by complying with one of

the conjuncts only. Rather, these actions are, we may say, *deontically interdependent*; there is no point in filling in a tax form if you do not return it, and there is no point in returning a blank form. Deontic interdependence, if we agree to call it that, is usually what the law giver of a conjunctive obligation has in mind. Otherwise the conjuncts would have been separated before including the norms. Base-contraction (I shall assume familiarity with the general idea) has the advantage that it respects these differences in the formulation of a code. In particular,  $(t, a)$  will not linger after  $(t, a \wedge b)$  has been removed from the *base*  $G$ .<sup>13</sup> I shall leave further exploration of this point for future research.

Returning now to the concept of positive permission, recall that exemptions are essentially exemptions *from* something. I. e. an exemption always relates to a background prohibition. Antithetic permissions, on the other hand, for instance constitutional guarantees, are derived from explicit permissions that do not relate to any prohibition in particular, but are meant to reject in advance certain prohibitions that could conceivably be passed. In both cases though, the permission acts as a constraint on valid or applicable law, so it is the mechanism of constraining that plays center stage. The basic difference is that in the case of antithetic permissions, no prohibition contradicting the pronounced permission yet exists, so the constraint is idle. Hence it seems natural to treat exemption as the principal concept of positive permission, and to proceed from there to an analysis of antithetic permission. A first shot at a definition might be:

**Definition 5.5 (Exemptions I).**  $(a, b)$  is an exemption according to a code  $\langle G, P \rangle$  iff  $(a, \neg b) \in \text{out}(G) \setminus \text{out}(G) - (c, \neg d)$  for some  $(c, d) \in P$ .

I shall say that  $(a, b)$  is an exemption *by* the explicit permission  $(c, d)$ . Exemptions are thus cast as cut-backs on the code required to respect the explicit permissions in  $P$ . More precisely  $(a, b)$  is an exemption if the code contains a prohibition that regulates the state of affairs  $a$  by prohibiting  $b$ , and  $(a, \neg b)$  is such that, unless it is removed, the code will contradict an explicit permission in  $P$ .

---

<sup>13</sup>This was pointed out to me by one of the referees. A related point is discussed in [29, chap. 5].

Note that according to definition 5.5 if there is some  $(a, b)$  such that  $(a, b)$  is an exemption by  $(c, d)$  then  $(c, d)$  is an exemption by itself:

**Lemma 5.6.** *If  $(a, \neg b) \in out(G) \setminus out(G) - (c, \neg d)$  for some  $(c, d) \in P$  then  $(c, \neg d) \in out(G) \setminus out(G) - (c, \neg d)$ .*

**Proof.** Since  $out(G) \not\subseteq out(G) - (c, \neg d)$ , it follows that  $\not\vdash \neg d$ , by **Failure**. Moreover, **Vacuity** gives us  $(c, \neg d) \in out(G)$ , for the same reason. Therefore  $(c, \neg d) \in out(G) \setminus out(G) - (c, \neg d)$ , by **Success**.

□

Note also, that if  $(c, \neg d) \notin out(G)$ , i.e. if  $(c, \neg d)$  is not a norm that is derivable in the system, then  $out(G) - (c, \neg d)$  will be identical to  $out(G)$ , by **Vacuity** and **Inclusion** so  $out(G) \setminus out(G) - (c, \neg d)$  will be empty. Therefore, unless an explicit permission is in direct conflict with a norm derivable in the code, the permission will not be an exemption from any norm in the code. This is intentional and indicates (by exclusion) the kind of case in which an explicit permission will be considered as a preemption, or a protection or a shield, against effective direct or indirect introduction of a certain potential mandatory (i.e. prohibitive) norm, rather than as an exemption from an existing norm.<sup>14</sup>

To see how definition 5.5 behaves, consider the following example:

**Example 5.7.** *Put  $G := \{(t, \neg p)\}$  and  $P := \{(c, p)\}$ . Think of these norms as a general prohibition against processing personal information and as an exception for express consent respectively. We have  $(c, \neg p) \in out(G)$  by input strengthening. By success for full meet derogation, however,  $(c, \neg p) \notin out(G) - (c, \neg p)$ , so  $(c, p)$  constitutes an exemption.*

Exemptions, so defined, satisfy several interesting properties, for instance;

**Lemma 5.8 (Output weakening).** *If  $(a, b)$  is an exemption in  $\langle G, P \rangle$  then so is  $(a, c)$ , given that  $(a, \neg c) \in out(G)$  and  $b \vdash c$ .*

---

<sup>14</sup>I owe this particular formulation to one of the referees.



**Proof.** Suppose that  $(a, b)$  is an exemption in  $\langle G, P \rangle$  and that  $b \vdash c$ . From the former assumption we have that  $(a, \neg b) \in \text{out}(G) \setminus \text{out}(G) - (g, \neg h)$  for some  $(g, h) \in P$ . Assume now that  $(a, \neg c) \in \text{out}(G)$ . It suffices to show that  $(a, \neg c) \notin \text{out}(G) - (g, \neg h)$ . Suppose for *reductio ad absurdum* that  $(a, \neg c) \in \text{out}(G) - (g, \neg h)$ . We have  $b \vdash c$  by assumption, whence  $\neg c \vdash \neg b$  by contraposition. It follows that  $(a, \neg b) \in \text{out}(G) - (g, \neg h)$ , by **Closure** for full meet derogation and one application of *WO*. But by assumption  $(a, b)$  is an exemption, so  $(a, \neg b) \notin \text{out}(G) - (g, \neg h)$ —a contradiction.

