Relevance, derogation and permission:
A case for a normal form for codes of norms

Audun Stolpe

Institute of Informatics, University of Oslo, Norway. E-mail: audus@ifi.uio.no

Abstract. We show that a recently developed theory of positive permission based on the notion of derogation is hampered by a triviality result that indicates a problem with the underlying full-meet contraction operation. We suggest a solution that presupposes a particular normal form for codes of norms, adapted from the theory of relevance through propositional letter sharing. We then establish a correspondence between contractions on sets of norms in input/output logic (derogations), and AGM-style contractions on sets of formulae, and use it as a bridge to migrate results on propositional relevance from the latter to the former idiom. Changing the concept accordingly we show that positive permission now incorporates a relevance requirement that wards off triviality.

Key words: Positive permission, antithetic permission, derogation, relevance, norm-system dynamics.

1 Introduction

In computer science, permission is perhaps the most ubiquitous of the so-called deontic concepts—although it often appears under different names such as privilege or access rights. A need for a rigorous definition of the concept arises for instance in access control logic [1] and information accountability [20]. Permission is a very subtle and many-faceted concept. In fact it denotes a rag bag of distinct but closely related action concepts that should be analysed in interconnection.

A unifying theory was proposed in [19], a prominent feature of which is that it shows how a single code of norms generates different categories of permission when it is manipulated in different ways that correspond roughly to natural operations on e.g. legal corpora. The operation of *derogation* in particular turns out to have considerable explanatory force, as all concepts of positive permission, so called, are analysable as ways of overriding a prohibitive norm. The resulting theory defines and interrelates the concept of *explicit permission* and the concept of *negative permission*, as well as two kinds of implicit positive permission, namely *exemption* and *antithetic permission*.

In this paper we shall focus on the latter two. We shall refer to both by the generic term *positive permission* when finer distinctions do not matter. As a tentative characterisation we may say that an *exemption* is an action that is implicitly recognized by the promulgating authority as constituting an exception

to a general prohibition.[1] 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. This regulation could have been worded differently of course, but it seems clear that the intent in any case is to make participation in public performances an exception to the prohibition.

The second form of implied positive permission may be called *antithetic permission*, since the idea is to see a norm (a, b) as permitted by a code G whenever, given the obligations already present in G, we can't forbid b under the condition a without thereby *contradicting* something d that is implicit in what has been explicitly permitted [13, p. 398]. Unlike exemption, the primary function of antithetic permissions is not to suspend an existing prohibition, but rather to prevent one from being passed in the first place. The paradigm example—but by no means the only one—is an action protected by constitutional law, for instance freedom of expression. We shall stick to this general idea, although the definition will differ slightly from that of [13, p. 398].

At the heart of the present paper is a triviality result that shows the following: According to the theory set out in [19], if an action described by b constitutes an exemption in circumstances described by a, according to a code G, then any arbitrarily chosen c is antithetically permitted in the same circumstances according to the same code G. This anomaly may be diagnosed in more than one way, but the line pursued in this paper is to consider it a problem of *relevance*. We do not view this triviality result as a refutation of the general idea behind the analysis of antithetic permission in [19]. Rather we view it as an indication that the well-behavedness of a code under revisions depends also on the form of that code, not just its content. Consequently, we require of a code that it be on what we shall call *right-splitting form*, whereby only those prohibitions that, in a sense to be made precise, are *relevantly* related to a permissive norm will be overridden by it. This take on the problem is inspired by parallell developments in the field of belief revision, particularly [16] and [15].

The remainder of this paper is organized as follows: Section 2 gives a bare-bones account of input/output logic, our idiom of choice. Section 3 recalls the main features of the theory of permission as set out in [19], and proves the aforementioned triviality result. Section 4 recalls the relationship between relevance through propositional letter-sharing and belief revision. Finally, section 5 establishes a correspondence between contraction on sets of sentences and derogation (as we choose to understand that term), that allows us to migrate the results from section 4 to input/output logic, and in turn to restore the concept of antithetic permission to the status of an informative concept.

[1] A similar idea was put forth in [4]. See section 7 for a comment on the differences.

2 Preliminaries

A few notational preliminaries first: We use lower case letters $a, b, c...$ to range over formulae of classical propositional logic, denoted L. The distinguished formula t will stand for an arbitrary tautology, and f for an arbitrary contradiction. Sets of formulae are denoted by upper case letters from A to D (we shall never need more), and $E(A)$ denotes the set of elementary letters that occur in A. Upper case letters from F to I denote sets of norms, that is, binary relations over subsets of L. When $G \subseteq L \times L$ we denote its pre-image under L as G^1 and its image under L as G^2. Image-formation will be denoted by ordinary parentheses, for instance $(G \cup H)(a)$ denotes the image of the relation $G \cup H$ under a. Classical consequence is written with a turnstile $\vdash$ when considered as a relation over $2^L \times L$, and as Cn when viewed as an operation on 2^L onto itself. To make the notation less verbose, we follow the convention of writing $A \cup a$ instead of $A \cup \{a\}$, and similarly for norms.

We shall stick to the general ideas of [19] and analyse exemption and antithetic permission in terms of derogation. Hence we need to work with *codes* of norms. Individual norms, permissive as well as mandatory, will emerge from the overall behaviour of the system—they cannot be analysed in isolation. This is true in particular of permissive norms, which, to the extent that they do anything at all, act as constraints on the generation of mandatary norms and therefore presuppose the existence of such.

Now, 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 (*prima facie*) mandatory norms that contains all norms it entails (by some as yet unspecified notion of entailment). This is abstract, true, but will do fine for our purposes. 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 ([12], [11] and [10]).

