Fundamenta Informaticae 45 (2001) 1–21
IOS Press

On the Representation of Action and Agency in the Theory of Normative Positions

Marek Sergot

Fiona Richards

Department of Computing

Imperial College of Science, Technology and Medicine

London, UK

{*mjs,fcmr*}*@doc.ic.ac.uk*

Abstract. The theory of normative positions attempts to apply a combination of deontic logic and a logic of action/agency to the formalisation of the 'Hohfeldian concepts' (duty, right, power, privilege, etc.) and other complex normative relations between agents. This paper is concerned with the practical applicability of the theory to such tasks as formalising the content of an existing set of regulations, designing a new set of regulations, or refining aspects of a computer system specification, with particular attention to the usefulness of the action component as a representational device. Points are made by reference to a simple example concerning rules in a car park. The second part of the paper covers three simple extensions: the representation of 'permitted to bring about', a treatment of interpersonal control relations, and the ability to distinguish between being permitted to bring about a new state of affairs and being permitted to sustain a state of affairs that already exists.

Keywords: Deontic logic, normative systems, logic of action, logic of agency, theory of rights, knowledge representation, computer system specification.

1. Introduction

The theory of normative positions is an attempt to apply the tools of modal logic to the formalisation of Hohfeld's [8] 'fundamental legal conceptions' (duty, right, power, privilege, etc.) and other complex normative relations that can hold between (pairs of) agents. The development was initiated by Stig Kanger [16, 17] and subsequently extended by Lars Lindahl [19]. Ingmar Pörn [25] applied similar techniques to the study of 'control and influence' relations in social interactions. Sergot [33, 34, 32] presents a generalisation and further development of the Kanger-Lindahl account together with techniques for automating the associated inference methods in computer programs. References to related works by other authors are given in [33, 34].

The extended theory has four components:

1. a deontic logic;
2. a logic of action/agency;
3. a method for generating specified classes of 'normative positions' and establishing relationships between them;
4. techniques for automating these methods, as incorporated in the computer program, Norman-G, intended to support application of the theory to the analysis of practical examples.

The deontic logic component is a standard one. Expressions of the form $\mathrm{O}A$ and $\mathrm{P}A$ may be read as 'it is obligatory that, or it it ought to be the case that, A' and 'it is permissible, or permitted, that A', respectively. P is the dual of O: $\mathrm{P}A \stackrel{\text{def}}{=} \neg\mathrm{O}\neg A$. Specifically, the deontic logic employed by Kanger and Lindahl is a classical modal logic of type *EMCP* in the Chellas [5] classification, which is essentially the system often referred to as Standard Deontic Logic (SDL).[1] Details of the deontic logic can be varied though we shall not do so here. The deontic logic component is not the main focus of this paper.

The action component adds to the language (relativised) modal operators E_a, E_b, $\ldots$, where a, b, $\ldots$, are the names of agents. An expression of the form $\mathrm{E}_x A$ may be read as 'agent x sees to it that A', 'agent x brings it about that A', 'agent x is responsible for its being the case that A'. Further discussion of this component follows below.

Combinations of the deontic and action operators allow various kinds of normative relations between agents to be expressed. Compare for example the following pairs of sentences, where F represents some state of affairs that agents a and b can bring about:

$$
\begin{array}{rcl}
\mathrm{P}\,\mathrm{E}_a F \wedge \mathrm{P}\,\mathrm{E}_b F & \textit{vs.} & \mathrm{P}\,\mathrm{E}_a F \wedge \neg\mathrm{P}\,\mathrm{E}_b F \\
\mathrm{P}\,\mathrm{E}_a F \wedge \mathrm{P}\,\mathrm{E}_b \neg F & \textit{vs.} & \mathrm{P}\,\mathrm{E}_a F \wedge \neg\mathrm{P}\,\mathrm{E}_b \neg F \\
\mathrm{O}\,\mathrm{E}_a F \wedge \mathrm{O}\,\mathrm{E}_b F & \textit{vs.} & \mathrm{O}\,(\mathrm{E}_a F \vee \mathrm{E}_b F) \wedge \neg\mathrm{O}\,\mathrm{E}_a F \wedge \neg\mathrm{O}\,\mathrm{E}_b F \\
\mathrm{P}\,\mathrm{E}_a F \wedge \neg\mathrm{P}\,\mathrm{E}_b \neg F \wedge \mathrm{P}\,\mathrm{E}_b \neg\mathrm{E}_a F & \textit{vs.} & \mathrm{P}\,\mathrm{E}_a F \wedge \neg\mathrm{P}\,\mathrm{E}_b \neg F \wedge \neg\mathrm{P}\,\mathrm{E}_b \neg\mathrm{E}_a F
\end{array}
$$

There are many other possibilities, of course.

[1] A logic of type *EMCP* can be characterised [5] as the smallest system containing propositional logic (*PL*) and the following axiom schemas and rules:

O.RE	If $\vdash A \leftrightarrow B$ then $\vdash \mathrm{O}A \leftrightarrow \mathrm{O}B$
O.M	$\mathrm{O}(A \wedge B) \rightarrow (\mathrm{O}A \wedge \mathrm{O}B)$
O.C	$(\mathrm{O}A \wedge \mathrm{O}B) \rightarrow \mathrm{O}(A \wedge B)$
O.P	$\neg\mathrm{O}\bot$

Standard Deontic Logic (SDL) is a normal modal logic of type *KD*, which is type *EMCP* together with the additional rule of necessitation If $\vdash A$ then $\vdash \mathrm{O}A$. The rule of necessitation plays no role in the generation of classes of normative positions.

The theory of normative positions provides a method for systematically and exhaustively mapping out the space of all logically possible normative relations that can be distinguished by specified combinations of deontic and action operators. These relations are the 'normative positions' of a given class. The method is summarised in Section 3. A simple example, illustrating also the operation of the Norman-G program, is presented in Section 4.

Viewed as a formal theory of duties and rights, the Kanger-Lindahl framework has important and well-documented limitations, especially in regard to the feature Hohfeld called '(legal) power', also known as 'legal capacity' or 'competence' (see e.g. the discussion in [21, 20, 15]). Nevertheless it provides a powerful and expressive framework that is capable of articulating many distinctions and nuances, and a way of exploring these distinctions in a systematic fashion. The aim of this present paper is to comment on the adequacy of the framework as a practical *representational* device. We have in mind applications such as formalising the content of an existing set of regulations, or designing a new set of regulations, or clarifying and refining aspects of a computer system specification. Here, we want to focus on the action component, that is, on the use of the 'brings it about' or 'sees to it that' operators.

The formal properties of such action operators have been extensively investigated, and there are many variants. See e.g. [4, 23, 24, 25, 26, 1, 29, 30, 31, 2, 3, 22, 12, 6, 27, 7]. However the emphasis in these works has been on problems of characterisation, on attempting to explicate what it means for an agent to 'act', what it means for an agent to be 'responsible' for such-and-such a state of affairs being the case. No attention has been paid to the adequacy of the approach for purposes of representation. Although the level of abstraction can be useful, it is not appropriate for representing all forms of action expressions; even where it is appropriate, the resulting representation can be cumbersome and unnatural. The aim of this present paper is to consider whether, or to what extent, the advantages of this approach to the logic of action are outweighed by its inadequacies as a representational formalism, in the context of representing normative relations. We limit ourselves to general considerations. Except where indicated otherwise, we do not want to commit in this paper to any particular version of the logic or to any particular semantical account. Although the E_x notation was introduced by Pörn for something specific, here we use it because it is concise, to stand for any variant.

2. The logic of action/agency: preliminary remarks

In generating classes of normative positions, the theory makes use of just two properties of the logic of action/agency, which are common to all variants:

$$\begin{array}{ll} \text{E.RE} & \text{If } \vdash A \leftrightarrow B \text{ then } \vdash \mathrm{E}_x A \leftrightarrow \mathrm{E}_x B \\ \text{E.T} & \mathrm{E}_x A \rightarrow A \end{array}$$

The schema E.T indicates that this is a notion of *successful* action: if agent x brings it about that A then it is indeed the case that A. Or to put it another way (Hilpinen [7] quoting Chellas [4]): x can be held responsible for its being the case that A only if it is the case that A.

It still remains to capture the concept of *agency* itself—the fundamental idea that state of affairs A is, in some sense, caused by or is the result of actions by agent x. Most accounts introduce a 'negative condition' or 'counterfactual condition' for this purpose, to express that if x had not acted in the way he or she did then the world would, or might, have been different. Hilpinen [7] provides an overview of the main semantical devices that have been used. In axiomatic presentations this feature of agency is often reflected (among other things) in the validity of the following schema:

$$\text{E.no}\top \qquad \neg \mathrm{E}_x \top$$

No agent can meaningfully bring about what is logically true, or more generally, what was unavoidable anyway. The schema E.no$\top$, though it plays no role in the generation of normative positions, is clearly central to the desired notion of agency and must be taken into account when interpreting expressions containing E_x operators. We return to this point at the end of this introductory section. A case can be made for other properties of the 'brings it about' operators but we will not discuss them here.

