Robust normative systems and a logic of norm compliance

THOMAS ÅGOTNES, *Department of Information Science and Media Studies, University of Bergen, PB. 7802, N-5020 Bergen, Norway*
E-mail: thomas.agotnes@infomedia.uib.no

WIEBE VAN DER HOEK and MICHAEL WOOLDRIDGE, *Department of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK*
E-mail: wiebe,mjw@csc.liv.ac.uk

Abstract

Although normative systems, or social laws, have proved to be a highly influential approach to coordination in multi-agent systems, the issue of *compliance* to such normative systems remains problematic. In all real systems, it is possible that some members of an agent population will not comply with the rules of a normative system, even if it is in their interests to do so. It is therefore important to consider the extent to which a normative system is *robust*, i.e., the extent to which it remains effective even if some agents do not comply with it. We formalise and investigate three different notions of robustness and related decision problems. We begin by considering sets of agents whose compliance is necessary and/or sufficient to guarantee the effectiveness of a normative system; we then consider quantitative approaches to robustness, where we try to identify the proportion of an agent population that must comply in order to ensure success; and finally, we consider a more general approach, where we characterise the compliance conditions required for success as a logical formula. We furthermore introduce a logic for specifying properties of norm compliance in general and norm robustness in particular.

Keywords: normative systems, temporal logic, robustness, fault tolerance, computational complexity

1 Introduction

Normative systems, or social laws, have been widely promoted as an approach to coordinating multi-agent systems [2, 3, 15, 19, 26, 27]. The basic idea is that a normative system is a set of constraints on the behaviour of agents in the system; after imposing these constraints, it is intended that some desirable overall property will hold. One of the most important issues associated with such normative systems – and one of the most ignored – is that of *compliance*. Put simply, what happens if some system participants do not comply with the regulations of the normative system? Non-compliance may be accidental (e.g., a message fails and so some participants are not informed about the regulations). Alternatively, it may be deliberate but rational (e.g., a participant chooses to ignore the norms because it does not see them as being in its own best interests), or deliberately irrational (e.g., a computer virus). Whatever the cause, it seems inevitable that, in real, large-scale systems, non-compliance will occur, and it is therefore important to consider the consequences of non-compliance. Existing research has addressed the issue of non-compliance in at least two ways.

First, one can design the normative system taking the goals and aspirations of system participants into account, so that compliance is the rational choice for participants [3]. Using the terminology of mechanism design [22, p.179], we try to make compliance *incentive*

Vol. 18 No. 1,
doi:10.1093/jigpal/jzp070

compatible. Where this approach is available, it seems highly attractive. However, given some desired objective for a normative system, it is not always possible to construct an incentive compatible normative system that achieves some outcome, and even where it is possible, it is still likely that large, open systems will fall prey to irrational behaviour.

Second, one can combine the normative system with some *penalty* mechanism, to punish non-compliance [10]. The advantage of this approach is that it can be applied to most scenarios, and that it is familiar (this is, after all, how normative systems often work in the real world). There are many disadvantages, however. For example, it may be hard to detect when non-compliance has occurred, and in large, Internet-like systems, it may be hard to impose penalties (e.g., across national borders).

For these reasons, in this paper we introduce the notion of *robustness* for normative systems. Intuitively, a normative system is robust to the extent to which it remains effective in the event of non-compliance by some agents.

The starting point is models of multi-agent systems in the form of *state-transition systems*, a very common model used in computer science, artificial intelligence and multi-agent systems. A normative system – a set of constraints on the behaviour of agents – is then modelled by labelling a subset of the transitions as ''illegal'' or ''bad'' or ''forbidden'' or ''red''. This semantic device is not novel here – it is ''frequently encountered in the deontic logic literature'' [25, p.223]. Meyer uses ''illegal'' labels on states, while Sergot and Craven [24, 25] use ''permitted'' labels on both transitions and states. As said, in this paper we label ''illegal'' transitions, but we point out that the concepts we study would make sense also in the alternative type of models where states where labelled as ''illegal'' instead of transitions. However, we find it more natural to label transitions, since we are interested in constraints on *behaviour*. Similar semantic models are used by Shoham and Tennenholtz in their seminal work on social laws for coordinating multi-agent systems [26, 27], and are also further studied in [2, 3, 15, 19]. Note also that this model of normative systems have a high abstraction level; they *only* describe ''illegal'' transitions and abstract away other aspects of normative systems such as sanctions and penalties [17], mechanisms for run-time norm change [11], the role of institutions [28], mechanisms for monitoring norm normative systems, and so on. This is not because we assume that these aspects are not present in a normative system, but we abstract away from them because the notion of ''illegal'' transitions is all that is needed for the notion of robustness studied in this paper. By not making assumptions about the details of other aspects, the results we obtain are more general. Furthermore, we point out that we study normative systems from a *semantic* perspective here, on the level of *models*, as contrasted with much of the literature on normative systems which study the *specification* of normative systems [12], most often using some kind of deontic logic [9, 21, 29] (of course, a logical specification would define a class of models).