In input/output logic a (*prima facie*) norm is simply a pair (a, b) correlating an *applicability condition*, *trigger* or *input* a with a *duty*, *optimality condition* or *output* b—these will sometimes be denoted neutrally as the antecedent and consequent of a norm respectively. The correlation between the antecedent and consequent 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. A *code* of norms in input/output logic is simply a set G of such pairs, from which 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 fundamental 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 fundamental model of a normative system is an operation *out* of type $2^{L\times L} \times L \mapsto 2^L$ defined as follows:

Definition 1. $out(G, a) = Cn(G(Cn(a)))$[2]

A syntactic representation of this operator, can be obtained by putting $(a, b) \in out(G)$ iff $b \in out(G, a)$. 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 the relation which is the value of (the monadic) *out* for some argument G—it is characterised by the following system of inference rules [10, Observation 1]):

Definition 2. $(a, b) \in deriv(G)$ *iff* (a, b) *is derivable from axioms* $(t, t) \cup G$ *by the rules of inference,*

$$SI\ \frac{(c,b)}{(a,b)}\quad \textit{if } a \vdash c \quad AND\ \frac{(a,b),(a,c)}{(a,b\wedge c)} \quad WO\ \frac{(a,b)}{(a,c)} \quad \textit{if } b \vdash c$$

We shall switch between operator notation and turnstile notation for derivability from G whenever we find it convenient, writing $(a, b) \in deriv(G)$ or $G \vdash_G (a, b)$ as we see fit. Note, as we shall have occasion to appeal to these properties later, that *out* and *deriv*, considered as operators on relations, are closure operators.

The remainder of this section records a few properties that we shall be needing later:

Lemma 1. $out(G)(a) = out(G, a) = out(G)(Cn(a))$

Proof. We prove the first equality only, the seond is essentially similar. Suppose $b \in out(G, a)$. Then $(a, b) \in out(G)$, whence, by definition, $b \in out(G, a)$. For the converse inclusion suppose $b \in out(G, a) = Cn(G(Cn(a)))$. By compactness for classical consequence there is a finite set of norms $(a_1, b_1), \ldots (a_n, b_n) \in G$ such that $a \vdash a_i$ for $1 \leq i \leq n$ and $\bigwedge_{i=1}^{n} b_i \vdash b$. Applying SI, AND and WO repeatedly we have $(a, b) \in out(G)$, whence $b \in out(G, a)$ as desired.

Next we have:

Lemma 2 (Easy half of conditionalization). *If* $(a \wedge c, b \to d) \in out(G)$ *and* $c \vdash a$ *then* $(c, d) \in out(G \cup (a, b))$.

Lemma 3 (Hard half of conditionalization). *If* $(c, d) \in out(G \cup (a, b))$ *then* $(a \wedge c, b \to d) \in out(G)$.

Lemma 4. *If* $(c, d) \in out(F \cup (a, b)) \setminus out(F)$ *then* $c \vdash a$.

[2] This should perhaps rather be called 'a system of *prima facie* norms', since it does not deal with contrary-to-duty conditionals. It relates, one might say, to normative reasoning as classical logic does to commonsense reasoning.

The proofs of these lemmata can be found in [19].

The reader should note that the *out*-operator considered here corresponds to the one Makinson and van der Torre call simple-minded output [10]. From the point of view of the philosophy of norms, certainly, it is the least interesting of the input/output operators. Nevertheless, simple-minded output differs from the other operators in the input/output family in that it can be viewed as a most natural and immediate generalization of classical logic. Classical logic is simply the special case where the set of norms G is the diagonal relation over L. Hence, simple-minded output is a generalization of classical logic that amounts to dropping the reflexivity property for material conditionals. This has the effect of isolating the heads from the bodies of the pairs, so information about the one cannot be carried backwards or forwards to the other. This, in our opinion, is an essential feature of (*prima facie*) norms. Norms are stipulations. Whatever holds according to a norm holds, one might say, by *conventional generation*. Indeed there may be warrant for speaking about *non-reflexive logics* as on a par with *non-monotonic logics*. The latter denotes a broad range of inference patterns that in some way or other involve assumptions of falsity, which in turn make arguments sensitive to the absence of information. The former may in a similar spirit be taken to stand for a family of inference patterns that share the characteristic that they are mediated by convention. Examples include permissive norms, mandatory norms, norms of etiquette and constitutive norms (which should also not be construed as reflexive). Simple-minded output has the virtue that it isolates this feature in its purest form, and makes it easy to track the things that change. Of course, the operative assumption is that we will learn something useful that we can apply to more sophisticated versions later.

3 Permission as derogation

In what may be considered the principal case of positive permission, an implied positive permission suspends a general prohibition that is *already* in force, that is, it constitutes an exception to 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,
> .....

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 suspended. These cases are in effect exempted from the ban, and therefore constitute permissions. Thus, 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*,[3] 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 we see little reason to deviate from it.[4]

Definition 3 (Remainders). *$G \perp (a, b)$ is the set of H such that 1) $H \subseteq G$, 2) $(a, b) \notin out(H)$, and 3) If $H \subset I \subseteq G$ then $(a, b) \in out(I)$*

This definition is neutral to the question of whether G is an open or a closed set (modulo *out*). In the latter case we have the following property:

Lemma 5. *$H = out(H)$ if $(a, b) \in G$, $H \in G \perp (a, b)$ and $G = out(G)$.*

Proof. This is is [19, lemma 5.2]:

Indeed this property can easily be verified to hold for *any* closure operator. It holds in particular for classical consequence Cn. In what follows we shall therefore allow ourselves to overload the operator $\perp$, using it to denote remainders of sets of norms as well as remainders of sets of formulae (the arguments to $\perp$ will generally suffice for disambiguation). Similarly we shall let the '$-$' operator do dual service both as a *contraction operator* and as a *derogation operator*:

Definition 4.