□

Thus if it is permitted to process some item of personal information on a given condition, and we assume that processing entails access, then accessing the information is allowed. An easy consequence of output weakening is the following property of disjunctive exemption:

**Lemma 5.9 (Disjunctive exemption).** *If  $(a, b)$  and  $(a, c)$  are exemptions according to  $\langle G, P \rangle$  then  $(a, b \vee c)$  is an exemption according to the same code.*

**Proof.** Suppose  $(a, b)$  and  $(a, c)$  are both exemptions according to  $\langle G, P \rangle$ . Then  $(a, \neg b), (a, \neg c) \in \text{out}(G)$ , so  $(a, \neg b \wedge \neg c) \in \text{out}(G)$ , by *AND*, and  $(a, \neg(b \vee c))$  by *WO*. Thus, since  $b \vdash b \vee c$ , it follows by lemma 5.8 that  $(a, b \vee c)$  is an exemption in  $\langle G, P \rangle$ .

□

Both these properties seem relatively intuitive and desirable. Much more problematic is the property of *input* weakening:

**Lemma 5.10 (Input weakening).** *If  $(a, b)$  is an exemption according to  $\langle G, P \rangle$  then  $(c, b)$  is an exemption according to the same code whenever  $(c, \neg b) \in \text{out}(G)$  and  $a \vdash c$ .*

**Proof.** Suppose  $(a, b)$  is an exemption in  $\langle G, P \rangle$ . Then  $(a, \neg b) \in \text{out}(G) \setminus \text{out}(G) - (g, \neg h)$  for some  $(g, h) \in P$ . Suppose further that  $(c, \neg b) \in \text{out}(G)$  and that  $a \vdash c$ . It suffices to show that  $(c, \neg b) \notin \text{out}(G) - (g, \neg h)$ . Suppose to the contrary that  $(c, \neg b) \in \text{out}(G) - (g, \neg h)$ . Then by **Closure** for full meet derogation we have that  $(a, \neg b) \in \text{out}(G) - (g, \neg h)$ , by one application of *SI*. This contradicts the assumption that  $(a, b)$  is an exemption, so the proof is complete.

□

The property of input weakening for conditional permission is briefly discussed in [17], where the following example is offered by way of motivation: “Given that it is permitted to mow the lawn on Sunday between 10h00 and 12h00, I may conclude that it is permitted to do so on Sunday, but I may not conclude from the latter that it is permitted to mow on Sunday afternoon”. However, a look at more realistic examples quickly reveals that this is not a valid pattern of reasoning. Consider this time the personal information act §9, c:

§9 Sensitive personal information may only be processed if one of the conditions in §8 is met and

- c) processing is required in order to attend to the registered persons’ vital interests, and the registered person is unable to consent

If we accept input weakening, then we may conclude that processing is permitted if the registered person is simply unable to consent. Thus, shooting him is one way of obtaining permission to access his information. The following regimented example brings out the problem more clearly:

**Example 5.11.** Put  $G := \{(t, \neg p)\}$  and  $P := \{(c, p)\}$ , where we interpret the norms as in preceding examples. Then  $(c, p)$  is an exemption according to  $\langle G, P \rangle$ , and, by input weakening, so is  $(t, p)$ . Hence, given that processing of personal information is permitted on some condition, then it is permitted unconditionally.

Imagine an information infrastructure such as that envisioned by Weitzner et al., where the system assumes responsibility for answering queries such as “am I allowed to use this information?”. Obviously, input weakening would constitute a serious systemic anomaly, as it would make the system answer ‘yes’ in all circumstances, given that *any* condition allows the information to be used. I conclude that input weakening is not a desirable property. Nor is input *strengthening*, actually, since a permission may be toggled on and off under increasingly specific circumstances. Norwegian intellectual property law (LOV-2006-12-22-103) provides one example. §2 states a general restriction on the production of copies: “Intellectual property gives exclusive rights

to produce copies, temporary or permanent”. An exception is recognised in §11a: “If a temporary representation of a work is essential to a process whose sole purpose is to facilitate the legitimate use of the work then §2 is suspended”. The statute then goes on to state an exception in turn to this exception: “this provision does not apply to computer programs and databases”. Hence, a permission to produce a copy of a piece of intellectual property may be toggled off again when more is known about the circumstances and the nature of the work.<sup>15</sup> This strongly suggests that permissive norms should be regarded as *classical with respect to the antecedent*, in the terminology of [11]. The definition should therefore be modified as follows:

**Definition 5.12 (Exemptions II).**  $(a, b)$  is an exemption according to the code  $\langle G, P \rangle$  iff  $(a, \neg b) \in \text{out}(G) \setminus \text{out}(G) - (c, \neg d)$  for some  $(c, d) \in P$  such that  $c \equiv a$ .