Following an introduction to the technical framework of normative systems in Section 2, we introduce and investigate three ways of characterising robustness. First, in Section 3, we consider trying to identify coalitions whose compliance is *necessary* and/or *sufficient* to ensure that the normative system is effective. We characterise the complexity of checking these notions of robustness, and consider cases where verifying these notions of robustness is easier. In addition to verification we consider the complexity of *robust feasibility* of a normative system: given a reliable coalition, does there exist a normative system which is effective whenever that coalition complies? We then, in Section 4, consider a more *quantitative* notion of robustness, called *k-robustness*, where we try to identify the *number* of agents that could deviate and still leave the normative system effective. With these two notions of robustness in mind, in Section 5 we introduce a formal logic called *Norm Compliance* CTL, which can be used to specify properties of norm compliance in general and norm robustness in particular. A formal logic opens the door for logical tools such as model checkers to be used to, e.g., verify robustness properties of multi-agent systems. Finally, in Section 6, we consider a third, more general, approach to characterising robustness. The logic introduced in Section 5 is used to define a predicate over sets of agents, which characterises exactly those sets of agents whose compliance will ensure the success of the normative system. We conclude with a discussion, including some pointers to related and future work. An earlier version of this paper appeared at the AAMAS 2008 conference [5].

2 Formal Preliminaries

In this section, we present the formal framework for normative systems that we use throughout the remainder of the paper. This framework is essentially that of [2, 3, 19], which is in turn descended from [26]. Although our presentation is self-contained, it is terse, and readers are referred to [2, 3, 19] for further details and discussion.

Kripke Structures: We use *Kripke structures* as our basic semantic model for multi-agent systems [14]. A Kripke structure is essentially a directed graph, with the vertex set S corresponding to possible *states* of the system being modelled, and the relation $R \subseteq S \times S$ capturing the possible *transitions* of the system; $s^0 \in S$ denotes the *initial state* of the system. Intuitively, transitions are caused by *agents* in the system performing *actions*, although we do not include such actions in our semantic model (see, e.g., [19, 26] for models which include actions as first class citizens). An arc $(s, s') \in R$ corresponds to the execution of an atomic action by one of the agents in the system. Note that we are therefore here *not* modelling *synchronous* action. This assumption is not essential, but it simplifies the presentation. However, we find it convenient to include within our model the agents that cause transitions. We therefore assume a set A of agents, and we label each transition in R with the agent that causes the transition via a function $\alpha : R \rightarrow A$. Finally, we use a vocabulary $\Phi = \{p, q, \ldots\}$ of Boolean variables to express properties of individual states S: we use a function $V : S \rightarrow 2^{\Phi}$ to label each state with the Boolean variables that are true (or satisfied) in that state.

Formally, an *agent-labelled Kripke structure* (over Φ) is a 6-tuple:

$$K = \langle S, s^0, R, A, \alpha, V \rangle,$$

where: S is a finite, non-empty set of *states*; $s^0 \in S$ is the *initial state*; $R \subseteq S \times S$ is a total binary relation on S, which we refer to as the *transition relation*; $A = \{1, \ldots, n\}$ is a set of *agents*; $\alpha : R \rightarrow A$ labels each transition in R with an agent; and $V : S \rightarrow 2^{\Phi}$ labels each state with the set of propositional variables true in that state.

We hereafter refer to an agent-labelled Kripke structure simply as a *Kripke structure*. A *path* over a transition relation R is an infinite sequence of states $\pi = s_0, s_1, \ldots$ such that $\forall u \in \mathbb{N}$: $(s_u, s_{u+1}) \in R$. If $u \in \mathbb{N}$, then we denote by $\pi[u]$ the component indexed by u in π (thus $\pi[0]$ denotes the first element, $\pi[1]$ the second, and so on). A path π such that $\pi[0] = s$ is an *s-path*. Let $\Pi_R(s)$ denote the set of s-paths over R; since it will usually be clear from context, we often omit reference to R, and simply write $\Pi(s)$. We will sometimes refer to and think of an s-path as a possible computation, or system evolution, from s.