1. $G - (a, b) := \bigcap(G \perp (a, b))$
2. $A - a := \bigcap(A \perp a)$

Note that this definitional pattern accomodates both the *base* and *theory* versions, so called, of derogation and contraction, depending on whether or not the

[3] Stated differently, we understand derogation in terms of *exception* or *defeat*. See e.g. [9] for a similar interpretation.

[4] The same paradigm of norm-system change was independently proposed in [4] and [18]. Both with input/output logic as their point of departure. They differ in that former adopts an axiomatic approach whereas the latter proceeds constructively. Another source that proceeds constructively is [9]. It is, however, based on a defeasible logic rather than on input/output logic. The ramifications of these differences remain to be explored.

set in question is closed under the corresponding consequence operation. Note also that both operators are full-meet.

Now, since permissions will be analysed in the larger context of a system, the proper unit of analysis is a *code* $\langle G, P\rangle$ consisting of a set of *explicitly stated mandatory norms* G and a set of *explicitly stated permissive norms* P. Sometimes, for brevity, P will be referred to 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, we 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, as the case may be, by that code whenever a is *true*. As observed by two of the reviewers, we do not give the mechanism to compute the closure of the code $\langle G, P\rangle$ as such. While this is indeed something that needs to be adressed at some point, we believe that the interplay of mandatory and permissive norms needs to be, and can be, studied first.

Now, recall that exemptions are essentially exemptions *from* something. That is, an exemption always relates to a background prohibition. In other words, the permissive provision acts as a *defeater* in relation to *prima facie* applicable law. This suggests the following definition, which was proposed in [19]:

Definition 5 (Exemptions). (a, b) *is an exemption according to the code* $\langle G, P\rangle$ *iff* $(a, \neg b) \in out(G) \setminus out(G) - (c, \neg d)$ *for some* $(c, d) \in P$ *such that* $c \equiv a$.

Thus, (a, b) is an exemption if a prohibition $(a, \neg b)$ can be derived from the code, but this prohibition is overriden by an explicit permissive norm (c, d). We 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 norms in G entail 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. To see how this concept behaves, consider the following example:

Example 1. Put $G := \{(t, \neg p)\}$ and $P := \{(c, p)\}$. Think of these norms as a general prohibition against processing personal information and as a permission for the case of consent respectively. We have $(t, \neg p) \in out(G)$, whence processing personal information is in general prohibited (that is, it is prohibited in the absence of information to the contrary). Note that the norm $(t, \neg p)$ covers all cases. In particular it covers the case of consent since $(c, \neg p) \in out(G)$ by SI. However, $(c, \neg p) \notin out(G) - (c, \neg p)$, so (c, p) constitutes an exemption.

Exemptions, as so defined, have several interesting properties, for instance:

Proposition 1 (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* $b \vdash c$.

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:

Proposition 2 (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.*

On the negative side, exemptions do not satisfy *input weakening*, i. e. one cannot infer an exemption $(a \vee c, b)$ from the existence of an exemption (a, b). This is as it should be. Consider again §8 of the Personal Information Act: If we were to endorse input weakening, then we would have to conclude that processing personal information is permitted if the registered person is simply unable to consent. Thus, shooting him would be one way of obtaining permission to access his information. Hence, exemptions should not satisfy input weakening.

Nor should they satisfy input *strengthening*, actually, because 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 §11*a*: '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. This strongly suggests that permissive norms should be regarded as *classical with respect to the antecedent* in the terminology of [6].

Classicality wrt. the antecedent is implement in definition 5 by the requirement that antecedents of exemptions be equivalent to the antecedent of some explicitly stated permissive provision. No such requirement is placed on consequents, so exemptions are classical with respect to the antecedent and *normal* with respect to the consequent. However, a complete characterisation remains an open problem.

Having the set of exemptions under reasonable control we are now in position to say something substantial about antithetic permission. The idea, as put in words by Makinson and van der Torre (who in turn give Alchourrón 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 comitting ourselves to forbid, under a condition c that could possibly be fulfilled, something d which is implicit in what has been explicitly permitted. Another way to put it is to say that antithetic permissions prevent the set of mandatory norms from growing in such a way as to render explicitly permitted actions forbidden. This checked-growth perspective may be expressed as follows:

Definition 6. (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* $a \equiv c$.

That is, (a, b) is antithetically permitted in $\langle G, P\rangle$ if it would be anulled by an explicit permission (c, d) in P were it to be *added* to the code G. As with exemptions, we shall say that (a, b) is antithetically permitted *by* (c, d). Antithetic permission as so defined does not coincide with the concept of exemption, but

there is obviously a quite close relationship between them. For one, the latter class is subsumed by the former:

Theorem 1. *If (a, b) is an exemption in $\langle G, P\rangle$ then (a, b) is antithetically permitted in the same code.*

Proof. This is [19, theorem 5.21]:

As regards the difference between definition 6 and the definition of antithetic permission in [13] (or as it is called, *dynamic positive permission*), it is mainly this: The latter only requires that $G \cup (a, \neg b)$ entail a norm that contradicts a positive permission. It is, in other words, closer to the following:

Definition 7. *(a, b) is dynamically permitted according to $\langle G, P\rangle$ iff $(c, d) \in out(G \cup (a, \neg b))$ for some $(c, d) \in P$ with $a \equiv c$.*