This definition satisfies neither input weakening nor strengthening. I’ll show the failure of the former only:

**Example 5.13.** Let  $G := \{(t, \neg p)\}$  and  $P := \{(c, p)\}$ . Then  $(c, \neg p) \in \text{out}(G)$  by *SI*, but  $(c, \neg p) \notin \text{out}(G) - (c, \neg p)$  by the property of success for derogation, so  $(c, p)$  is an exemption according to  $\langle G, P \rangle$ . However, there is no  $(t, q) \in P$  for any  $q$ , so the condition  $(t, \neg p) \in \text{out}(G) \setminus \text{out}(G) - (t, q)$  fails for all  $q$ . Hence  $(t, p)$  is not an exemption according to  $\langle G, P \rangle$  so input weakening fails.

I shall take 5.12 as my “official” definition. As can be seen, conditional permission, although classical with respect to the antecedent, is nevertheless *normal* with respect to the consequent, and thus still satisfies output weakening and disjunctive exemption. Moreover, it should be clear that the definition conforms to our desiderata. The permission that acts as a constraint is itself immune to derogation, and therefore always assumes priority

---

<sup>15</sup>A natural extension to the system presented here would thus be to allow exceptions to permissive norms as well. One idea that suggests itself is to give the notion of a code a recursive structure. For instance: A code is either a pair  $\langle G, P \rangle$  or it is pair  $\langle G, C \rangle$  where  $G$  is a set of mandatory norms, and  $C$  is a code. All references to  $P$  in the definitions presented in this section would then be replaced by  $C$ . I have not yet looked into this though, and do not really know if it would work.

over prohibitions.

Finally, it can be shown that the set of exemptions is closed under exemptions, i.e. that 5.12 is a definitional schema that can be used iteratively in the following sense:<sup>16</sup>

**Lemma 5.14.**  *$(a, b)$  is an exemption in  $\langle G, P \rangle$  whenever  $(a, \neg b) \in out(G) \setminus out(G) - (c, \neg d)$  for some exemption  $(c, d)$  in  $\langle G, P \rangle$  with  $a \equiv c$ .*

**Proof.** Suppose  $(a, \neg b) \in out(G) \setminus out(G) - (c, \neg d)$  for  $a \equiv c$ . Then, there is an  $F \in out(G) \perp (c, \neg d)$  such that  $(a, \neg b) \notin F$ . It follows, by the maximality of  $F$  that  $(c, \neg d) \in out(F \cup \{(a, \neg b)\})$ , whence  $(a \wedge c, \neg b \rightarrow \neg d) \in F \subseteq out(G)$ , by lemma 3.3. Now,  $(c, d)$  is an exemption in  $\langle G, P \rangle$ , by assumption, so there is a pair  $(g, h) \in P$  with  $g \equiv c$  such that  $(c, \neg d) \notin out(G) \setminus out(G) - (g, \neg h)$ . Hence  $(c \wedge g, \neg d \rightarrow \neg h) \in out(G)$  by reasoning similar to that in the previous step. We therefore have the following derivation:

$$\begin{array}{c} SI \frac{(a \wedge c, \neg b \rightarrow \neg d)}{(a, \neg b \rightarrow \neg d)} \quad SI \frac{(c \wedge g, \neg d \rightarrow \neg h)}{(a, \neg d \rightarrow \neg h)} \\ AND \frac{\quad}{(a, (\neg b \rightarrow \neg d) \wedge (\neg d \rightarrow \neg h))} \\ WO \frac{\quad}{(a, \neg b \rightarrow \neg h)} \end{array}$$

So  $(a, \neg b \rightarrow \neg h) \in out(G)$ , by the equality  $out(G) = deriv(G)$ . Now, to prove that  $(a, \neg b) \in out(G) \setminus out(G) - (g, \neg h)$ , it suffices to find an  $H \in out(G) \perp (g, \neg h)$  such that  $(a, \neg b) \notin H$ . Consider the set  $H^- := \{(a, \neg b \rightarrow \neg h)\}$ . Clearly  $(g, \neg h) \notin out(H^-)$ , unless  $\vdash \neg b$  which, by **Success**, must be false since  $(a, \neg b) \notin out(G) - (c, \neg d)$ . Hence  $H^-$  can be extended to a maximal subset  $H$  of  $out(G)$  such that  $(g, \neg h) \notin H$ . Moreover  $(a, \neg b) \notin H$ , by **AND**, since  $(a, \neg b \rightarrow \neg h)$  is. Therefore  $(a, \neg b) \notin out(G) - (g, \neg h)$ , since the derogation operator is full meet. It follows that  $(a, b)$  is an exemption in  $\langle G, P \rangle$  as desired. □

---

<sup>16</sup>This possibility was suggested by one of the referees.