There is still much scope for variation. For example, Hilpinen [7] observes: "The expression 'seeing to it that A' usually characterises deliberate, intentional action. 'Bringing it about that A' does not have such a connotation, and can be applied equally well to the unintentional as well as intentional (intended) consequences of one's actions, including highly improbable and accidental consequences." Similarly, it has been suggested [27, 28] that for certain purposes it may be important to distinguish between an agent's direct action and its indirect action whereby it exerts control or influence over the actions of another. We will not make such distinctions in this paper.

For present purposes, the essential point is that this general approach to the treatment of action abstracts away from considerations of state change and the temporal dimension, focussing only on the agent concerned and the states of affairs that result from his or her actions. For use as a representational device, this level of abstraction is both a strength and a weakness. It is a strength because when dealing with the complexity of human affairs it is sometimes difficult, or impossible, to pinpoint exactly what are the actions of an agent x by means of which some state of affairs A is brought about. This is especially true when there are actions of other agents to be taken into account. For example, suppose A represents that agent a has access to some piece of information i. It could be that x brings it about that A by granting a read-access to the computer file containing i, or by unlocking the door to the room in which i is kept, or by leaving a document lying in a train compartment for a moment, or by telling a the password to the computer system in which the information i is stored, or by retrieving i from the computer system and then passing it on to a. The possibilities are endless. In some cases the actions of x are quite deliberate and the result is intended; in others, x brings it about that A by chance. But when specifying what is to be permitted, perhaps as part of a computer system's access control mechanisms or as part of the general security policy of an organisation, it may be that only the end result A is important, and the fact that it is x who brings it about: the exact nature of x's actions do not have to be pinpointed, even if this were possible.

In other cases, this same feature is a major weakness, which can even make the approach unusable as a representational device. This is after all a very abstract representation of action. For instance, sometimes it is essential to be able to refer to the *means* by which an agent brings about a state of affairs: x may be permitted to bring about A by performing action α but not be permitted to bring about the same state of affairs by performing action β. A partial solution is offered by Elgesem [6] who has studied dyadic operators of the form 'x brings it about that A by bringing it about that B', but this is a partial solution only because it is still not possible to refer to the actions by means of which x brings it about that B. See also the discussion in [7].

Sometimes it is necessary to refer explicitly to what some call the 'primitive actions' of an agent. These are often, though not always, identified with intentional bodily movements. When we say that agent x is permitted to kick an opponent this is not adequately represented by saying that agent x is permitted to see to it that an opponent is kicked. The best we can do is to resort to some circumlocution such as agent x is permitted to see to it that an opponent is kicked by x. Hohfeld uses an example of eating a prawn salad—with the E_x device we have to say that x sees to it that the prawn salad is eaten by x. Lindahl uses an example of walking on a neighbour's land—agent x sees to it that x is walking on a neighbour's land. It is difficult to see what is added by introducing an operator for *agency* to represent such expressions. And likewise for many other kinds of actions that need to be represented: sending a message, driving a car, sending a message, writing to a file.

When modelling computer systems there is often a limited number of action-types to consider, and it is easier and more natural to refer to these actions directly instead of indirectly to their end results. This can also be true when modelling human interactions, if those interactions can be regarded as taking place in some closed world where for the purposes of the representation it is sufficient to consider a fixed number of primitive action types, and where the effect of each action type is easily expressed. In these cases, it is much more natural to employ the kind of action representation typically encountered in computer science and artificial intelligence.

Sometimes there is a choice. Suppose we want to represent that x is permitted, or obliged, to sign a certain document. We may want to refer explicitly to the action of 'signing'. But perhaps what is important is just that the document contains the signature of x. If $s(x)$ represents that the document contains the signature of x, it is again difficult to see what is gained by adding that it is agent x who brings it about that the document contains the signature of x. However, one way of looking at it is to say that there are additional (non-logical) axioms to be included in the representation to the effect that: $\vdash s(x) \leftrightarrow \mathrm{E}_x\, s(x)$. Indeed, a case can be made that it is precisely the property $\vdash \alpha \leftrightarrow \mathrm{E}_x\, \alpha$ that is the core of the characterisation of α as an act description (see [25, pp11–12]). Now, even though $s(x)$ and $\mathrm{E}_x\, s(x)$ are logically equivalent, we might still want to say e.g. $\mathrm{E}_x \neg s(x)$, which represents a kind of refraining by x from signing the document, or $\mathrm{E}_y \neg s(x)$, which represents a kind of prevention by y of the signing of the document by x. The expression $\mathrm{E}_y\, s(x)$ can also make sense, in certain readings. Whether such expressions are *useful* depends of course on the purpose for which the representation is constructed.

In summary, although there are severe limitations on the use of the 'brings it about' or 'sees to it that' constructions in practical problems, there are circumstances where, on the face of it, they can offer something valuable. This is what we wish to explore. Most of the points can be illustrated by reference to a simple, though also rather typical, example, introduced in Section 4. Section 3 summarises briefly the methods for generating classes of normative positions as required for discussion of the examples.

First, we want to make one final observation. We find that when speaking loosely, there is a tendency to read expressions containing E_x with emphasis on the 'end result' feature and insufficient attention to the agency component. It can be tempting, in particular, to read an expression $\mathrm{OE}_x F$ as a representation of an ought-to-do statement that 'x ought to see to it that F'. Thus, if F represents that there is a fence between two neighbouring properties (an example used by Lindahl) then $\mathrm{OE}_x F$ might be read as saying that x ought to do something—to perform some action or omit to perform some action—which results in there being a fence. And further, if F represents that a certain car is parked in a car park (an example used in Section 4) then there is a temptation to read $\mathrm{OE}_x F$ as saying that x ought to park the car. However, this is not what the expression $\mathrm{OE}_x F$ says. As has also been pointed out by several other authors (see e.g. [10, 9, 11, 7]), the statement that x ought to be an agent of F is to be distinguished from the statement that x ought to do something that results in F. For example, x does not fulfil the obligation $\mathrm{OE}_x F$ if F would have happened anyway, whatever x did. To emphasise the more accurate reading, we sometimes write $\mathrm{O}\,(F \wedge \mathrm{E}_x F)$ rather than $\mathrm{O}\,\mathrm{E}_x F$. Although the two expressions are logically equivalent, we find there is less temptation to read $\mathrm{O}\,(F \wedge \mathrm{E}_x F)$ as 'x ought to see to it that F'. $\mathrm{O}\,(F \wedge \mathrm{E}_x F)$ suggests 'it ought to be the case that F and that x is responsible for its being the case that F', which is a more accurate reading of the expression $\mathrm{O}\,\mathrm{E}_x F$.

3. Classes of normative positions

In this section we summarise the theory of normative positions as introduced by Kanger [16, 17] and Lindahl [19] and subsequently developed and extended by Sergot [33, 34].

The core of Kanger's analysis of the 'Hohfeldian' relations are what he called the 'atomic types of rights relations' of two agents a and b with respect to some state of affairs F. They can be characterised [21] as the expressions belonging to the set:

$$\left[\!\left[\pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right]\!\right] \tag{1}$$

The expression inside the brackets stands for the set of sixteen sentences of the form $\pm\, \mathrm{O} \pm \mathrm{E}_a \pm F$ or $\pm\, \mathrm{O} \pm \mathrm{E}_b \pm F$, where $\pm$ stands for the two possibilities of affirmation and negation. The brackets denote *maxi-conjunctions*: $[\![\Phi]\!]$ stands for the set of *maxi-conjunctions* of a set of expressions Φ, i.e., the maximal consistent conjunctions[2] of expressions belonging to the set Φ. A conjunction is 'maximal consistent' when addition of any other conjunct from Φ yields an inconsistent conjunction. A conjunction Γ is thus a maxi-conjunction of Φ if and only if Γ is consistent, and every expression of Φ either appears as a conjunct in Γ or is inconsistent with Γ.

It can be seen that, by construction, the maxi-conjunctions (1) are consistent, mutually exclusive, and their disjunction is a tautology. In any given situation precisely one of them must be true, according to the logical principles employed. They partition the set of logical possibilities.

As in [33, 34], we say that a (finite) set of sentences $\{P_1,\ P_2, \ldots, P_n\}$ *partitions*, or *is a partition of*, a sentence Q when

1. every element P_i of is logically consistent: $\nvdash \neg P_i$;
2. the state of affairs represented by each P_i is a special case of the state of affairs represented by Q: $\vdash P_i \rightarrow Q$;
3. distinct elements of P_i and P_j are mutually exclusive: $\vdash \neg(P_i\ \wedge\ P_j) \quad (i \neq j)$;
4. the set $\{P_1,\ P_2, \ldots, P_n\}$ 'exhausts' Q: $\vdash Q \rightarrow (P_1 \vee \ldots \vee P_n)$.