We call it *dynamic* permission to underscore its affinity with the corresponding definition in [13], of which it is a straightforward translation with minor modifications. Now, unlike 7, definition 6 requires that $(a, \neg b)$ be actually *overriden* by (c, d) if added to the code. This difference is subtle. According to definition 6 (a, b) is not antithetically permitted if $(a, \neg b)$ is deemed more worthy of retaining than other norms already in the code. When the derogation operation is full-meet there is no difference between the two—definition 6 and 7 are equivalent. However, they come apart in the general case: There are *partial meet* derogation operators capable of distinguishing between the two. Thus another way to express the difference between them is to say that definition 6 but not 7 is sensitive to the priority structure of the code, as expressed e.g. by a selection function on remainders. Definition 6 does not deem (a, b) antithetically permitted just because the addition of $(a, \neg b)$ would violate a positively permitted norm, for there may be ways of restoring compliance that do not require the exclusion of $(a, \neg b)$. One consequence of this is that the relationship between antithetic permissions and exemptions becomes more stable and regular, as witnessed by the following theorem:

Theorem 2. *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. This is [19, theorem 5.19]

In other words antithetic permissions are exemptions in a larger code. This agrees well with intuition, and also finds support in the literature:

> 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 [2, pp. 397–398].

Theorem 2 spells 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 Alchourrón and Bulygin put it.

Now, as it turns out, definition 6 admits an equivalent representation:

Theorem 3. *(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. This is [19, theorem 5.16].

Alas, we can now prove that the existence of an exemption makes everything antithetically permitted under the conditions of the exemption. This is the announced triviality result:

Theorem 4. *If (a, b) is an exemption according to $\langle G, P\rangle$, then (a, e) is antithetically permitted in $\langle G, P\rangle$ for an arbitrarily chosen e.*

Proof. If (a, b) is an exemption in $\langle G, P\rangle$, then $(a, \neg b) \in out(G) \setminus out(G) - (c, \neg d)$ for $(c, d) \in P$ with $a \equiv c$. There is thus an $F \in out(G) \perp (c, \neg d)$ with $(a, \neg b) \notin F$, which, since $(a, \neg b) \in out(G)$ means that $(c, \neg d) \in out(F \cup (a, \neg b))$. Hence, it follows by lemma 3 that $(a, \neg b \rightarrow \neg d) \in out(G)$. Now, $(a, \neg b) \in out(G)$ by assumption, and we have $(a, \neg b \rightarrow (\neg e \rightarrow \neg b))$ from (t, t) by SI, whence $(a, \neg e \rightarrow \neg b)$ by AND. Taking stock we have $(a, \neg e \rightarrow \neg b)$, $(a, \neg b \rightarrow \neg d) \in out(G)$, so $(a, \neg e \rightarrow \neg d) \in out(G)$ by AND and WO.

Note that this result has general import as it shows that full meet theory contraction on sets of norms is very heavy handed—as one would expect. As regards the concept of permision, one response would be to climb up the ladder of generality and introduce a selection function on remainders. However, in this paper we have chosen the road less travelled by: We shall require of positive permissions that they be *relevant* to the explicit provisions from which they follow. As we shall see, relevance will counterbalance the negative impact of the more notorious quirks of material implication, thereby restoring the concept of antithetic permission to the status of a genuinely informative concept.

4 Relevance and contraction: A brief recapitulation

The present section lists the most important definitions and results of the theory of relevance through propositional letter sharing as developed in, among others, [17] and [16]. This material is included for expository completeness—in order to make the paper reasonably self-contained. We lay no claim to originality, and the reader should consult the cited papers for details and proofs.

Consider any two formulae a and b. As a first shot, one may consider them relevant to each other iff they share an elementary letter. It is well known, however, that this definition makes the concept of relevance highly syntax-dependent (see [16]). Indeed, by this simple criterion every formula c will be relevant to any other

formula d under some logically equivalent representation, for we may choose to work with c in the form $c \vee (c \wedge d)$.

Syntax-dependence may be overcome by comparing, not the formulae *simpliciter*, but rather a least letter-set representation of them. The least letter-set theorem was proved in the general case in [14], and says that for any contingent set A of formulae, finite or infinite, there is a unique least set of elementary letters such that A can be equivalently expressed using only letters from that set. We shall follow [16] in using a choice function $*$ to select least letter-set forms. Thus A^* will denote a least letter-set representation of A and similarly for formulae. Now, to eliminate syntax-dependence from the (or rather *this*) concept of relevance one deems a and b relevant to each other iff a^* and b^* have an elementary letter in common. It is not difficult to show that this solves the problem of syntax-dependence as it has been stated thus far.

However, consider now any three distinct elementary letters a, b and c. Since they do not share any letters they are not relevant to each other by either of the above criteria. But, if we take into consideration the set $A := \{a \to b, b \wedge c\}$ (the particular form of these sentences doesn't matter, only what letters they contain), then it seems natural to say, since a is relevant to b and b to c, that a is relevant to c *modulo* A. This is a more general notion as relevance is now mediated by a background theory (which is undoubtedly more realistic). The challenge though is how to avoid reintroducing syntax-dependence, this time in the background theory. A solution was provided by Parikh [17], who came up with the ingenuous concept of a *splitting*:

Definition 8 (Splitting). *Let A be a contingent set of formulae and let $\mathbf{E} = \{E_i\}_{i \in I}$ be a partition of its least-letter set. Then $\mathbf{E}$ is a splitting of A iff there is a family $\{B_i\}_{i \in I}$ of sets of formulae such that $E(B_i) \subseteq E_i$ and $A \equiv \bigcup\{B_i\}_{i \in I}$.*[5]

Note that $\mathbf{E}$ is a special kind of partition of the least letter set of A, it is not a splitting of A itself. Moreover, it is not required that $B_i \subseteq A$ as long as the union of all B_i is equivalent to A. A splitting is said to be *finer* than another if each cell of the former is included in some cell of the latter. We have:

Theorem 5. *Every contingent set A of formulae has a unique finest splitting.*

Proof. This is [15, theorem 2.4].