CTL: We use Computation Tree Logic (CTL), a well-known and widely used branching time temporal logic, to express the *objectives* of normative systems [14]. Given a set $\Phi = \{p, q, \ldots\}$ of atomic propositions, the syntax of CTL is defined by the following grammar, where $p \in \Phi$:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E}\bigcirc\varphi \mid \mathsf{E}(\varphi\mathcal{U}\varphi) \mid \mathsf{A}\bigcirc\varphi \mid \mathsf{A}(\varphi\mathcal{U}\varphi)$$

The semantics of CTL are given with respect to the satisfaction relation "$\models$", which holds between *pointed structures* K, s, (where K is a Kripke structure and s is a state in K), and formulae of the language. The satisfaction relation is defined as follows:

$K, s \models \top$;
$K, s \models p$ iff $p \in V(s)$ (where $p \in \Phi$);
$K, s \models \neg\varphi$ iff not $K, s \models \varphi$;
$K, s \models \varphi \vee \psi$ iff $K, s \models \varphi$ or $K, s \models \psi$;
$K, s \models \mathsf{A}\bigcirc\varphi$ iff $\forall \pi \in \Pi(s) : K, \pi[1] \models \varphi$;
$K, s \models \mathsf{E}\bigcirc\varphi$ iff $\exists \pi \in \Pi(s) : K, \pi[1] \models \varphi$;
$K, s \models \mathsf{A}(\varphi\mathcal{U}\psi)$ iff $\forall \pi \in \Pi(s), \exists u \in \mathbb{N}$, s.t. $K, \pi[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \pi[v] \models \varphi$
$K, s \models \mathsf{E}(\varphi\mathcal{U}\psi)$ iff $\exists \pi \in \Pi(s), \exists u \in \mathbb{N}$, s.t. $K, \pi[u] \models \psi$ and $\forall v, (0 \leq v < u) : K, \pi[v] \models \varphi$

The remaining classical logic connectives ("$\wedge$", "$\rightarrow$", "$\leftrightarrow$") are defined as abbreviations in terms of $\neg, \vee$ in the conventional way. The remaining CTL temporal operators are defined:

$$\begin{array}{lcl} \mathsf{A}\Diamond\varphi & \equiv & \mathsf{A}(\top\mathcal{U}\varphi) \\ \mathsf{A}\Box\varphi & \equiv & \neg\mathsf{E}\Diamond\neg\varphi \end{array} \qquad \begin{array}{lcl} \mathsf{E}\Diamond\varphi & \equiv & \mathsf{E}(\top\mathcal{U}\varphi) \\ \mathsf{E}\Box\varphi & \equiv & \neg\mathsf{A}\Diamond\neg\varphi \end{array}$$

We say φ is *satisfiable* if $K, s \models \varphi$ for some Kripke structure K and state s in K; φ is *valid* if $K, s \models \varphi$ for all Kripke structures K and states s in K. The problem of checking whether $K, s \models \varphi$ for given K, s, φ (*model checking*) can be done in deterministic polynomial time, while checking whether a given φ is satisfiable or whether φ is valid is EXPTIME-complete [14]. We write $K \models \varphi$ if $K, s^0 \models \varphi$, and $\models \varphi$ if $K \models \varphi$ for all K.

Later, we will make use of two fragments of CTL: the universal language L^u (with typical element μ), and the existential fragment L^e (typical element ε):

$$\begin{array}{l} \mu ::= \top \mid \bot \mid p \mid \neg p \mid \mu \vee \mu \mid \mu \wedge \mu \mid \mathsf{A}\bigcirc\mu \mid \mathsf{A}\Box\mu \mid \mathsf{A}(\mu\mathcal{U}\mu) \\ \varepsilon ::= \top \mid \bot \mid p \mid \neg p \mid \varepsilon \vee \varepsilon \mid \varepsilon \wedge \varepsilon \mid \mathsf{E}\bigcirc\varepsilon \mid \mathsf{E}\Box\varepsilon \mid \mathsf{E}(\varepsilon\mathcal{U}\varepsilon) \end{array}$$

The key point about these fragments is as follows. Let us say, for two Kripke structures $K_1 = \langle S, s^0, R_1, A, \alpha, V\rangle$ and $K_2 = \langle S, s^0, R_2, A, \alpha, V\rangle$ that K_1 is a subsystem of K_2 and K_2 is a supersystem of K_1, (denoted $K_1 \sqsubseteq K_2$), iff $R_1 \subseteq R_2$. Then we have (cf. [19]).