Now, having the set of exemptions under reasonable control we are now in position to say something substantial about antithetic permission. Recall that the idea, as put in words by Makinson and van der Torre (who in turn give Alchourron credit for it), is to see  $(a, b)$  as permitted whenever, given the mandatory norms in  $G$ , we can't forbid  $b$  under the condition  $a$  without thereby committing ourselves to forbid, under a condition  $c$  that could possibly be fulfilled, something  $d$  which is implicit in what has been explicitly permitted. Now, that which a code explicitly pronounces to be permitted are just the elements in  $P$ , and exemptions are permissions implicit in  $P$ . Hence antithetic permission becomes;

**Definition 5.15 (Antithetic permission).**  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$  iff  $(c, \neg d) \in out(G \cup \{(a, \neg b)\})$  where  $(c, d)$  is an exemption or an explicit permission according to the same code, and  $a \equiv c$ .

This is a straightforward translation, with minor modifications, of the definition of dynamic positive permission from [21], the differences being two: Firstly, we are now plugging in a different concept of positive permission. Secondly, in order to avoid the problems that ensue from input weakening, antithetic permissions are required to be classical only with respect to the antecedent. The next theorem gives another representation that will come in handy later:

**Theorem 5.16.**  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$  iff  $(a, \neg b \rightarrow \neg d) \in out(G)$  where  $(c, d)$  is an exemption or an explicit permission in the same code such that  $a \equiv c$ .

**Proof.** From left to right, suppose  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$ . Then  $(c, \neg d) \in out(G \cup \{(a, \neg b)\})$  for some exemption or explicit permission  $(c, d)$ . It follows by lemma 3.3 that  $(a \wedge c, \neg b \rightarrow \neg d) \in out(G)$ , whence, since  $a \equiv c$ ,  $(a, \neg b \rightarrow \neg d) \in out(G)$ , by *SI*. For the converse direction, suppose  $(a, \neg b \rightarrow \neg d) \in out(G)$ , and that  $(c, d)$  is such that  $a \equiv c$ . Then by *AND* and *SI* we have  $(c, \neg d) \in out(G \cup \{(a, \neg b)\})$ , so we are done.

□

The next example gives a simple illustration of the behaviour of this concept:

**Example 5.17.** Put  $G := \{(a, d \rightarrow b)\}$  and  $P := \{(a, d)\}$ . Then  $(a, b)$  is antithetically permitted since  $(a, \neg b \rightarrow \neg d) \in out(G)$  and  $(a, d) \in P$ . However,  $(a, b)$  is not an exemption, since  $(a, \neg b) \notin out(G)$ .

As the example shows, antithetic permission does not coincide with exemption, but there is obviously a quite close relationship between them. The next theorem brings this relationship out clearly:

**Theorem 5.18.** If  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$ , then it is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$ .

**Proof.** Suppose  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$ . Then, by theorem 5.16,  $(a, \neg b \rightarrow \neg d) \in out(G)$  for a permissive norm  $(c, d)$  in  $\langle G, P \rangle$  with  $a \equiv c$ . By the definition of antithetic permission  $(c, d)$  is either an exemption or an explicit permission. The argument thus splits into cases:

If  $(c, d)$  is an exemption in  $\langle G, P \rangle$ , then we have that  $(c, \neg d) \in out(G) \setminus out(G) - (g, \neg h)$  for some  $(g, h) \in P$  with  $g \equiv c$ . It follows that  $(c, \neg d) \notin H$  for some  $H \in out(G) \perp (g, \neg h)$ , whence  $(g, \neg h) \in out(H \cup \{(c, \neg d)\})$  by the maximality of  $H$ . Therefore  $(c \wedge g, \neg d \rightarrow \neg h) \in H \subseteq out(G)$ , by lemma 3.3 and lemma 5.2, whence  $(a, \neg d \rightarrow \neg h) \in H \subseteq out(G)$ , by *SI* and  $a \equiv c \equiv g$ . Since both  $(a, \neg b \rightarrow \neg d), (a, \neg d \rightarrow \neg h) \in out(G)$ , therefore, we have  $(a, \neg b \rightarrow \neg h) \in out(G)$  by *AND* and *WO*. Now, to show that  $(a, b)$  is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$  it suffices to show that  $(a, \neg b) \in out(G \cup \{(a, \neg b)\})$  and that  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (g, \neg h)$ . The former follows immediately from the fact that *out* is a closure operator. For the latter it suffices to find an  $H \in out(G \cup \{(a, \neg b)\}) \perp (g, \neg h)$  such that  $(a, \neg b) \notin H$ . But the existence of such a set follows immediately from the fact that  $(a, \neg b \rightarrow \neg h) \in out(G \cup \{(a, \neg b)\})$ , for we may expand  $\{(a, \neg b \rightarrow \neg h)\}$  to a maximal subset  $H$  of  $out(G \cup \{(a, \neg b)\})$  such that  $(g, \neg h) \notin H$ , in which case  $(a, \neg b) \notin H$ , by construction of  $H$ , *AND* and  $g \equiv a$ .