The finest splitting theorem tells us that any set of formulae has a finest representation as a family of letter disjoint sets [15, p. 994]. Although it is the uniqueness of the partition $\mathbf{E} = \{E_i\}_{i \in I}$ that is asserted, it turns out that the associated B_i are unique too, up to logical equivalence [16, observation 3.2], so we may abuse notation a bit by also calling $\bigcup\{B_i\}_{i \in I}$ the finest splitting of A.

[5] This definition may be extended to cover the limiting case where A is inconsistent or tautologous, but at the cost of limiting-case clauses in definitions, theorems and proofs [16, p. 380]. We shall therefore consider only contingent sets henceforth. The reader should bear in mind though, whenever this proviso occurs, that it is not a real restriction.

We shall follow [16] in writing $A^{\#}$ to denote both, relying on context to disambiguate. Parikh's relevance criterion is formulated in terms of finest splittings as follows:

Definition 9 (Cell-relevance modulo A). *Let $\mathbf{E} := \{E_i\}_{i\in I}$ be the finest splitting of a contingent set A. We say that a is cell-relevant to b modulo A iff either $E(a^*) \cap E(b^*) \neq \emptyset$ or $E(a^*) \cap E_i \neq \emptyset \neq E(b^*) \cap E_i$ for some $i \in I$.*

This definition exploits the fact that finest splittings, unlike least letter-sets, not only minimise the set of elementary letters, they also *disentagle* them, as far as possible. For instance, $\{a \wedge b\}$ is on least letter-set form but not on finest splitting form, which is $\{\{a\}, \{b\}\}$. The finest splitting of A, one might say, chops the logical content of A up into cells that encapsulate its least logically independent parts. One may therefore deem two formulae relevant to each other, modulo a given set, if they both share some letter (not necessarily the same) with a cell in the splitting, because they can then not be independent of each other 'in the eyes' of that set. This makes finest splittings behave rather more predictably under revisions than sets in general. Indeed there is an intimate and interesting relationship between finest splittings and contractions. What it amounts to is this: If contractions are performed on $A^{\#}$ instead of A, then, despite the classical equivalence of A and $A^{\#}$, $A^{\#} - a$ only removes elements that are cell-relevant (henceforth simply 'relevant') to a modulo A, whereas $A - a$ does not. The reason for this is that when $A^{\#} - a$ is the result of a partial meet contraction of a from $A^{\#}$, then it is the intersection of some family of maximal a-nonimplying subsets of $A^{\#}$. But if there is no cell E_i of the splitting that contains letters from both a and a formula $b \in A^{\#}$, then the addition of b to any a-nonimplying subset of $A^{\#}$ will leave it a-nonimplying, so that b must be in all the maximal a-nonimplying subsets of $A^{\#}$, and so in $A^{\#} - a$ [15, p. 1000]. As a result we have:

Theorem 6. *If $A^{\#} \vdash a$ and $A^{\#} - b \vdash a$ then a is relevant to b modulo A.*

Proof. This is [15, theorem 4.1].

What has all this got to do with permission? Well, as we have defined the concept, a positive permission is essentially a putting out of play of a contrary-prohibition effected by means of a contraction on the set of norms. If we could generalize theorem 6 to input/output logic, perhaps we could get the permission concepts to behave more regularly, and thereby avoid the triviality result. That is the ambition.

5 Bringing relevance to bear

For any $H \subseteq L \times L$ there is more than one way to define a splitting, as we may split on one or both sides of the relation. That is, we may construct the splitting of H from a splitting of either of H^1, H^2 or both. For reasons that will be explained shortly, we shall choose the second option:

Definition 10 (Right-splitting). *Let G be any subset of $L \times L$ such that G^2 is consistent. Then the right-splitting of G, denoted $G^{\#}_{\rightarrow}$, is the set*

$$\bigcup\{\{a\} \times G(a)^{\#} : a \in G^1\}$$

Right-splittings are well defined since G^2 is consistent. For our purposes there is no need to split on the left. The concepts we wish to charaterise, i.e. the permission concepts, are only defined for norms with logically equivalent applicability conditions, whence no question of relevance arises on the input side. Right splittings have two properties that are crucial for our purposes. We set them out as separate lemmata:

Lemma 6. *Assume $G = out(G)$ and that G^2 is consistent. If $G^{\#}_{\rightarrow}(a) \neq \emptyset$ then $G^{\#}_{\rightarrow}(Cn(a))$ is the finest splitting of $out(G, a)$.*

Proof. Suppose G^2 is consistent. Then $G^{\#}_{\rightarrow}$ exists. By assumption $G^{\#}_{\rightarrow}(a) \neq \emptyset$ whence $G^{\#}_{\rightarrow}(a) \equiv G(a)^{\#}$ by definition 10. Now, $out(G) = G$ so $G(a)^{\#} = out(G)(a)^{\#} = out(G, a)^{\#}$ by lemma 1. Hence it suffices to show that $G^{\#}_{\rightarrow}(a) = G^{\#}_{\rightarrow}(Cn(a))$. Left-in-right is immediate. For the other direction suppose $b \in G^{\#}_{\rightarrow}(Cn(a))$. Then there is a pair $(a', b) \in G^{\#}_{\rightarrow}$ s. t. $a' \in Cn(a)$. By SI it follows that $(a, b) \in G^{\#}_{\rightarrow}$ whence $b \in G^{\#}_{\rightarrow}(a)$ as desired.