Theorem 1 ([19]) *Suppose $K_1 \sqsubseteq K_2$, and $s \in S$. Then:*

$$\begin{array}{lcl} \forall \varepsilon \in L^e : K_1, s \models \varepsilon & \Rightarrow & K_2, s \models \varepsilon; \quad \textit{and} \\ \forall \mu \in L^u : K_2, s \models \mu & \Rightarrow & K_1, s \models \mu. \end{array}$$

Normative Systems: For our purposes, a *normative system* (or "norm") is simply *a set of constraints on the behaviour of agents in a system* [2]. More precisely, a normative system defines, for every possible system transition, whether or not that transition is considered to be legal or not. Different normative systems may differ on whether or not a transition is legal. Formally, a normative system η (w.r.t. a Kripke structure $K = \langle S, s^0, R, A, \alpha, V \rangle$) is simply a subset of R, such that $R \setminus \eta$ is a total relation. The requirement that $R \setminus \eta$ is total is a *reasonableness* constraint: it prevents normative systems which lead to states with no successor. Let $N(R) = \{\eta : (\eta \subseteq R) \ \& \ (R \setminus \eta \text{ is total})\}$ be the set of normative systems over R. The intended interpretation of a normative system η is that $(s, s') \in \eta$ means transition (s, s') is forbidden in the context of η. We denote the *empty* normative system by $\eta_\emptyset$, i.e., $\eta_\emptyset = \emptyset$. Let $A(\eta) = \{\alpha(s, s') \mid (s, s') \in \eta\}$ denote the set of agents involved in η.

Implementing Normative Systems: The effect of *implementing* a normative system on a Kripke structure is to eliminate from it all transitions that are forbidden according to this normative system (see [2, 19]). If K is a Kripke structure, and η is a normative system over K, then $K \dagger \eta$ denotes the Kripke structure obtained from K by deleting transitions forbidden in η. Formally, if $K = \langle S, s^0, R, A, \alpha, V \rangle$, and $\eta \in N(R)$, then let $K \dagger \eta = K'$ be the Kripke structure $K' = \langle S', s^{0'}, R', A', \alpha', V' \rangle$ where:

- $S = S'$, $s^0 = s^{0'}$, $A = A'$, and $V = V'$;
- $R' = R \setminus \eta$; and
- α' is the restriction of α to R':

$$\alpha'(s, s') = \begin{cases} \alpha(s, s') & \text{if } (s, s') \in R' \\ \text{undefined} & \text{otherwise.} \end{cases}$$

The next most basic question we can ask in the context of normative systems is as follows. We are given a Kripke structure K, representing the state transition graph of our system, and we are given a CTL formula φ, representing the *objective* of a normative system designer (that is, the objective characterises what a designer wishes to accomplish with a normative system). The *feasibility* problem is then whether or not there exists a normative system η such that implementing η in K will achieve φ, i.e., whether $K \dagger \eta \models \varphi$. We say that η is effective for φ in K if $K \dagger \eta \models \varphi$.

Restrictions on Normative Systems: We make use of an operator on normative systems which corresponds to groups of agents "defecting" from the normative system. Formally, let $K = \langle S, s^0, R, A, \alpha, V \rangle$ be a Kripke structure, let $C \subseteq A$ be a set of agents over K, and let η be a normative system over K. Then $\eta \upharpoonright C$ denotes the normative system that is the same as η except that it only contains the arcs of η that correspond to the actions of agents in C, i.e., $\eta \upharpoonright C = \{(s, s') : (s, s') \in \eta \ \& \ \alpha(s, s') \in C\}$.

3 Necessity and Sufficiency

As we noted in the introduction, the basic intuition behind robust normative systems is that they remain effective in the presence of deviation, or non-compliance, by some members of the agent population. As we shall see, there are several different ways of formulating robustness. Our first approach is to try to characterise "lynchpin" agents – those agents whose compliance with the normative system is somehow crucial for the successful operation of the system. This seems appropriate when there are "key players" in the normative system – for example, where there is a single point of failure. In this section, we therefore consider coalitions whose compliance is *necessary and/or sufficient* to ensure that the normative system is effective.