In that case, $\vdash (P_1 \vee \ldots \vee P_n) \leftrightarrow Q$. When Q is a tautology we just say that $\{P_1, \ldots, P_n\}$ is a partition. Kanger's 'atomic types' (1) are a partition in this sense.

Kanger's 'atomic types of rights relations' (1), also referred to as the Kanger set of (two-agent) *normative positions*, can be written equivalently [33, 34] as conjunctions of two simpler expressions:

$$\left[\!\left[\pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right]\!\right] \quad = \quad \left[\!\left[\pm\, \mathrm{O} \pm \mathrm{E}_a \pm F \right]\!\right] \cdot \left[\!\left[\pm\, \mathrm{O} \pm \mathrm{E}_b \pm F \right]\!\right] \tag{2}$$

[2] 'Consistent' refers to the underlying logic, here the logics for O and E_x specified above. 'Conjunction' means a conjunction without repetitions, and with some standard order and association of conjuncts.

The $\cdot$ notation in (2) stands for the set of all the *consistent* conjunctions that can be formed by conjoining an expression from the left hand argument with an expression from the right hand argument. Equation (2) shows that each of Kanger's two-agent normative positions can be expressed as a conjunction of two one-agent normative positions.

Lindahl [19] (besides discussing also various kinds of *change* of normative positions, which we ignore here) extended Kanger's analysis in two ways. He provides a more refined analysis of one-agent normative positions (for agent a with respect to F) by means of the following construction[3]:

$$\left[\!\!\left[\pm\, \mathrm{O} \pm [\![\pm\, \mathrm{E}_a \pm F]\!] \right]\!\!\right] \tag{3}$$

in which now there is a maxi-conjunction expression $[\![\pm\, \mathrm{E}_a \pm F]\!]$ inside the scope of the O operator. By basing the construction on the richer set of 'act positions' $[\![\pm\, \mathrm{E}_a \pm F]\!]$, Lindahl obtains a finer-grained analysis than Kanger's, allowing one to distinguish, for example, between $\mathrm{P}\,\mathrm{E}_x F \wedge \mathrm{P}\neg\mathrm{E}_x F \wedge \mathrm{P}\,(\neg\mathrm{E}_x F \wedge \neg\mathrm{E}_x\neg F)$ and $\mathrm{P}\,\mathrm{E}_x F \wedge \mathrm{P}\neg\mathrm{E}_x F \wedge \neg\mathrm{P}\,(\neg\mathrm{E}_x F \wedge \neg\mathrm{E}_x\neg F)$.

Lindahl's second extension was to distinguish between what he called 'individualistic' and 'collectivistic' types of two-agent normative positions, based on the following observation:

$$\left[\!\!\left[\pm\, \mathrm{O} \pm [\![\pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F]\!] \right]\!\!\right] \quad \neq \quad \left[\!\!\left[\pm\, \mathrm{O} \pm [\![\pm\, \mathrm{E}_a \pm F]\!] \right]\!\!\right] \cdot \left[\!\!\left[\pm\, \mathrm{O} \pm [\![\pm\, \mathrm{E}_b \pm F]\!] \right]\!\!\right] \tag{4}$$

The equality does not hold because, e.g., O does not distribute across disjunction in the logics employed. The 'collectivistic' form on the left provides a finer analysis than the 'individualistic' form on the right, in the sense that each of the 'individualistic' positions is partitioned by (is equivalent to a disjunction of mutually exclusive) 'collectivistic' positions.

The relationship between these analyses can be described in general terms as follows. Each provides a partition, i.e. an exhaustive characterisation of the logically possible positions, but some of these partitions are finer than others. Each of Kanger's two-agent positions ('atomic types of right') is logically equivalent to a disjunction of Lindahl's 'individualistic' positions; each of these 'individualistic' positions in turn is logically equivalent to a disjunction of 'collectivistic' positions. And likewise for the one-agent positions from which the ('individualistic') two-agent positions are constructed. As in [33, 34], we call such relationships *refinements*: the partition $\mathbf{P}$ is a *refinement* of partition $\mathbf{Q}$ when every element of $\mathbf{P}$ logically implies some element of $\mathbf{Q}$. Because of the structure of partitions, this means that every element of $\mathbf{Q}$ is logically equivalent to a disjunction of (by definition, mutually exclusive) elements of $\mathbf{P}$.

The constructions employed by Lindahl, however, do not yet produce the most refined set of normative positions. For agents a and b and state of affairs F, the most refined set of normative positions (when the logic of O is of type *EMCP*) is given by the following [33, 34]:

$$\left\langle \pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right\rangle \overset{\text{def}}{=} \left[\!\!\left[\pm\, \mathrm{O} \pm \left\langle \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right\rangle \right]\!\!\right] \tag{5}$$

$$\left\langle \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right\rangle \overset{\text{def}}{=} \left[\!\!\left[\pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm F \right]\!\!\right] \cdot \left[\!\!\left[\pm F \right]\!\!\right] \tag{6}$$

[3] The construction used by Lindahl is actually $[\![\pm\, \mathrm{P}\, [\![\pm\, \mathrm{E}_a \pm F]\!]]\!]$. This simpler form is equivalent to (3) when the logic of O is of type *EMCP* or stronger [33, 34].

These are the positions underlying the computer program Norman-G whose use is described presently. The corresponding most refined one-agent positions are as follows:

$$\langle \pm\ \mathrm{O} \pm \mathrm{E}_a \pm F \rangle \quad \stackrel{\text{def}}{=} \quad [\![\pm\ \mathrm{O} \pm \langle \pm\ \mathrm{E}_a \pm F \rangle]\!] \tag{7}$$

$$\langle \pm\ \mathrm{E}_a \pm F \rangle \quad \stackrel{\text{def}}{=} \quad [\![\pm\ \mathrm{E}_a \pm F]\!] \cdot [\![\pm\ F]\!] \tag{8}$$

The account can be generalised to any number of agents $\{a, \ldots, b\}$ and any number of separate states of affairs $\{F, \ldots, G\}$. The constructions supported by the Norman-G program are the following:

$$\left\langle \pm\ \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \vdots \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right\rangle \quad \stackrel{\text{def}}{=} \quad \left[\!\!\left[\pm\ \mathrm{O} \pm \left\langle \pm \begin{pmatrix} \mathrm{E}_a \\ \vdots \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right\rangle \right]\!\!\right] \tag{9}$$

$$\left\langle \pm \begin{pmatrix} \mathrm{E}_a \\ \vdots \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right\rangle \quad \stackrel{\text{def}}{=} \quad \left[\!\!\left[\pm \begin{pmatrix} \mathrm{E}_a \\ \vdots \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right]\!\!\right] \cdot \left[\!\!\left[\pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right]\!\!\right] \tag{10}$$

An example is provided in the following section. For uniformity we define :

$$\langle \pm\ \mathrm{O} \pm F \rangle \quad \stackrel{\text{def}}{=} \quad [\![\pm\ \mathrm{O} \pm \langle \pm\ F \rangle]\!] = [\![\pm\ \mathrm{O} \pm F]\!] = \{\mathrm{O}F,\ \mathrm{P}F \wedge \mathrm{P}\neg F,\ \mathrm{O}\neg F\}$$

We note in passing that the constructions for multiple agents and multiple states of affairs given by (9) and (10) do not produce the most refined set of normative positions, even for logics of type *EMCP*. Act expressions such as $\mathrm{E}_a(F \wedge G)$, $\mathrm{E}_a(F \wedge \neg G)$, $\mathrm{E}_a(F \vee G)$ are not covered by this scheme. What the constructions (9) and (10) produce is the most refined set of normative positions that can be obtained when only elementary propositions or their negations are allowed within the scope of an action operator. We find that in practice any more refined set is excessively detailed and not worth the trouble. For this reason we do not support it in Norman-G.

The purpose of the Norman-G program is to facilitate exploration of classes of normative positions, and application of the theory to the analysis of practical examples. A typical example is the case study discussed by Jones and Sergot [13, 14] concerning the specification of access rights to patient data in a hospital database. The problem is to clarify and expand an incomplete and imprecise statement of requirements into a precise specification at whatever level of detail is desired. Generally, the program is intended to support the interpretation and disambiguation of legal texts, the design of new sets of rules and regulations, and the construction of computer system specifications.

Automated support is required for such applications not just to keep track of decisions made during the course of an analysis but also because of the number and complexity of the symbolic expressions to be manipulated. On Kanger's analysis, there are 26 distinct 'atomic types of right' for any pair of agents with respect to some given state of affairs. In principle one could present the complete list and ask which one applies in the case under consideration: there must be exactly one, according to the logical principles employed, though of course it is always possible that some element of uncertainty remains. In practice it is easy to generate simpler questions which will discriminate between several possible positions to make the task of identifying a position easier. As illustrated presently, Norman-G provides such a mechanism.