Secondly, right-splittings are equivalent, modulo *out*, to the relations they split:

Theorem 7. $out(G^{\#}_{\rightarrow}) = out(G)$

Proof. For the left in right it suffices, by monotony and idempotence for Cn, to show that $G^{\#}_{\rightarrow}(Cn(a))) \subseteq Cn(G(Cn(a)))$. Suppose therefore $b \in G^{\#}_{\rightarrow}(Cn(a)))$. Then there is a norm $(a', b) \in G^{\#}_{\rightarrow}$ with $a \vdash a'$. From the former we have $b \in G(a')^{\#}$, by definition 10, which since $G(a')^{\#}$ is equivalent to $G(a')$, means that $G(a') \vdash b$. Therefore, since $a' \in Cn(a)$, we have $b \in Cn(G(Cn(a)))$ as desired. For the converse direction, suppose $b \in G(Cn(a))$. Then there is a norm $(a', b) \in G$ with $a \vdash a'$. From the former it follows, since $G(a) \equiv G(a)^{\#}$, that $G(a')^{\#} \vdash b$. By compactness for logical consequence there is therefore a finite set $b_1, \ldots, b_n$ of elements of $G(a')^{\#}$ such that $\bigwedge_{i=1}^{n} \vdash b$. Since $a' \in G^1$ we have $(a', b_i) \in G^{\#}_{\rightarrow}$ for $1 \leq i \leq n$, by definition 10. It follows that $G^{\#}_{\rightarrow}(a') \vdash b$. Therefore, since $a' \in Cn(a)$, we have that $b \in Cn(G^{\#}_{\rightarrow}(Cn(a)))$, and we are done.

Our strategy now is as follows: We establish a one-to-one correspondence between *derogations* on a code and *contractions* on its output. More specifically, we show that each full meet derogation of a norm (a, b) from the consequences of a code G, corresponds to the full meet contraction of b from the consequences of $G(a)$. Combining this with theorem 6 then, we may conclude that right-splittings of codes of norms respect relevance among outputs under revisions of the code.

Definition 11. *Let μ be a function such that $\mu(F) = F(Cn(a))$ for $F \in G \perp (a, b)$.*

Then μ maps remainders of a normative system to remainders of its output under the input of the derogandum:

Lemma 7. *If $F \in G \perp (a, b)$ then $\mu(F) \in G(Cn(a)) \perp b$.*

Proof. Suppose $F \in G \perp (a, b)$ and suppose for reduction that $\mu(F) \notin G(Cn(a)) \perp b$. By definition $\mu(F) = F(Cn(a))$, and since $F \in G \perp (a, b)$ we have $F \subseteq G$ whence $F(Cn(a)) \subseteq G(Cn(a))$. There are thus two cases to consider:

1. Either $F(Cn(a)) \vdash b$: By compactness for logical consequence, there is thus a finite set of rules $(a_1, b_1), \ldots, (a_n, b_n) \in F$ such that $a_i \in Cn(a)$ for each $i \leq n$, and $\bigwedge_{i=1}^{n} b_i \vdash b$. Hence $(a, b) \in out(F)$, by repeated applications of SI, AND and WO, contradicting $F \in G \perp (a, b)$.
2. Or there is a $B \in G(Cn(a)) \perp b$ such that $F(Cn(a)) \subset B$: It follows that there is a $d \in B \setminus F(Cn(a))$. Since $B \subseteq G(Cn(a))$ we have $(c, d) \in G$ for some $c \in Cn(a)$. Clearly $(c, d) \notin F$ so $(a, b) \in out(F \cup (c, d))$ by the membership of F in $G \perp (a, b)$. Now, since $b \notin B \supset F(Cn(a))$ it follows that $(a, b) \notin F$ whence $(a, b) \in out(F \cup (c, d)) \setminus out(F)$. By lemma 4 we therefore have $a \vdash c$, and by lemma 3 we have $(a, d \rightarrow b) \in out(F)$. Hence $F(Cn(a)) \vdash d \rightarrow b$ so $B \vdash d \rightarrow b$ by monotony for classical logic. Since $d \in B$ therefore, it follows that $B \vdash b$, contrary to the assumption that $B \in G(Cn(a)) \perp b$.

This mapping is bijective:

Lemma 8. *μ is surjective*

Proof. We want to show that $B = \mu(F)$ for every $B \in G(Cn(a)) \perp b$ and some $F \in G \perp (a, b)$. Put $F := \{(c, d) \in G : B \vdash d \text{ and } c \in Cn(a)\}$. We first show that $B = F(Cn(a))$. For the left-in-right inclusion suppose $d \in B \subseteq G(Cn(a))$ then $(c, d) \in G$ for some $c \in Cn(a)$, so $(c, d) \in F$ by the construction of F. It follows that $d \in F(Cn(a))$. The converse inclusion is immediate from the construction. Next we show that $F \in G \perp (a, b)$. Since $b \notin B$ by the assumption that $B \in G(Cn(a)) \perp b$, it follows that $(a, b) \notin F$ since $B = F(Cn(a))$. Moreover, $F \subseteq G$ by the construction of F so it may be expanded to an $F' \subseteq G$ such that $F \subseteq F'$ and such that $F' \in G \perp (a, b)$. We show that $F(Cn(a)) = F'(Cn(a))$. The left in right inclusion is immediate. For the converse inclusion, suppose $F'(Cn(a) \nsubseteq F(Cn(a))$. Then there is a $d \in F'(Cn(a)) \setminus F(Cn(a))$, whence $d \notin B = F(Cn(a))$. It follows that $B \cup d \vdash b$ so $F(Cn(a)) \cup d \vdash b$, whence $F(Cn(a)) \vdash d \rightarrow b$. In other words, we have $d \rightarrow b \in Cn(F(Cn(a)))$, which means that $(a, d \rightarrow b) \in out(F) \subseteq out(F')$. But then $(a, b) \in out(F')$ by AND, since $d \in F'(Cn(a))$ by assumption, contradicting $F' \in G \perp (a, b)$.