We say that $C \subseteq A$ are *sufficient* for η in the context of K and φ if the compliance of C with η is effective, i.e., iff:

$$\forall C' \subseteq A : (C \subseteq C') \quad \Rightarrow \quad [K \dagger (\eta \restriction C') \models \varphi]. \tag{1}$$

The following example illustrates this notion of sufficiency.

Example 1 *Consider four agents who are attending a conference with an on-site computer facility. This service centre has currently one printer, two scanners and three PCs available. Agent a has tasks that require access to a printer and PC, agent b needs a printer and scanner, agent c is in need of a scanner and PC and agent d will need a scanner only. The set of agents is $A = \{a, b, c, d\}$. They are interested in using resources of type R_1, R_2, R_3, of each resource type R_j there are j instances of each: $R_1 = \{printer_1\}$, $R_2 = \{scanner_1, scanner_2\}$, $R_3 = \{pc_1, pc_2, pc_3\}$. At a given point in time, a resource can be owned by an agent. The actions available to the agents are making available a resource they currently own, or taking possession of a resource which is available. We assume that the agents never act at exactly the same time; in particular we assume that actions are turn-based – first a can perform some action, then b, and so on. A state s is a tuple*

$$s = \langle O_a, O_b, O_c, O_d, i \rangle$$

where, for each $i \in A$, O_i is the set of resources currently owned by i.

The number of agents that own a resource of type j cannot be greater than j. Let, for each resource R_j and state s, $avail(j, s)$ be the number of resources of type j that are not owned by an agent. The component $i \in A$ of s denotes whose turn it is: we write $turn(s) = i$. If $R_j \cap O_i \neq \emptyset$, we say that i owns a resource of type j and write $R_j \prec O_i$.

Our agents are not equal. In order to fulfill his task, agent a would every now and then like to use resources of type R_1 and R_3 simultaneously. We write $Useful(a) = \{R_1, R_3\}$. Similarly, $Useful(b) = \{R_1, R_2\}$, $Useful(c) = \{R_2, R_3\}$ while $Useful(d) = \{R_2\}$.

Let $s = \langle O_a, O_b, O_c, O_d, i \rangle$ and $s' = \langle O'_a, O'_b, O'_c, O'_d, i' \rangle$ be two states. Then $(s, s') \in R$ iff

1. *If $i = a$ then $i' = b$; if $i = b$ then $i' = c$; if $i = c$ then $i' = d$; and if $i = d$ then $i' = a$;*
2. *for all $k \neq i$ and all j: $R_j \prec O_k \Leftrightarrow R_j \prec O'_k$;*
3. *if $R_j \prec O'_i$ and $R_j \not\prec O_i$ then $avail(j, s) > 0$.*

Furthermore, $\alpha(s, s') = i$ when $turn(s) = i$.

Let the starting state of the system be such that it is agent a's turn, and nobody owns any resource. If we call this system K_0, then a first norm η_0 we impose on K is that no agent (i) owns two resources of the same type at the same time, (ii) takes possession of a resource that he does not need, (iii) takes possession of two new resources simultaneously, and (iv) fails to take possession of some useful resource if it is available when it is his turn:

$$\eta_0 = \left\{ (s, s') \mid \begin{array}{l} turn(s) = i, \textit{ and} \\ (\exists j : |O'_i \cap R_j| \geq 2, \textit{ or} \\ \exists j : |O'_i \cap R_j| \geq 1 \textit{ and } R_j \notin Useful(i), \textit{ or} \\ \exists x, y : x \neq y, x, y \in O'_i \textit{ and } x, y \notin O_i, \textit{ or} \\ \forall j : (R_j \in Useful(i), |O_i \cap R_j| = 0, \\ \quad avail(j, s) > 0) \Rightarrow |O'_i \cap R_j| = 0). \end{array} \right\}$$

Let $K_1 = K_0 \dagger \eta_0$. Now, in order to formulate some objectives of the system, let a_j^o denote that agent a owns a resource of type j and similarly for the other agents. Let

$$happy(i) = \bigwedge_{R_j \in Useful(i)} i_j^o$$

Thus $happy(i)$ means that i is in possession of all his useful resources, simultaneously. Our first objective is:

$$\varphi_1 = \mathsf{A}\,\Box \bigwedge_{i \in A} \mathsf{A}\Diamond happy(i).$$