Similarly, when  $(c, d)$  is an explicit permission we need to show that  $(a, b) \in out(G \cup \{(a, \neg b)\})$  and that  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (c, \neg d)$ , the former of which is immediate. The argument for the latter is essentially a rerun of the argument for the existence of an appropriate maximal subset of  $out(G \cup \{(a, \neg b)\})$  from above: We have  $(a, \neg b \rightarrow \neg d) \in out(G \cup \{(a, \neg b)\})$  by lemma 5.16 and monotony for *out*. By reasoning similar to that above, the set  $\{(a, \neg b \rightarrow \neg d)\}$  can be extended to a maximal subset  $F$  of  $out(G \cup \{(a, \neg b)\})$  such that  $(c, \neg d) \notin F$ . Clearly  $(a, \neg b)$  is not in  $F$  so we are done.

□

Despite its simple appearance, this is a non-trivial property that depends crucially on the underlying derogation operation being a full meet operation, for we need to establish the existence of a remainder in which the added element  $(a, \neg b)$  is not included. Were we to take a *partial* meet operation as basic then the property would not in general hold. An interesting exercise would then be to investigate the conditions under which it *would* be valid. Although I shall leave that question for future research, it is clear that an answer would consist in formulating the conditions under which the added element  $(a, \neg b)$  is to be given a lower priority than certain elements already present in the code, thus seeing to it that a remainder not having  $(a, \neg b)$  would be chosen by the selection function. Clearly therefore, the property above would say something substantial about the priority structure of the given code. I shall leave all this aside for the moment. Let me just say that I regard it as a virtue of the theory developed here, that such questions become clearly visible.

The converse of theorem 5.18 holds on *any* derogation operation:

**Theorem 5.19.** *If  $(a, b)$  is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$  then it is antithetically permitted in  $\langle G, P \rangle$ .*

**Proof.**  $(a, b)$  is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$  if  $(a, \neg b) \in out(G \cup \{(a, \neg b)\}) \setminus out(G \cup \{(a, \neg b)\}) - (a, \neg d)$  for some  $(a, d) \in P$ . It thus suffices to show that  $(a, \neg b \rightarrow \neg d) \in out(G)$  by theorem 5.16. By **Local Recovery** we have  $(a, \neg d) \in out((out(G) - (a, \neg b)) \cup \{(a, \neg b)\})$ , whence  $(a, \neg b \rightarrow \neg d) \in out(G) - (a, \neg b) \subseteq out(G)$  by **Inclusion** for full meet derogation.

□

Antithetic permissions are thus precisely the pairs that would constitute exemptions if the code were expanded with a covering prohibition. This agrees well with intuition I think, and also finds support in the sources:

This is what happens with constitutional rights and guarantees: the constitution rejects in advance certain norm-contents (that

would affect basic rights), preventing the legislature from promulgating this norm-content, for if the legislature promulgates such a norm-content, it can be declared unconstitutional by the courts and will not be added to the system [5, pp. 397–398].

Theorems 5.18 and 5.19 spell out, with welcome precision, what it means for a permissive provision, such as e.g. a constitutional guarantee, to reject a norm *in advance*, as Alchourron and Bulygin puts it. Another way to say the same is that a permissive provision in a merely antithetic use represents a commitment not to allow a code to *grow* in certain specified ways. Indeed, the ‘checked growth’-perspective is an equivalent way of looking at things:

**Theorem 5.20.**  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$  iff  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (c, \neg d)$  for some  $(c, d) \in P$  with  $a \equiv c$ .

**Proof.** For the left-to-right direction, if  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$ , then it is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$ , by lemma 5.18, whence  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (c, \neg d)$  for some  $(c, d) \in P$  with  $a \equiv c$ , as desired. For the converse direction suppose  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (c, \neg d)$  for some  $(c, d) \in P$  with  $c \equiv d$ . If  $(c, \neg d) \notin out(G \cup \{(a, \neg b)\})$ , then  $out(G \cup \{(a, \neg b)\}) - (c, \neg d) = out(G \cup \{(a, \neg b)\})$ , by **Vacuity** for full meet derogation, contradicting  $(a, \neg b) \notin out(G \cup \{(a, \neg b)\}) - (c, \neg d)$ . Hence  $(c, \neg d) \in out(G \cup \{(a, \neg b)\})$ , so  $(a, \neg b \rightarrow \neg d) \in out(G)$  by lemma 3.3 and *SI* and the assumption that  $a \equiv c$ . Hence  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$ , which is what we wished to show. □

As regards entailment relationships, example 5.17 shows that antithetic permissions need not be exemptions, but the converse direction holds:

**Theorem 5.21.** *Exemptions are antithetically permitted*

**Proof.** Suppose  $(a, \neg b) \in out(G) \setminus out(G) - (a, \neg d)$  for some  $(c, d) \in P$  with  $a \equiv c$ . Then there is an  $H \in out(G) \perp (c, \neg d)$  such that  $(a, \neg b) \notin H$ . Since  $(a, \neg b) \in out(G)$ , it follows by the maximality of  $H$  that  $(c, \neg d) \in out(H \cup \{(a, \neg b)\})$ , whence  $(a, \neg b \rightarrow \neg d) \in H \subseteq out(G)$ , by lemma 3.3, *SI* and the assumption that  $a \equiv c$ . Therefore,  $(a, b)$  is antithetically permitted by theorem 5.16.