Lemma 9. *μ is injective.*

Proof. We need to show that $\mu(F) = \mu(F')$ implies $F = F'$ for $F, F' \in G \perp (a, b)$. Suppose for reductio that $F(Cn(a)) = F'(Cn(a))$, but $F \neq F'$. Assume without loss of generality, that $(c, d) \in (F' \setminus F) \neq \emptyset$. Then $(a, b) \in out(F \cup (c, d))$ so $a \vdash c$ and $(a \wedge c, d \rightarrow b) \in F$, whence $(a, d \rightarrow b) \in out(F)$. Moreover, $(a, d) \in$

$out(F')$, by SI, since $(c, d) \in F'$. It follows that we have $F(Cn(a)) \cup F'(Cn(a)) \vdash d$ and $F(Cn(a)) \cup F'(Cn(a)) \vdash d \rightarrow b$. However, since $F(cn(a) = F'(Cn(a))$ this is tantamount to saying $F'(Cn(a)) \vdash d$ and $F'(Cn(a)) \vdash d \rightarrow b$. Therefore $F'(Cn(a)) \vdash b$, whence $(a, b) \in out(F')$, contradicting $F' \in G \perp (a, b)$.

The correspondence between derogations and contractions follows as an easy consequence:

Theorem 8. *For any $G \subseteq L \times L$: $(a, d) \in out(G - (a, b))$ iff $G(Cn(a)) - b \vdash d$.*

Proof. It suffices to show that $(a, d) \in \bigcap(G \perp (a, b))$ iff $d \in \bigcap(G(Cn(a)) \perp b$. Suppose $d \in RHS$. Then there is a $B \in G(Cn(a)) \perp b$ such that $B \nvdash d$. Now, $B = F(Cn(a))$ for some $F \subseteq G$, since μ is surjective, so $(a, d) \notin out(F)$. Hence $(a, d) \notin LHS$ as desired. The other direction is similar, except that we appeal to μ^- rather than μ.

These theorems in hand, it is relatively straightforward to redefine the concept of derogation in such a way that it can be shown to respect relevance *among outputs*. Let's first introduce the following shorthands:

Definition 12.

1. $G \backsim (a, b) = \bigcap(G^{\#}_{\rightarrow} \perp (a, b))$
2. $A \backsim a = \bigcap(A^{\#} \perp a)$

Note that $G \backsim (a, b) = G^{\#} - (a, b)$, and $A \backsim a = A^{\#} - a$. The idea now is to make positive permissions relevance-respecting by redefining the concept of antithetic permission in the following manner:

Definition 13. *(a, b) is antithetically permitted in $\langle G, P \rangle$ iff (a, b) is not derivable from $out(G \cup (a, b)) \backsim (c, \neg d)$ for some $(c, d) \in P$ such that $a \equiv c$*

The next, theorem, which is the main result of this paper, shows that this strategy does work:

Theorem 9. *According to definition 13, if G^2 is consistent then (a, b) is antithetically permitted in $\langle G, P \rangle$ by (c, d) only if d is relevant to b modulo $out(G \cup (a, \neg b), a)$.*

Proof.

1. $out(G \cup (a, \neg b)) \backsim (c, \neg d) \nvdash_G (a, \neg b)$, by assumption
2. $out(G \cup (a, \neg b))^{\#}_{\rightarrow} - (c, \neg d) \nvdash_G (a, \neg b)$, by definition 12
3. $(a, \neg b) \notin out(out(G \cup (a, \neg b))^{\#}_{\rightarrow} - (c, \neg d))$, by $out = deriv$
4. $(a, \neg b) \notin out(out(G \cup (a, \neg b))^{\#}_{\rightarrow} - (a, \neg d))$, by $c \equiv a$
5. $out(G \cup (a, \neg b))(Cn(a))^{\#}_{\rightarrow} - \neg d \nvdash \neg b$, by theorem 8
6. $out(out(G \cup (a, \neg b)), a)^{\#} - \neg d \nvdash \neg b$, by lemma 6
7. $out(G \cup (a, \neg b), a)^{\#} - \neg d \nvdash \neg b$, by idempotence for out

8. b is relevant to d modulo $out(G \cup (a, \neg b))$, by theorem 6.

It remains to show that this suffices to avoid the triviality result from theorem 4. The following simple example will do:

Example 2. Let $G := \{(a, \neg b)\}$ and $P := \{(a, b)\}$, where b is an elementary letter. Then (a, b) is an exemption regardless of whether we use $-$ or $\backsim$ in the definition of exemptions. However, (a, e) is not antithetically permitted for arbitrary e, since we may choose e in such a way that it is not relevant to b modulo $out(G \cup (a, \neg e), a)$. For instance, let e be another elementary letter then:

- $out(G \cup (a, \neg e), a)^{\#} = \{\{b\}, \{e\}\}$

Clearly $E(b^*) \cap E(e^*) = \emptyset$, whence a is cell-relevant to b modulo $out(G \cup (a, \neg e), a)$ only if $E(b^*)$ and $E(e^*)$ each share an elementary letter with the same cell in the finest splitting. By inspection this is not the case.