The normative system that we will use for it is

$$\eta_1 = \{(s, s') \mid turn(s) = i \;\&\; O_i = Useful(i) \& \;\; O_i' \neq \emptyset\}$$

In words: if at some point an agent simultaneously owns all the resources that are useful for him, then he will make them available if it is his turn. Which coalitions are sufficient for this norm in the context of K_1 and φ_1? First of all, consider a coalition without agent a. If a does not comply with norm η_1, then he can grab the printer and hold on to it forever. Thus, agent b will not be happy, because there is only one printer. The same argument holds for a coalition without agent b. Thus, it seems that any sufficient coalition must include both agents a and b. But $\{a, b\}$ alone is not a sufficient coalition, as the following scenario illustrates: (1) a grabs a PC; (2) b grabs the printer; (3) c grabs a scanner; (4) d grabs the other scanner. Now, if c and d do not comply with η_1, it might be that they never give up their scanners, in which case b never will be happy. However, if a and b are joined by c in complying with η_1, the objective is obtained:

$$K_1 \dagger (\eta_1 \restriction \{a, b, c\}) \models \varphi_1$$

– it is easy to see that in fact $\{a, b, c\}$ is sufficient for η_1 in the context of K_1 and φ_1. But $\{a, b, c\}$ and its extension $\{a, b, c, d\}$ are not the only sufficient coalitions in this context: $\{a, b, d\}$ is also sufficient.

Now, associated with this notion is a decision problem: we are given K, η, φ, and C, and asked whether C are sufficient for η in the context of K and φ. It may appear at first sight that this is an easy decision problem: don't we just need to check that $K \dagger (\eta \restriction C) \models \varphi$? The answer is no. For suppose the objective is an *existential* property $\epsilon \in L^e$. Then the fact that $K \dagger (\eta \restriction C) \models \epsilon$ and $C \subseteq C'$ does not guarantee that $K \dagger (\eta \restriction C') \models \epsilon$. Intuitively, this is because, if more agents than C comply, then this might eliminate transitions from K, causing the existential property ϵ to be falsified.

Example 2 *We continue Example 1. To demonstrate that sufficiency for a norm in the context of a system and an objective is not monotonic in the coalition C, consider the following existential objective:*

$$\varphi_2 = \mathsf{E}\,\Box \neg happy(b)$$

That is, it is possible that b is forever unhappy (we will not discuss why the designer of the normative system might have such an objective). We have that:

$$K_1 \dagger (\eta_1 \restriction \{b\}) \models \varphi_2.$$

That is, if b complies with the norm η_1, the objective is true. This is because, for example, agent a can block b's access to the printer. However, as we saw in Example 1, $K_1 \dagger (\eta_1 \restriction \{a,b,c\}) \models \neg\varphi_2$, so $\{b\}$ is not sufficient for the objective φ_2.

We can prove that, in general, checking sufficiency is computationally hard.

Theorem 2 *Deciding C-sufficiency is co-NP-complete.*

Proof. Membership of co-NP is straightforward from the definitions of the problems. We prove hardness by reducing TAUT, the problem of showing that a formula Ψ of propositional logic is a tautology, i.e., is true under all interpretations. Let $x_1, \ldots, x_k$ be the Boolean variables of Ψ. The reduction is as follows. For each Boolean variable x_i we create an agent a_i, and in addition create one further agent, d. We create $3k+3$ states, and create the transition relation R and associated agent labelling α and valuation V as illustrated in Figure 1(a): inside states are the propositions true in that state, while arcs between states are labelled with the agent associated with the transition. Let s_0 be the initial state. We have thus defined the Kripke structure K. For the remaining components, define $C = \emptyset$, $\eta = \{(s_0, s_2), (s_2, s_3), (s_3, s_5), (s_5, s_6), \ldots, (s_{3k+2}, s_{3k+3})\}$ (i.e., all the lower arcs in the figure), and finally, define φ to be the formula obtained from Ψ by systematically replacing each Boolean variable x_i by $(\mathsf{E}\Diamond x_i)$. Now, we claim that η is C-sufficient for φ in K iff Ψ is a tautology. First, notice that since $C = \emptyset$, then for all $C' \subseteq A$, we have $C \subseteq C'$, and so the problem reduces to the following:

$$\forall C' \subseteq A : [K \dagger (\eta \restriction C') \models \varphi].$$

The correctness of the reduction is illustrated in Figure 1(b), where we show the Kripke structure obtained when only agent 1 defects from the normative system; in this case, the Kripke structure we obtain corresponds to a valuation of Ψ which makes variable x_1 true and all others false. ∎

However, the news is not all bad: for *universal* objectives, checking sufficiency is easy.