□

Similarly:

**Theorem 5.22.** *Explicit permissions are antithetically permitted*

**Proof.** Suppose  $(c, d) \in P$ . Since  $(a, \neg d \rightarrow \neg d) \in out(G)$ , for any  $G$ , it follows that  $(c, d)$  is an antithetic permission in  $\langle G, P \rangle$  by theorem 5.16.

□

Moreover, exemptions are *precisely* the antithetically permitted norms whose local negation—where  $(a, \neg b)$  is the local negation of  $(a, b)$ —is in the given code  $out(G)$ :

**Theorem 5.23.** *If  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$  and  $(a, \neg b) \in out(G)$  then  $(a, b)$  is an exemption in  $\langle G, P \rangle$*

**Proof.** Since  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$  it is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$  by theorem 5.18. Since by assumption  $(a, \neg b) \in out(G)$  we have  $\langle G \cup \{(a, \neg b)\}, P \rangle = \langle G, P \rangle$ , so we are done.

□

These theorems taken together go a long way, I think, towards clarifying the exact relationship between explicit permission (items in  $P$ ), exemptions and antithetic permission (the three species of positive permission). To complete the picture we need to bring negative permission back into it. Note first that the set of negatively permitted actions and the set of exemptions are disjoint:

**Theorem 5.24.** *If  $(a, b)$  is negatively permitted then it is not an exemption.*

**Proof.** If  $(a, b)$  is negatively permitted, then  $(a, \neg b) \notin out(G)$ , so  $(a, b)$  is not an exemption.

□

This is as it should be, since it allows us to distinguish sharply between, on the one hand, those actions that a legislature or other norm-issuing authority can prohibit *at its discretion*, and, on the other hand, those actions that may require the prior retraction of an intentionally granted permission. I. e. the disjointness of these two classes of action is necessary to preserve the priority ordering as described in section 2. Antithetic permissions, on the other hand, can clearly be negatively permitted, since their local negations need not be derivable from the code:

**Theorem 5.25.** *If  $(a, b)$  is antithetically permitted according to  $\langle G, P \rangle$ , but not an exemption according to the same code, then it is negatively permitted.*

**Proof.** Suppose  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$ , but not an exemption. We need to establish that  $(a, \neg b) \notin out(G)$ . Theorem 5.18 tells us that  $(a, b)$  is an exemption in  $\langle G \cup \{(a, \neg b)\}, P \rangle$ . Thus if  $(a, \neg b) \in out(G)$  then  $\langle G \cup \{(a, \neg b)\}, P \rangle = \langle G, P \rangle$ , so  $(a, b)$  is an exemption in  $\langle G, P \rangle$  contrary to assumption.

□

Before bringing this paper to a close we should return to the problem of how to compute the permissions that can be said to be implicit in a code or policy specification. This problem was briefly discussed in the introduction, and we are now in position to give it a tentative solution: By definition,  $(a, b)$  is antithetically permitted in  $\langle G, P \rangle$  only if  $(a, \neg b)$  contradicts an explicit permission *or* an exemption in the same code. It is possible to eliminate the latter disjunct and to show that  $(a, \neg b)$  must ultimately contradict an explicit permission in  $P$ . The proof relies on the representation of antithetic permission given by theorem 5.16:

**Theorem 5.26.** *If  $(a, \neg b \rightarrow \neg d) \in out(G)$  for some exemption  $(c, d)$  with  $a \equiv c$ , then  $(a, \neg b \rightarrow \neg h) \in out(G)$  for some explicit permission  $(g, h) \in P$  with  $g \equiv a$ .*

**Proof.** Suppose  $(c, d)$  is an exemption according to  $\langle G, P \rangle$  and that  $c \equiv a$ . Then  $(c, \neg d) \in out(G) \setminus out(G) - (g, \neg h)$  for some  $(g, h) \in P$  with  $g \equiv c$ . It

follows that  $(c, \neg d) \notin H$  for some  $H \in out(G) \perp (g, \neg h)$ , whence  $(g, \neg h) \in out(H \cup \{(c, \neg d)\})$  by the maximality of  $H$  so  $(a, \neg d \rightarrow \neg h) \in H \subseteq out(G)$ , by lemma 3.3, *SI* and the assumption that  $a \equiv c \equiv g$ . Hence, since by assumption  $(a, \neg b \rightarrow \neg d) \in out(G)$ , we have  $(a, \neg b \rightarrow \neg h) \in out(G)$  by *AND* and *WO*. Since  $(g, h) \in P$  and  $g \equiv a$  therefore, the proof is complete.

□

Taken together with theorem 5.21 and 5.23 this gives us a way to represent exemptions that does not appeal to derogation at all:

**Theorem 5.27.**  $(a, b)$  is an exemption according to  $\langle G, P \rangle$  iff  $(a, \neg b) \in out(G)$  and  $(a, \neg b \rightarrow \neg d) \in out(G)$  for some  $(c, d) \in P$  with  $c \equiv a$ .