6 Discussion

In order to exploit the correspondence from theorem 8 in combination with theorem 6, the output of the split code, on an arbitrary input, must itself be a set of formulae on finest splitting form. This is not in general ensured by definition 10, and it does not hold for *open* sets of norms G. For open G, G^1 does not contain all possible inputs. Therefore the image of an open G under an arbitrary input might not be on finest splitting form, as the following example shows:

Example 3. Put $A := \{a\}$ and $B := \{a \vee b\}$. Then $A^{\#} = \{\{a\}\}$ and $B^{\#} = \{\{a, b\}\}$. However $(A \cup B)^{\#} = \{\{a\}\}$, whence, since finest splitting forms are unique up to logical consequence, it follows that $A \cup B$ is not on finest splitting form.

The significance of theorem 9 thus consists in the fact that it shows the potential gain in searching for canonical forms also for codes of norms, and in the fact that operations of change emerge as an important factor in evaluating the suitability of such forms. Theorem 9 shows that splittings, and least letter set forms more generally, are worthy of attention.

To be sure, it is not obvious that splitting is always deontically harmless. Some norms it seems, such as 'don't drink and drive' are essentially *conjunctive* and should not be split, lest driving be forbidden *per se*. However, as shown in [16], it is possible to protect chosen elements from the splitting, thus suspending the relevance criterion for particular distinguished cases. We conjecture that this technique is easily adapted to input/output logic, for instance to certain cases of conjunctive norms. In combination with splitting it would give us very good control of the behaviour of a code of norms under revisions of the code, and is a topic for future investigations.

Apart from this, the correspondence between contraction on sets of sentences and sets of norms (in the input/output sense) recorded in theorem 8 may have an *eigenvalue* as it allows us to carry results back and forth between the two idioms. It would for instance be interesting to see how entrenchment relations on sets of sentences translate into priorities on sets of norms.

7 Related work

The idea of analysing positive permission as exceptions to mandatory norms has forerunners in the litterature. It is discussed in [4] where, ignoring a few technical details and a few limiting cases, the set of positive permissions on input a according to a code $\langle G, P\rangle$ is defined in terms of the set $\{out(H \cup (c, z), a) : H \subseteq G, (c, z) \in P$ and $out(G' \cup (c, z), a) \cup a \nvdash f\}$. This is similar to the account given in [13] insofar as permissive norms are treated as *weak obligations*. That is, permissive norms are allowed to generate output alongside obligations. In the framework presented in this paper, explicit permissive norms are used only as constraints, i.e. they do not contribute to the output of the code. In [7] permission is defined as the non-derivability under defeaters of a contrary obligation. This is also, it seems, a close relative of the idea presented in this paper. However, [7] is based on a rather complex default logic-framework, which precludes a point for point comparison with the present work. It is a topic for future investigations.

Acknowledgements

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 *Proc. DEON 2008*, pp. 96–109. Springer-Verlag, 2008.
2. C. E. Alchourron and E. Bulygin: "The expressive conception of norms", in Hilpinen (ed.): *New Studies in Deontic Logic*. D. Rediel Publishing Company, 1981.
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. G. Boella and L. van der Torre: "Institutions with a hiearchy of authorities in distributed dynamic environments", in *Artificial Intelligence in Law*, vol. 16 no. 1, 2008.
5. G. Boella, G. Pigozzi and L. van der Torre: "Normative framework for normative system change". AAMAS (1), pp. 169-176, 2009.
6. B. F. Chellas: *Modal Logic: An Introduction*. Cambridge University Press, 1980.
7. G. Governatori, A. Rotolo, and G. Sartor:. "Temporalised Normative Positions In Defeasible Logic". *Proceedings of the International Conference on Artificial Intelligence and Law*, pp. 25-34, 2005

8. G. Governatori, A. Rotolo, R. Riveret, M. Palmirani and G. Sartor (2007). "Variants Of Temporal Defeasible Logic For Modelling Norm Modifications". ICAIL 2007, pp. 155-159.
9. G. Governatori and A. Rotolo: "Changing Legal Systems: Abrogation and Annulment Part I: Revision of Defeasible Theories", in *Proc DEON 2008*, Springer 2008.
10. D. C. Makinson and L. van der Torre: "Input/output logics", in *Journal of Philosophical Logic* vol. 29, pp. 383–408, 2000.
11. D. C. Makinson and L. van der Torre: "Constraints for input/output logics", in *Journal of Philosophical Logic*, vol. 30, pp. 155–185, 2001.
12. D. C. Makinson and L. van der Torre: "What is input/output logic", in *Applications of Mathematical Logic in Philosophy and Linguistics* vol. 17, pp. 163–174, 2003.
13. 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.
14. D. C. Makinson: "Friendliness and sympathy in logic", in: J.-Y. Beziau (Ed.), *Logica Universalis*, 2nd edition, Birkhauser Verlag, Basel, pp. 195-224, 2007.
15. D. C. Makinson and G. Kourousias: "Parallel interpolation, splitting, and relevance in belief change", in *Journal of Symbolic Logic*, vol. 72, issue 3, pp. 994-1002, 2007.
16. D. C. Makinson: "Propositional relevance through letter-sharing", in *Journal of Applied Logic*, vol. 7, pp. 377-387, 2009.
17. Rohit Parikh: "Beliefs, belief revision, and splitting languages", in: Lawrence Moss, Jonathan Ginzburg, Maarten de Rijke (Eds.), *Logic, Language, and Computation*, vol. 96, CSLI Publications, Stanford, CA, pp. 266-278, 1999,
18. A. Stolpe: *Norms and Norm-System Dynamics*. PhD thesis. Department of Philosophy, University of Bergen, Norway. 2008.
19. A. Stolpe: "A theory of permission based on the notion of derogation". *Journal of Applied Logic* vol. 8, pp. 97-113, 2010.
20. 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.