The number of possibilities increases very rapidly as more refined analyses are considered. On Lindahl's analysis, for example, there are 127 'collectivistic' two-agent positions for any pair of agents and a given state of affairs; the most refined set of positions, as shown in [33, 34], actually contains 255 positions. Again, it is possible to generate a series of questions which will discriminate between the possibilities and help identify the one that applies. In general, when there are m agents and n states of affairs to consider, the set of normative positions of the form (9) and (10) above contains $2^{2^{(m+1)n}} - 1$ distinct elements. This is a very large number of positions to have to examine even when m and n are small.

The reason that it can be practical to perform such an analysis is that the analysis can be conducted by a process of progressive refinement. If $\mathbf{P}$ and $\mathbf{Q}$ are both sets of normative positions such that $\mathbf{P}$ is a refinement of $\mathbf{Q}$ there are very strong relationships between their elements. We do not present the details here (see [33, 34]), except to say that picking out any set of positions in $\mathbf{Q}$ immediately reduces the number of positions to be considered in the more refined set $\mathbf{P}$. As long as one starts the analysis at a coarse level of detail, and picks out appropriate positions at that level before moving to more refined partitions, the choice of positions at each stage is quite manageable and not difficult to track. Determining the choices at each stage, keeping track of the inferences, and displaying the remaining possibilities are the main functions of the Norman-G program.

4. Example (Car park)

The purpose of this section is to make concrete the summary provided in the previous section and to illustrate the operation of the Norman-G program on a simple example. Later sections will pick out features of the example for further discussion.

The example is a modified form of one used by Ronald Lee in [18] to illustrate a rule-based language for specifying permitted, obligatory and forbidden actions. (We use it here to make a different set of points.) The example concerns the specification of which categories of staff are permitted and not permitted to park in a car park. We choose it because it is familiar and requires no further explanation. In Lee's example, administrators are permitted to park in the car park. Other categories of staff may be permitted to park in certain circumstances, which we ignore.

Consider the following scenario:

> a is an administrator, permitted to park in the car park. a has two cars, car_a_1 and car_a_2. b is a disgraced administrator, banned from the car park. b has one car, car_b. c is a passer-by. g is the gatekeeper, charged with controlling access to the car park and ensuring that the rules are obeyed.

We shall not attempt in this paper to cover every feature of the example. In particular the representation of what it means to say that the gatekeeper g is responsible for ensuring that the rules of the car park are obeyed raises a number of difficult points most of which are outside the scope of this paper.

Let $p(a_1)$, $p(a_2)$, $p(b)$ represent that cars car_a_1, car_a_2, car_b are parked in the car park, respectively. We take it that the following at least is implicit and obvious from the scenario description as given above: that it is not permitted that car_b is parked in the car park, $\neg \mathrm{P}\, p(b)$; that it is permitted but not obligatory that car_a_1 is parked in the car park, $\mathrm{P}\, p(a_1) \wedge \mathrm{P}\, \neg p(a_1)$; and that it is permitted but not obligatory that car_a_2 is parked in the car park, $\mathrm{P}\, p(a_2) \wedge \mathrm{P}\, \neg p(a_2)$.

What else holds according to the rules of the car park (as we imagine them to be from the scenario and previous experience of typical car parks)? In order to investigate the possibilities in a systematic fashion and identify any points requiring further clarification, the task is to pick out one or, in the case of some residual uncertainty, several of the positions from the following scheme:

$$\left\langle \pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \\ \mathrm{E}_c \\ \mathrm{E}_g \end{pmatrix} \pm \begin{pmatrix} p(a_1) \\ p(a_2) \\ p(b) \end{pmatrix} \right\rangle \tag{11}$$

We refer to this as the *target partition* for this exercise. We want to restrict attention to those positions in the target partition that are consistent with the initial assertions:

$$\neg \mathrm{P}\, p(b) \;\wedge\; (\mathrm{P}\, p(a_1) \wedge \mathrm{P}\, \neg p(a_1)) \;\wedge\; (\mathrm{P}\, p(a_2) \wedge \mathrm{P}\, \neg p(a_2)) \tag{12}$$

One might ask why anyone would be interested in representing the rules of a car park at this level of precision. One answer is to say that a precise specification may be essential if we were assigned the task of implementing an automated system for controlling access to the car park, for instance. But really the point is this: the car park is just a representative example for a wide range of similar problems that arise in practice. Instead of controlling who may put cars in a car park, imagine that the car park is a computer file of some kind, and that $p(x)$ represents that data entry x is stored in the file. The task is to specify which agents (computer agents or human) are to be permitted to insert and delete data entries in the file. Suppose, let us say, that at midnight every night a set of electronic transactions will be generated automatically according to the data entries present in the file at that time. This is not at all a fanciful suggestion. In such cases, it will be essential to specify with precision which agents are permitted to enter which data entries and in which combinations. The gatekeeper g of the car park example corresponds to the 'file monitor' (human or computer agent) which controls access to the file.

The user interface of the Norman-G program is based on the display of a graph. Nodes of the graph are labelled by act expressions. The user interacts with the program by colouring the nodes: blue to signify that the act or state of affairs represented by that expression is permitted, and red to signify that it is not permitted. For the car park example, the graph contains nodes labelled $p(b)$, $\neg p(b)$, $p(a_1)$, $\neg p(a_1)$, $p(a_2)$, $\neg p(a_2)$, $\mathrm{E}_a\, p(a_1)$, $p(a_1) \wedge \neg \mathrm{E}_a\, p(a_1)$, and a great many others besides. The initial description (12) of what is permitted/not permitted in the car park is entered by colouring node $p(b)$ red—node $\neg p(b)$ becomes blue automatically—and nodes $p(a_1)$, $\neg p(a_1)$, $p(a_2)$, and $\neg p(a_2)$ blue. Edges of the graph represent implications between the node expressions, as determined by the logical principles employed. For example, there is an edge from the node labelled $\mathrm{E}_a\, p(a_1)$ to the node labelled $p(a_1)$. If node $\mathrm{E}_a\, p(a_1)$ is coloured blue (say) then node $p(a_1)$, and all other nodes implied by $\mathrm{E}_a\, p(a_1)$, are coloured blue automatically. The propagation of colours through the graph as individual nodes are coloured gives a visualisation of the underlying inference methods. A position in the target partition is uniquely determined when the graph is completely coloured. One can leave the graph partially uncoloured where there is ambiguity or doubt about some of the detailed choices.

The dynamic behaviour of the graphical interface is difficult to illustrate in the space available here. Instead we show the operation of Norman-G in *dialogue mode*, whereby the user, having coloured parts of the graph to represent initial assertions, responds to questions generated by the program. The user is able to postpone giving an answer, or to terminate the dialogue at any time.

We now show the transcript of such a dialogue. To keep the illustration manageable we ignore the agents c and g for the time being and consider the simpler target partition:

$$\left\langle \pm \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} p(a_1) \\ p(a_2) \\ p(b) \end{pmatrix} \right\rangle \tag{13}$$

The dialogue for the original target partition (11) proceeds in exactly the same way as shown, though is obviously longer than the one shown here.

The left hand column of the transcript shows the question generated by the Norman-G program and the answer given in response. The right hand column shows the corresponding assertion that is made and displayed on the graph. The three expressions above the horizontal line are the initial assertions made before the dialogue is invoked. (The dialogue can also be invoked with an empty set of initial assertions, or at any other point of colouring the graph.)