**Proof.** For the left-to-right direction, suppose  $(a, b)$  is an exemption in  $\langle G, P \rangle$ . Then  $(a, \neg b) \in out(G)$ . By theorem 5.21,  $(a, b)$  is antithetically permitted according to the same code. I. e.  $(a, \neg b \rightarrow \neg c) \in out(G)$  for some exemption or explicit permission  $(a, c)$ . Hence, by lemma 5.26, it follows that  $(a, \neg b \rightarrow d) \in out(G)$  for some  $(a, d) \in P$ , as desired. For the converse direction, suppose  $(a, \neg b) \in out(G)$  and  $(a, \neg b \rightarrow \neg d) \in out(G)$  for some  $(a, d) \in P$ . Then  $(a, b)$  is antithetically permitted by theorem 5.16. Since  $(a, \neg b) \in out(G)$ , it follows by lemma 5.23, that  $(a, b)$  is an exemption in  $\langle G, P \rangle$ , so we are done.

□

This in turn yields a uniform procedure for calculating exemptions and antithetic permissions (explicit permission being immediately given by  $P$ , of course). To check whether  $(a, b)$  is an exemption according to  $\langle G, P \rangle$  do:

```

If (a, -b) in out(G):
    For c in P(a):
        If (a, -b -> -c) in out(G):
            return 1
    return 0
else:
    return 0

```

To check whether  $(a, b)$  is antithetically permitted, on the other hand, it suffices, by theorem 5.18 and 5.19, to check whether it is exempted if added to the code  $G$ . Of course this means that the initial membership test is superfluous, so the procedure is simply;

```
For c in P(a):
  If (a, -b -> -c) in out(G U (a, -b)):
    return 1
return 0
```

## 6. Summary and conclusion

I have proposed a new analysis of the concept of positive permission, understood as that which is implied by a set of explicitly pronounced permissions. Positive permission has been analysed as constraints on the generation of output from a code. As such they naturally assume priority over the mandatory norms they override, which in turn take priority over negatively permitted actions. This is all in conformity with the desiderata presented in section 2.

The analysis shows that there is a close relationship between explicit permission and exemptions. Exemptions are antithetically permitted, and antithetic permissions are exemptions in a larger code. Moreover, exemptions are characterisable as the set of antithetic permissions whose local negations are included in the code. As regards the concept of negative permission, it is more loosely coupled with the other two than most other accounts will have it. Antithetic permissions are negative permissions if they are not also exemptions. But exemptions, on the other hand, are never negative permissions, that is, the class of negative permissions and the class of exemptions are disjoint. I have argued that this is as it should be, since it allows us to distinguish between, on the one hand, those actions that a legislature can prohibit at its discretion, and those that are protected by a permissive provision, on the other. In other words, the disjointness of these two classes of actions reflects their relative positions in the priority ordering of normative concepts.

Let  $EX(G, P)$  be the set of exemptions in  $\langle G, P \rangle$ , and let  $ANT(G, P)$  and  $NEG(G)$  be the set of antithetic permissions and negative permissions respectively. For any set of norms  $S$ , let  $S' := \{(a, \neg b) : (a, b) \in S\}$ . The result of the preceding investigations may then be summarised as follows:

1.  $EX(G, P)' \subseteq out(G)$
2.  $ANT(G, P) \setminus EX(G, P) \subseteq NEG(G)$
3.  $EX(G, P) \subseteq ANT(G, P)$
4.  $P \subseteq ANT(G, P)$  for any  $G$ .
5.  $ANT(G, P) \cap out(G)' \subseteq EX(G, P)$
6.  $EX(G, P) \cap NEG(G, P) = \emptyset$
7.  $ANT(G, P) \subseteq EX(G \cup ANT(G, P), P)$

1 follows immediately from the definition of exemptions, 2 is theorem 5.25, 3 is theorem 5.22, 4 is theorem 5.21, 5 is theorem 5.23, 6 is theorem 5.24 and 7 is theorem 5.18.

None of the results I have presented are difficult, and the theory that emerges is simple. I consider this one of its principal virtues.

**Acknowledgement.** This research was partially funded by the Semicolon project supported by the Norwegian Research Council, contract no 183260.

## References

- [1] M. Abadi: “Variations in access control logic”, in *Proceedings of the Ninth International Conference on Deontic Logic in Computer Science (DEON 2008)*, pp. 96–109. Springer-Verlag, 2008.
- [2] M. Abadi, M. Burrows, B. Lampson and G. Plotkin: “A calculus for access control in distributed systems”, in *ACM Transactions on Programming Languages and Systems*, vol. 15(4). pp. 706–734, 1993.
- [3] C. E. Alchourron, P. Gärdenfors and D. C. Makinson: “On the logic of theory change: Partial meet contraction and revision functions”, in *Journal of Symbolic Logic* vol. 50(2), pp. 510–530, 1985.
- [4] C. E. Alchourron and E. Bulygin: “Permission and permissive norms”, in *Theorie der Normen*. Dunker and Humbolt, Berlin 1984.