		$\neg\mathrm{P}\, p(b)$	
		$\mathrm{P}\, p(a_1) \wedge \mathrm{P}\neg p(a_1)$	
		$\mathrm{P}\, p(a_2) \wedge \mathrm{P}\neg p(a_2)$	
$\mathrm{P}\,(p(a_1) \wedge p(a_2))$?	—n	$\neg\mathrm{P}\,(p(a_1) \wedge p(a_2))$	Note [1]
$\mathrm{P}\,(\neg p(a_1) \wedge \neg p(a_2))$?	—y	$\mathrm{P}\,(\neg p(a_1) \wedge \neg p(a_2))$	
$\mathrm{P}\,\mathrm{E}_a\, p(a_1)$?	—y	$\mathrm{P}\,\mathrm{E}_a\, p(a_1)$	
$\mathrm{P}\,(p(a_1) \wedge \neg\mathrm{E}_a\, p(a_1))$?	—n	$\neg\mathrm{P}\,(p(a_1) \wedge \neg\mathrm{E}_a\, p(a_1))$	Note [2]
$\mathrm{P}\,\mathrm{E}_a\neg p(a_1)$?	—y	$\mathrm{P}\,\mathrm{E}_a\neg p(a_1)$	
$\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_a\neg p(a_1))$?	—n	$\neg\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_a\neg p(a_1))$	
$\mathrm{P}\,\mathrm{E}_b\, p(a_1)$?	—n	$\neg\mathrm{P}\,\mathrm{E}_b\, p(a_1)$	Note [3]
$\mathrm{P}\,\mathrm{E}_b\neg p(a_1)$?	—n	$\neg\mathrm{P}\,\mathrm{E}_b\neg p(a_1)$	
`Treat` $p(a_2)$ `like` $p(a_1)$		$\mathrm{P}\,\mathrm{E}_a\, p(a_2)$	Note [4]
		$\neg\mathrm{P}\,(p(a_2) \wedge \neg\mathrm{E}_a\, p(a_2))$	
		$\mathrm{P}\,\mathrm{E}_a\neg p(a_2)$	
		$\neg\mathrm{P}\,(\neg p(a_2) \wedge \neg\mathrm{E}_a\neg p(a_2))$	
		$\neg\mathrm{P}\,\mathrm{E}_b\, p(a_2)$	
		$\neg\mathrm{P}\,\mathrm{E}_b\neg p(a_2)$	
$\mathrm{P}\,\mathrm{E}_a\neg p(b)$?	—n	$\neg\mathrm{P}\,\mathrm{E}_a\neg p(b)$	Note [5]

Note [1] We are assuming that it would not be permitted for both of administrator a's cars to be parked at the same time. This would need to be checked with the car park authorities, or left undetermined if it were not important. One purpose of the analysis is to identify points of detail that may have remained undetected otherwise.

Note [2] We are supposing for the purpose of the example that if one of the administrator a's cars is parked, then a must be at least one of those responsible for the car's being parked, i.e. that $\mathrm{O}\,(p(a_1) \rightarrow \mathrm{E}_a\, p(a_1))$ holds. It might be tempting to read this as saying that if car_a1 is parked ($p(a_1)$ is true) then it must have been the administrator a who parked it. But as discussed in the introductory sections, the expression $\mathrm{E}_a\, p(a_1)$ does not necessarily signify that a parks car_a1. a may bring about $p(a_1)$ in some quite different way.

Note [3] One might suppose that the answer to this question is already implied by answers given earlier, in particular at Note [2]. After all, one might argue that if $\mathrm{O}\,(p(a_1) \rightarrow \mathrm{E}_a\, p(a_1))$ holds then how could b be permitted to park a's car. But this does not follow in the logics employed, as will be discussed in detail later. For the time being, we have answered this question in the negative since it seems right to say that the banned administrator b is not permitted to park a's car. We comment further on this point in Section 5.

Note [4] Norman-G provides a range of shortcuts for the user's convenience. (The actual syntax is slightly different from that shown here.) Another common shortcut, for example, is to specify that expressions containing agents a and b, or specific patterns of such expressions, are to be treated in the same way: what holds for a holds for b, and vice versa. These are not features of the language or of the logic but merely shorthand devices for entering commonly occurring patterns of assertions.

Note [5] The answer to this question is far from obvious. We assume it is 'no' for simplicity. Similar comments apply to this question as to the one at Note [3].

At this point (i.e. at the end of the transcript above) it turns out that we have identified a unique position in both

$$\left\langle \pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} p(a_1) \\ p(a_2) \end{pmatrix} \right\rangle \quad \text{and} \quad \left\langle \pm\, \mathrm{O} \pm \mathrm{E}_a \pm \begin{pmatrix} p(a_1) \\ p(a_2) \\ p(b) \end{pmatrix} \right\rangle$$

This is obvious in the graphical display of Norman-G but difficult to demonstrate succinctly here. It now remains to consider b's bringing it about that $\neg p(b)$, i.e. to determine the appropriate positions in

$$\left\langle \pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm p(b) \right\rangle, \quad \left\langle \pm\, \mathrm{O} \pm \mathrm{E}_b \pm \begin{pmatrix} p(a_1) \\ p(a_2) \\ p(b) \end{pmatrix} \right\rangle, \quad \left\langle \pm\, \mathrm{O} \pm \begin{pmatrix} \mathrm{E}_a \\ \mathrm{E}_b \end{pmatrix} \pm \begin{pmatrix} p(a_1) \\ p(a_2) \\ p(b) \end{pmatrix} \right\rangle$$

The expressions grow too large to display side by side so we show only the questions and answers and not the asserted facts as well. The transcript continues:

$$
\begin{array}{ll|}
\mathrm{P}\,\mathrm{E}_b\neg p(b)\ ? & \text{—y} \\
\mathrm{P}\,(\neg p(b) \wedge \neg\mathrm{E}_b\neg p(b))\ ? & \text{—y} \\
\mathrm{P}\,(p(a_1) \wedge \neg\mathrm{E}_b\, p(a_1) \wedge \mathrm{E}_b\neg p(b))\ ? & \text{—y} \\
\mathrm{P}\,(p(a_1) \wedge \neg\mathrm{E}_b\, p(a_1) \wedge \neg p(b) \wedge \neg\mathrm{E}_b\neg p(b))\ ? & \text{—y} \\
\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_b\neg p(a_1) \wedge \mathrm{E}_b\neg p(b))\ ? & \text{—y} \\
\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_b\neg p(a_1) \wedge \neg p(b) \wedge \neg\mathrm{E}_b\neg p(b))\ ? & \text{—y}
\end{array}
$$

Four more assertions follow from the earlier declaration that $p(a_2)$ is to be treated like $p(a_1)$:

$$
\begin{array}{|r}
\mathrm{P}\,(p(a_2) \wedge \neg\mathrm{E}_b\, p(a_2) \wedge \mathrm{E}_b\neg p(b)) \\
\mathrm{P}\,(p(a_2) \wedge \neg\mathrm{E}_b\, p(a_2) \wedge \neg p(b) \wedge \neg\mathrm{E}_b\neg p(b)) \\
\mathrm{P}\,(\neg p(a_2) \wedge \neg\mathrm{E}_b\neg p(a_2) \wedge \mathrm{E}_b\neg p(b)) \\
\mathrm{P}\,(\neg p(a_2) \wedge \neg\mathrm{E}_b\neg p(a_2) \wedge \neg p(b) \wedge \neg\mathrm{E}_b\neg p(b))
\end{array}
$$

The system can detect that these follow from earlier inputs and does not need to generate questions about them. The rest of the transcript contains just two further questions:

$$
\begin{array}{ll|}
\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_b\neg p(a_1) \wedge \neg p(a_2) \wedge \neg\mathrm{E}_b\neg p(a_2) \wedge \mathrm{E}_b\neg p(b))\ ? & \text{—y} \\
\mathrm{P}\,(\neg p(a_1) \wedge \neg\mathrm{E}_b\neg p(a_1) \wedge \neg p(a_2) \wedge \neg\mathrm{E}_b\neg p(a_2) \wedge \neg p(b) \wedge \neg\mathrm{E}_b\neg p(b))\ ? & \text{—y}
\end{array}
$$

At this stage we have identified a unique position in the simplified target position (13). Identification of a position from the original target partition (11) continues in exactly the same way. In this example, depending on the answers given, there are about a dozen or so further questions in a typical complete dialogue. Norman-G provides a range of features for keeping track of progress and examining which of the partitions remain to be explored as the analysis proceeds.

5. Permitted to bring about

We turn now to the remarks made at Note [3] of the illustrative run in the previous section. Here we had the question '$\mathrm{P}\,\mathrm{E}_b\, p(a_1)$?' to which we answered 'no'. Intuitively it seems right to say that the banned administrator b is not permitted to park the administrator a's car, or more precisely, to see to it or bring it about that the administrator a's car is parked. But is this correctly represented by $\neg\mathrm{P}\,\mathrm{E}_b\, p(a_1)$? If we were to leave the question unanswered, Norman-G would eventually generate another question later in the run:

$$\mathrm{P}\,(\mathrm{E}_a\, p(a_1) \wedge \mathrm{E}_b\, p(a_1)) \quad ?$$

Are the administrator a and the banned administrator b, perhaps acting together, permitted to park the administrator a's car? It seems that this would be within the rules of the car park. We could certainly create other similar examples where this would be the case. We have $\mathrm{P}\,(\mathrm{E}_a\, p(a_1) \wedge \mathrm{E}_b\, p(a_1))$ and the logic contains the valid schema $\mathrm{P}\,(A \wedge B) \to \mathrm{P}\, B$, and so we have also:

$$\vdash \quad \mathrm{P}\,(\mathrm{E}_a\, p(a_1) \wedge \mathrm{E}_b\, p(a_1)) \to \mathrm{P}\mathrm{E}_b\, p(a_1)$$

It seems that we should have answered the question at Note [3] in the *affirmative*.

The same point arises frequently. At Note [5] in the same transcript we also answered '$\mathrm{P}\,\mathrm{E}_a \neg p(b)$?' with 'no' (remarking only that the answer was not obvious). Later in the same run, we asserted $\mathrm{P}\,\mathrm{E}_b \neg p(b)$, which seems quite clearcut: surely it must be permitted for the banned administrator b to see to it that his (banned) car is not parked in the car park. But then it is easy to imagine circumstances in which $\mathrm{P}\,(\mathrm{E}_a \neg p(b) \land \mathrm{E}_b \neg p(b))$ should hold, perhaps when a and b act together. And then by the same argument as above, it also follows that $\mathrm{P}\,\mathrm{E}_a \neg p(b)$ is true in the car park after all.

The erroneous reading of such expressions seems very easy to slip into. For instance, Lindahl uses the example of two adjoining properties, one of which is owned by an agent called John. When discussing the possible normative relations between John and his neighbour in regard to various kinds of acts, Lindahl [19, pp93–94] suggests: "Finally, ... illustrates a case in which John is completely unauthorized to influence the situation (since it is no business of his): John may neither bring about nor prevent the main building on his neighbour's property being painted white." In the light of the previous discussion, this is unlikely to be correct. More likely, it is permitted that John and his neighbour together act in such a way that both bring about that the neighbour's property is painted white. It then follows that John *is* permitted to influence the situation, even though the colour of his neighbour's house is no business of his.

The conjunction $\mathrm{E}_a\, p(a_1) \land \mathrm{E}_b\, p(a_1)$ may but does not necessarily signify (intentional) joint action by a and b. It could be that both a and b choose independently to see to it that $p(a_1)$. It could be that both bring it about that $p(a_1)$ by chance. It could be that one does it intentionally and the other by chance. Nor does the conjunction represent a composite agent a-and-b-together.

One might feel uncomfortable with the idea that two distinct agents, acting independently, can both be held responsible for bringing about the same state of affairs. However, to deny the possibility is tantamount to accepting as valid the following additional schema:

$$\mathrm{E}_x A \rightarrow \neg \mathrm{E}_y A \qquad \text{for all pairs of agents } x \neq y$$

We call this the 'single agent assumption'.

It is very easy to incorporate such a schema into the framework, and into the Norman-G program, should it be desired. It actually simplifies the generation of the graph and reduces the number of nodes substantially. Special instances of the schema can be represented if desired using an auxiliary modality provided in Norman-G (see [34, Section 6]). However, none of the versions of 'brings it about' or 'sees to it that' cited earlier in Section 1 has a 'single agent assumption'. Nor is such a property compatible with any of the semantical accounts given in those works.

What the example illustrates is that the notion of permission represented by P is very much weaker than what is meant by permission in ordinary discourse. Of course, this is a very familiar point, but one worth reiterating in this setting. If I am told that I am permitted to bring about some state of affairs I do not assume that every conceivable means of achieving this end is open to me. This is a familiar point. However, unless there were a specific instruction to the contrary, I would expect that this permission does not depend on my acting in concert with other, unspecified, agents.

In order to guard against possible common misunderstandings, therefore, and because the need arises so frequently in practice, we have introduced a special abbreviation, as follows. When the set of agents in the representation is $\{x_1, \ldots, x_n\}$

$$\mathrm{P!E}_{x_i} A \quad \stackrel{\text{def}}{=} \quad \mathrm{P}\,(\mathrm{E}_{x_i} A \land \neg \mathrm{E}_{x_1} A \land \ldots \land \neg \mathrm{E}_{x_{i-1}} A \land \neg \mathrm{E}_{x_{i+1}} A \land \ldots \land \neg \mathrm{E}_{x_n} A)$$

$\mathrm{P!E}_x F$ is intended to represent that agent x is permitted to bring it about that F, and that this permission is not dependent on the actions of any other agents. $\mathrm{P!E}_x$ should be regarded as an atomic symbol in such expressions: $\mathrm{P!}\neg\mathrm{E}_x F$ is not meaningful.

The notation could be generalised to cover other combinations of agents, e.g. expressions such as $\mathrm{P}\,(\mathrm{E}_{x_1} A \wedge \mathrm{E}_{x_2} A \wedge \neg\mathrm{E}_{x_3} \wedge \ldots \wedge \neg\mathrm{E}_{x_n} A)$, but this does not seem to be particularly useful or frequently required and so we have not bothered with it.

6. Interpersonal control positions

For certain purposes we need to consider extending the initial characterisation of the class of act expressions from which the normative positions are constructed. Some regulations pertain not to *individual* agent positions of the form $\mathrm{E}_x F$, but to what have been termed (see [25, Ch.3]) interpersonal *control* positions, e.g. of type $\mathrm{E}_x\,\mathrm{E}_y F$ or $\mathrm{E}_x \neg\mathrm{E}_y F$. Indeed, the ability to iterate action operators in this way is one of the generally perceived benefits of employing this approach to the treatment of action. We now consider how it might work out in the representation of a specific example.

For instance, we have a rule in the car park that the banned administrator b's car may not be parked, $\neg\mathrm{P}\,p(b)$. It follows in the logic that the banned administrator b may not see to it that his car is parked, $\neg\mathrm{P}\,\mathrm{E}_b\,p(b)$. Consider now the responsibilities of the gatekeeper. It seems reasonable to say that the gatekeeper g is permitted to see to it that the banned administrator does not park his car, or more generally that $\mathrm{P}\,\mathrm{E}_g \neg\mathrm{E}_b\,p(b)$ holds according to the rules of the car park. (One might even be tempted to say that there is an *obligation* on the gatekeeper g to see to it that b does not park his car. However, as discussed in the introductory section, an expression of the form $\mathrm{O}\,\mathrm{E}_g \neg\mathrm{E}_b\,p(b)$ does not represent such an obligation adequately. We will not discuss it further.) One would surely not *insist*, however, that g sees to it that $\mathrm{E}_b \neg p(b)$—surely we would expect that $\mathrm{P}\neg\mathrm{E}_g\,\mathrm{E}_b \neg p(b)$ holds in the car park. Are there any other possibilities?

We are led to examine normative positions of the following form:

$$\left\langle \pm\,\mathrm{O} \pm \begin{pmatrix} \mathrm{E}_x \\ \vdots \\ \mathrm{E}_y \end{pmatrix} \pm \begin{pmatrix} \mathrm{E}_x \\ \vdots \\ \mathrm{E}_y \end{pmatrix} \pm \begin{pmatrix} F \\ \vdots \\ G \end{pmatrix} \right\rangle \tag{14}$$

(x and y need not be distinct).

The general principles and methods of construction are exactly the same as outlined in Section 3, though more complicated in application. For simplicity we consider here only the simplest case:

$$\langle \pm\ \mathrm{O} \pm \mathrm{E}_x \pm \mathrm{E}_y \pm F \rangle$$

As usual

$$\langle \pm\ \mathrm{O} \pm \mathrm{E}_x \pm \mathrm{E}_y \pm F \rangle \quad \stackrel{\text{def}}{=} \quad [[\pm\,\mathrm{O} \pm \langle \pm\ \mathrm{E}_x \pm \mathrm{E}_y \pm F \rangle]] \tag{15}$$

but there is some choice as to how we define the set of 'control positions' $\langle \pm\,\mathrm{E}_x \pm \mathrm{E}_y \pm F \rangle$. We choose to employ not the most refined set of positions that could be constructed (which depends on detailed properties of the E_x operator) but the most refined set of control positions containing an atom or the negation of an atom inside the scope of the innermost action operator:

$$\langle \pm\ \mathrm{E}_x \pm \mathrm{E}_y \pm F \rangle \quad \stackrel{\text{def}}{=} \quad \langle \pm\, \mathrm{E}_x \pm\ (\mathrm{E}_y\, F) \rangle \cdot \langle \pm\, \mathrm{E}_x \pm\ (\mathrm{E}_y \neg F) \rangle \cdot \langle \pm\, \mathrm{E}_y \pm F \rangle \tag{16}$$

This seems to provide a reasonable compromise between precision and usability.

The positions of type (16) consistent with F (equivalently, those that partition F) are the following:

- $\mathrm{E}_x\, \mathrm{E}_y F \wedge \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $\mathrm{E}_x\, \mathrm{E}_y F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $\mathrm{E}_y F \wedge \neg \mathrm{E}_x\, \mathrm{E}_y F \wedge \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $\mathrm{E}_y F \wedge \neg \mathrm{E}_x\, \mathrm{E}_y F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $F \wedge \mathrm{E}_x \neg \mathrm{E}_y F \wedge \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $F \wedge \mathrm{E}_x \neg \mathrm{E}_y F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y F \wedge \mathrm{E}_x \neg \mathrm{E}_y \neg F$
- $F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y F \wedge \neg \mathrm{E}_x \neg \mathrm{E}_y \neg F$

There is a symmetric list of positions consistent with (and that partition) $\neg F$.