- [5] C. E. Alchourron and E. Bulygin: “The expressive conception of norms”, in Hilpinen (ed.): *New Studies in Deontic Logic*. D. Reidel Publishing Company, 1981.
- [6] M. A. Brown: ”Conditional obligation and positive permission for agents in time”, in *Nordic Journal of Philosophical Logic* vol. 5(2), pp. 83–111, 2000.
- [7] C. E. Alchourron and E. Bulygin; *Normative Systems*. Springer, 1971.
- [8] M. Atienza and J. R. Manero: *A Theory of Legal Sentences*. Kluwer Academic Publishers, 1998.
- [9] E. Bulygin: “Norms and logic”, in *Law and Philosophy*, vol. 4(2), pp. 145–163, 1985.
- [10] T. Campbell: *Rights: A Critical Introduction*. Routledge, 2006.
- [11] B. F. Chellas: *Modal Logic: An Introduction*. Cambridge University Press, 1980.
- [12] D. Ferraiolo and R. Kuhn: “Role-based access control”, in *Proceedings of the 15th National Computer Security Conference*, 1992
- [13] S. O. Hansson: ”Reversing the Levi identity”, in *Journal of Philosophical Logic*, vol. 22(6), pp. 637–669, 1993.
- [14] J. L. Hart: *The Concept of Law*. 2nd edition. Oxford UNiversity Press, 1997.
- [15] H. Kelsen: *General Theory of Norms*. Oxford: Clarendon Press, 1991.
- [16] D. Lewis: “A problem about permission”, in *Essays in Honour of Jaakko Hintikka*. D. Reidel Dodrecht, Boston-London, 1979.
- [17] D. C. Makinson: “On a fundamental problem of deontic logic”, in P. MacNamara and H. Prakken (eds.) *Norms, Logic and Information Systems. New Studies in Deontic Logic and Computer Science*. IOS Press, 1999.
- [18] D. C. Makinson and L. van der Torre: “What is input/output logic”, in *Foundations of the Formal Sciences II: Applications of Mathematical Logic in Philosophy and Linguistics* vol. 17, pp. 163–174, 2003.

- [19] D. C. Makinson and L. van der Torre: “Constraints for input/output logics”, in *Journal of Philosophical Logic*, vol. 30, pp. 155–185, 2001.
- [20] D. C. Makinson and L. van der Torre: “Input/output logics”, in *Journal of Philosophical Logic* vol. 29, pp. 383–408, 2000.
- [21] D. C. Makinson and L. van der Torre: “Permission from an input/output perspective”, in *Journal of Philosophical Logic*, vol. 32(4), pp. 391–416, 2003.
- [22] L. McCarty: “Permissions and obligations”, in *Proceedings IJCAI-83*, pp. 287–294, 1983.
- [23] R. van der Meyden: “A dynamic logic of permission”, in *Journal of Logic and Computation* vol. 6(3), pp. 465–479, 1996.
- [24] K. Opalek and J. Wolenski: “On strong and weak permission”, in *Rechtstheorie* vol. 4, pp. 169–182, 1973.
- [25] K. Opalek and J. Wolenski: “Normative systems, permission and deontic logic”, in *Ratio Juris* vol. 4(3), pp. 334–348, 1991.
- [26] J. Raz: *Practical Reason and Norms*. Oxford University Press, 1990.
- [27] A. Ross: *Directives and Norms*. Routledge and Kegan Paul, 1968.
- [28] R. Sandhu, D. F. Ferraiolo and D. R. Kuhn: “The NIST model for role-based access control”, in *Proceedings of the 5th ACM Workshop on Role-Based Access Control*, 2000.
- [29] A. Stolpe: *Norms and Norm-System Dynamics*. PhD thesis. Department of Philosophy, University of Bergen, Norway. 2008.
- [30] Y.-H. Tan and W. Thon: “A logical model of directed obligations and permissions to support electronic commerce”, in *Intl. J. Electronic Commerce* vol. 3(2), pp. 87–104, 1998.
- [31] L. van der Torre and G. Boella: “Permissions and obligations in hierarchical normative systems”, in *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, 2003.

- [32] Y. B. Udipi and M. P. Singh: “Multiagent policy architecture for virtual business organizations”, in *Proceedings of the IEEE International Conference on Services Computing (SCC)*, 2006.
- [33] O. Weinberger: “The expressive conception of norms—an impasse for the logic of norms”, in *Law and Philosophy*, vol. 4(2), pp. 165–198, 1985.
- [34] O. Weinberger: “Normenlogik und logische Bereiche”, in A. G. Conte, R. Hilpinen and G.H. Henrik von Wright (eds.) *Deontische Logik und Semantik*. Wiesbaden, 1977.
- [35] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler and G. J. Sussman: “Information accountability”, in *Communications of the ACM*, vol. 51(6), 2008.
- [36] G. H. vonWright: *Norm and Action*. Routledge & Kegan Paul, 1963.
- [37] G. H. von Wright: “Deontic Logic”, in *Mind* vol. 60, pp. 1–15, 1951.