In some versions of action/agency, notably 'stit', it is not meaningful to say x 'sees to it' that y 'sees to it' that F (see e.g. the discussion in [2]). We do not build any such assumption into the general framework. It can be incorporated very easily if desired (since it simply reduces the number of nodes in the graph). If the 'stit' version is adopted for E_x, then the list of positions above (those that partition F) can be simplified. The first two are eliminated, and the next two can be simplified by deleting the second conjunct.

We can also consider the special case where $x = y$. Further simplifications are then possible because it is natural to adopt as valid the following extra property: $\mathrm{E}_x\, \mathrm{E}_x A \leftrightarrow \mathrm{E}_x A$. The details of this special case are straightforward and we leave them to the reader.

If we look now to the car park and the question of the gatekeeper's control over the banned administrator then we need to consider the positions in partition (15) that are consistent with $\mathrm{P}\,(\mathrm{E}_g \neg \mathrm{E}_b\, p(b) \wedge \neg p(b))$ (supposing, as we do, that this is settled). According to the methods summarised in Section 3 (see [33, 34] for the details of the procedure) we must consider which of the following set of control positions are permitted. At least one of them must be permitted, but there may be more than one.

(a) $\mathrm{E}_g\, \mathrm{E}_b \neg p(b) \wedge \mathrm{E}_g \neg \mathrm{E}_b\, p(b)$

(b) $\mathrm{E}_b \neg p(b) \wedge \neg \mathrm{E}_g\, \mathrm{E}_b \neg p(b) \wedge \mathrm{E}_g \neg \mathrm{E}_b\, p(b)$

(c) $\neg p(b) \wedge \mathrm{E}_g \neg \mathrm{E}_b \neg p(b) \wedge \mathrm{E}_g \neg \mathrm{E}_b\, p(b)$

(d) $\neg p(b) \wedge \neg \mathrm{E}_g \neg \mathrm{E}_b \neg p(b) \wedge \mathrm{E}_g \neg \mathrm{E}_b\, p(b)$

In each case b's car is not parked and g's actions are such as to ensure that b does not see to it that his car is parked. In each case however the interaction between g and b is subtly different. Which are permitted in the car park (as we imagine it to be)?

The first of these positions (a) is not present when the 'stit' reading is given to the action operator. In other versions where the first conjunct is meaningful, there seems to be nothing problematic about asserting that the whole conjunction is permitted.

In (b), the second conjunct is eliminated in the 'stit' reading. Again, there seems to be nothing problematic about saying that it is permitted both that b sees to it that his car is not parked, without g's influence, while g sees to it that b does not, maybe cannot, park.

The difference between (c) and (d) is that in (c) g acts in such a way as to prevent b from ensuring that his car is not parked, while in (d) g merely prevents b from parking (or rather, prevents b from seeing to it that his car is parked).

It is not easy to give a concise reading to these expressions. The above suggestions are just approximations. A more careful reading of each expression would be quite involved. And this is one of the main points we are trying to make in the paper. Implementation of these more complex forms (and others, such as those provided by further iteration of the action operators) can be accomplished. One obvious drawback is the huge number of new positions created. Even with a relatively small number of agents and states of affairs the representation within Norman-G grows enormously. We need to consider not only the gatekeeper g's influence over the banned administrator b, but also the banned administrator b's influence over the gatekeeper g. And this must be repeated for every distinct pair of agents.

But keeping track of all these possibilities is not the main problem: that is what machines are for, after all. What is not clear is whether there is any real value in providing this level of analysis. As the example illustrates, deciphering what these complex expressions signify is far from straightforward. One may be offering a level of precision that is simply unusable in practice. We remain unconvinced. We need to acquire more experience with a wider range of examples.

One of the main difficulties in deciphering the control positions is in interpreting negatives. It is hard to decide what 'x does not see to it that F is not the case' actually means. This is made all the harder by the fact that it is unclear what 'not being parked' means exactly: do we mean that the car was never in the car park, or that it was in the car park and was then removed? This can make a big difference. We turn to that next.

7. Permitted to sustain and permitted to bring about

It is very easy to imagine a car park in which the gatekeeper g is permitted to prevent a banned car from parking but not permitted to remove a car even if it is illegally parked. With the presently available resources all we can say is that $\mathrm{PE}_g\neg p(b)$ —the gatekeeper is permitted to see to it that b's car is not parked.

One way to enrich the representation is to leave the logical components alone and try to be more creative about the choice of propositional symbols. For example, instead of speaking only about a car c being parked, represented $p(c)$, we could introduce symbols $ne(c)$ and $r(c)$ to represent that car c never entered the car park and was removed from the car park, respectively. We could then speak directly about an agent's seeing to it that a car is removed from the car park or seeing to it that a car never enters the car park. To capture something of the intended interpretation of these new symbols we would need further (non-logical) axioms in the representation: $\vdash\ ne(c) \rightarrow \neg r(c)$, and $\vdash\ \neg p(c) \leftrightarrow (ne(c) \vee r(c))$. Such axioms can be expressed in Norman-G using an auxiliary modality provided for this purpose (see [34, Section 6]). However we have never worked out an example in full in this style and so are unable to comment on whether there are any hidden difficulties with it. We leave that discussion for another paper.

A second possible approach is to follow a suggestion by Risto Hilpinen in [7]. He puts forward an account with two components: first, he makes explicit the idea that actions are associated with transitions between states; and second, to provide the 'negative condition' required to capture the notion of agency, he uses a structure which distinguishes transitions corresponding to the agent's activity from transitions corresponding to the agent's inactivity. There are then eight possible modes of action, and because of the symmetry between A and $\neg A$, four basic forms to consider:

- x brings it about that A ($\neg A$ to A, x active);
- x lets it become the case that A ($\neg A$ to A, x inactive);
- x sustains the case that A (A to A, x active);
- x lets it remain the case that A (A to A, x inactive).

The first two, x brings it about that A and x lets it become the case that A, correspond to a transition from a state where $\neg A$ to a state where A. The first of these is a kind of bringing about that A by agent x; the second corresponds to inactivity by x (with respect to A)—here the agent x lets nature take its own course, as it were. The last two, sustains the case that A and lets it remain the case that A, correspond to a transition from a state where A to a state where A. Again, the first of them is a kind of bringing about that A by agent x; the second corresponds to inactivity by x (with respect to A).

As discussed by Hilpinen there are still a number of fundamental problems to resolve with this account. Moreover, not discussed by Hilpinen, the picture is considerably more complicated when there are the actions of other agents to take into account and not just the effect of nature's taking its course. For these reasons we have not adopted Hilpinen's suggested treatment in its entirety. We do find it extremely useful, however, to be able to distinguish between an agent x's bringing it about that A (implying that it was not the case that A before x acted) and x's sustaining the case that A (implying that A was the case before x acted). We introduce two new (relativised) operators E_x^+ and E_x^- to represent these two notions respectively. Both have exactly the same logical properties as E_x since they are both forms of 'bringing it about' in the same sense. They are obviously mutually exclusive, and so the following relationships hold:

E1. $\mathrm{E}_x A \leftrightarrow (\mathrm{E}_x^+ A \ \vee\ \mathrm{E}_x^- A)$

E2. $\mathrm{E}_x^+ A \rightarrow \neg \mathrm{E}_x^- A$

We do not (at present) distinguish between letting it remain the case that A and letting it become the case that A since we do not find this so useful. Both remain represented by $\neg \mathrm{E}_x A$.

The partitions generated by these various act expressions may be summarised diagrammatically as follows:

A			$\neg A$		
$\mathrm{E}_x A$		$A \wedge \neg \mathrm{E}_x A$	$\mathrm{E}_x \neg A$		$\neg A \wedge \neg \mathrm{E}_x \neg A$
$\mathrm{E}_x^+ A$	$\mathrm{E}_x^- A$	$A \wedge \neg \mathrm{E}_x A$	$\mathrm{E}_x^+ \neg A$	$\mathrm{E}_x^- \neg A$	$\neg A \wedge \neg \mathrm{E}_x \neg A$

One further property is required. The following scheme is valid, for any pair of agents x and y:

E3. $\quad \mathrm{E}_x^+ A \rightarrow \neg \mathrm{E}_y^- A$

These extensions are all easily incorporated into the Norman-G implementation.

With these additional resources we are able to distinguish between seeing to it that a car not parked in the car park remains not parked (approximately, preventing a car from entering), and seeing to it that a car which was parked is no longer parked (approximately, removing it). In the (imaginary) car park, the first is permitted for the gatekeeper g, the second is not:

$$\mathrm{P}\,\mathrm{E}_g^- \neg p(b) \;\wedge\; \neg \mathrm{P}\,\mathrm{E}_g^+ \neg p(b)$$

Of course what we really mean to say (cf. the discussion in Section 5) is this:

$$\mathrm{P!E}_g^- \neg p(b) \;\wedge\; \neg \mathrm{P!E}_g^+ \neg p(b)$$

The gatekeeper's being permitted to prevent the banned car from entering the car park is not dependent on the actions of other agents. $\mathrm{P!E}_x^-$ and $\mathrm{P!E}_x^+$ are defined in analogous fashion to the $\mathrm{P!E}_x$ notation introduced in Section 5.

8. Conclusion

The theory of normative positions, with its associated inference methods and implementation techniques, provides the promise of an extremely powerful and precise tool to aid in the disambiguation and refinement of regulatory provisions, of many different kinds. In practice, application of the theory can be surprisingly difficult, largely we suggest, because of the very high level of abstraction built into its treatment of action and agency. It remains unclear to us whether difficulties in overcoming representational problems are reflections of fundamental limitations in this approach to the treatment of action, or simply due to lack of familiarity with applying the tools for purposes of representation for which they were not originally designed. We have not been able to come to any definite conclusion. Many of the more obvious difficulties can be overcome by simple extensions of the kind described in the second part of this paper. Further experience is required to evaluate their effectiveness properly.

One problem we have not discussed in this paper is the representation of conditional structures. Clearly these are essential if the theory is to have any wide applicability. However, the combination of conditionals with this approach to the treatment of action is very problematic. Unless all actions can be assumed to be instantaneous (an assumption which is made in some of the versions cited in the paper) there is a great deal to sort out. If we say that Fiona is permitted to park her car if, and only if, it is raining, and if the action of parking takes some significant length of time, do we check that it is raining when she begins to park, or when she completes the job? Do we require it to be raining throughout the entire process? The first of these seems the most natural but that would require far-reaching adjustments to the logic of action that has been employed. Resolution of such questions is very far from straightforward. We intend to comment in more detail about the difficulties, and alternative approaches, in another paper.

References

[1] Åqvist, L.: A new approach to the logical theory of actions and causality, in: *Logical Theory and Semantic Analysis* (S. Stenlund, Ed.), number 63 in Synthese Library, D. Reidel, Dordrecht, 1974, 73–91.

[2] Belnap, N., Perloff, M.: Seeing to it that: a canonical form for agentives, *Theoria*, **54**, 1988, 175–199.

[3] Belnap, N., Perloff, M.: The way of the agent, *Studia Logica*, **51**, 1992, 463–484.

[4] Chellas, B. F.: *The Logical Form of Imperatives*, Dissertation, Stanford University, 1969.

[5] Chellas, B. F.: *Modal Logic—An Introduction*, Cambridge University Press, 1980.

[6] Elgesem, D.: *Action Theory and Modal Logic*, Doctoral thesis, Department of Philosophy, University of Oslo, 1992.

[7] Hilpinen, R.: On Action and Agency, in: *Logic, Action and Cognition—Essays in Philosophical Logic* (E. Ejerhed, S. Lindstr¨om, Eds.), vol. 2 of *Trends in Logic, Studia Logica Library*, Kluwer Academic Publishers, Dordrecht, 1997, 3–27.

[8] Hohfeld, W. N.: Some fundamental legal conceptions as applied in judicial reasoning, *Yale Law Journal*, **23**, 1913. Reprinted with revisions as: *Some Fundamental Legal Conceptions as Applied in Judicial Reasoning, and Other Legal Essays* (W. W. Cook, Ed.), Yale University Press, 1919, 1923, 1964, and as: *Some Fundamental Legal Conceptions as Applied in Judicial Reasoning, and Other Legal Essays* (W. C. Wheeler, Ed.), Greenwood Press, 1978.

[9] Horty, J. F.: Agency and obligation, *Synthese*, **108**, 1996, 269–307.

[10] Horty, J. F.: Combining agency and obligation (Preliminary version), in: *Deontic Logic, Agency and Normative Systems—Proc. DEON'96: 3rd International Workshop on Deontic Logic in Computer Science, Sesimbra (Portugal)* (M. A. Brown, J. Carmo, Eds.), Workshops in Computing Series, Springer-Verlag, Berlin-Heidelberg, 1996, 98–122.

[11] Horty, J. F.: *Agency and Deontic Logic*, Oxford University Press, 2000.

[12] Horty, J. F., Belnap, N.: The deliberative stit: a study of action, omission, ability, and obligation, *Journal of Philosophical Logic*, **24**, 1995, 583–644.

[13] Jones, A. J. I., Sergot, M. J.: Formal Specification of Security Requirements using the Theory of Normative Positions, in: *Computer Security—ESORICS 92* (Y. Deswarte, G. Eizenberg, J.-J. Quisquater, Eds.), number 648 in Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 1992, 103–121.

[14] Jones, A. J. I., Sergot, M. J.: On the Characterisation of Law and Computer Systems: The Normative Systems Perspective, in: *Deontic Logic in Computer Science: Normative System Specifi cation* (J.-J. C. Meyer, R. J. Wieringa, Eds.), chapter 12, John Wiley & Sons, Chichester, England, 1993, 275–307.

[15] Jones, A. J. I., Sergot, M. J.: A formal characterisation of institutionalised power, *Journal of the IGPL*, **4**(3), 1996, 429–445. Reprinted in: *Normative Systems in Legal and Moral Theory. Festschrift for Carlos E. Alchourrón and Eugenio Bulygin* (E. G. Valdés, W. Krawietz, G. H. von Wright, R. Zimmerling, Eds.), Duncker & Humboldt, Berlin, 1997, 349–367.

[16] Kanger, S.: New foundations for ethical theory, in: *Deontic Logic: Introductory and Systematic Readings* (R. Hilpinen, Ed.), D. Reidel, Dordrecht, 1971, 36–58. Originally published as Technical Report, Stockholm University, 1957

[17] Kanger, S., Kanger, H.: Rights and Parliamentarism, *Theoria*, **32**, 1966, 85–115.

[18] Lee, R. M.: Bureaucracies as deontic systems, *ACM Transactions on Offi ce Information Systems*, **6**(2), 1988, 87–108.

[19] Lindahl, L.: *Position and Change—A Study in Law and Logic*, Number 112 in Synthese Library, D. Reidel, Dordrecht, 1977.

[20] Lindahl, L.: Stig Kanger's Theory of Rights, *9th International Congress of Logic, Methodology and Philosophy of Science. Stig Kanger Memorial Symposium on the Logic of Rights and Choices, Uppsala*, August 1992.

[21] Makinson, D.: On the formal representation of rights relations, *Journal of Philosophical Logic*, **15**, 1986, 403–425.

[22] Perloff, M.: 'Stit' and the language of agency, *Synthese*, **86**, 1991, 379–408.

[23] Pörn, I.: *The Logic of Power*, Blackwells, Oxford, 1970.

[24] Pörn, I.: Some basic concepts of action, in: *Logical Theory and Semantic Analysis* (S. Stenlund, Ed.), number 63 in Synthese Library, D. Reidel, Dordrecht, 1974, 93–101.

[25] Pörn, I.: *Action Theory and Social Science: Some Formal Models*, Number 120 in Synthese Library, D. Reidel, Dordrecht, 1977.

[26] Pörn, I.: On the nature of a social order, in: *Logic, Methodology and Philosophy of Science VIII* (J. E. Fenstad, et al., Eds.), Elsevier Science Publishers, 1989, 553–567.

[27] Santos, F., Carmo, J.: Indirect action, influence and responsibility, in: *Deontic Logic, Agency and Normative Systems—Proc. DEON'96: 3rd International Workshop on Deontic Logic in Computer Science, Sesimbra (Portugal)* (M. A. Brown, J. Carmo, Eds.), Workshops in Computing Series, Springer-Verlag, Berlin-Heidelberg, 1996, 194–215.

[28] Santos, F., Jones, A. J. I., Carmo, J.: Action concepts for describing organised interaction, in: *Proc. 13th Annual Hawaii International Conf. on System Sciences*, vol. V, IEEE Computer Society Press, Los Alamitos, California, 1997.

[29] Segerberg, K.: Routines, *Synthese*, **65**, 1985, 185–210.

[30] Segerberg, K.: Bringing it about, *Journal of Philosophical Logic*, **18**, 1989, 327–347.

[31] Segerberg, K.: Getting started: Beginnings in the logic of action, *Studia Logica*, **51**, 1992, 347–378.

[32] Sergot, M. J.: *A Computational Theory of Normative Positions. II Non-regular logics*, Technical report, Department of Computing, Imperial College, January 1996.

[33] Sergot, M. J.: Normative Positions, in: *Norms, Logics and Information Systems* (P. McNamara, H. Prakken, Eds.), IOS Press, Amsterdam, 1998, 289–308.

[34] Sergot, M. J.: A Computational Theory of Normative Positions, *ACM Transactions on Computational Logic*, 2001. In press