Corollary 1 *Deciding C-sufficiency for objectives $\mu \in L^u$ is polynomial time decidable.*

Proof. Simply check that $K \dagger (\eta \restriction C) \models \mu$; since $\mu \in L^u$, the fact that $K \dagger (\eta \restriction C') \models \mu$ for all $C \subseteq C' \subseteq A$ follows from Theorem 1. ∎

Next, we consider the obvious counterpart notion to sufficiency; that of *necessity*. We say that C are *necessary* for η in the context of K and φ iff C *must* comply with η in order for it to be effective, i.e., iff:

$$\forall C' \subseteq A : [K \dagger (\eta \restriction C') \models \varphi] \quad \Rightarrow \quad (C \subseteq C'). \tag{2}$$

The following example illustrates necessity.

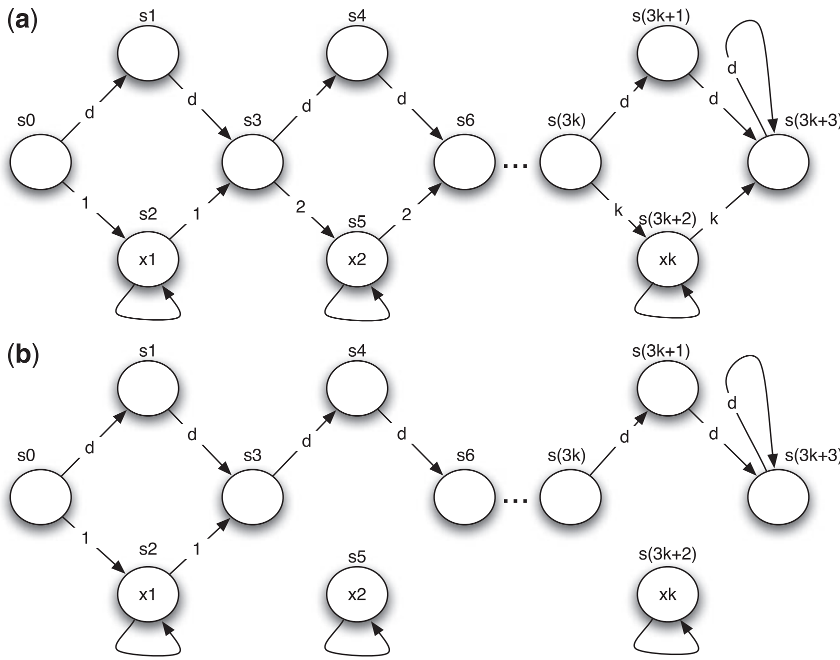

FIG. 1. Illustrating the reduction used in Theorem 2: (a) the Kripke structure produced in the reduction; (b) how the construction corresponds to a valuation: if only agent 1 defects, then the Kripke structure we obtain corresponds to a valuation in which x_1 is true (a state in which x_1 is true is reachable in the resulting structure – $\mathsf{E}\Diamond x_1$ in the objective we construct) and all other variables are false (i.e., are true in unreachable states).

Example 3 *We continue Example 1. We observed that* $\{a,b,c\}$ *and* $\{a,b,d\}$ *are sufficient for* η_1 *in the context of* K_1 *and* φ_1*. Indeed,* $\{a,b\}$ *is necessary for* η_1 *in the context of* K_1 *and* φ_1*. Both a and b must comply with the norm for the objective to be satisfied.*

Theorem 3 *Deciding* C*-necessity is co-NP-complete.*

PROOF. Membership of co-NP is obvious from the statement of the problem, so consider hardness. Note that proof of Theorem 2 does not go through for this case: since we set $C=\emptyset$ in the reduction, C are trivially necessary. However, we can use the same basic construction as Theorem 2 to prove NP-hardness of the complement problem to C-necessity, i.e., the problem of showing that

$$\exists C' \subseteq A : [K \dagger (\eta \upharpoonright C') \models \varphi] \wedge \neg (C \subseteq C').$$

We reduce SAT. Given a SAT instance Ψ, we follow the construction of Theorem 2, except that set the input coalition C to be $C=\{d\}$. It is now easy to see, using a similar argument to Theorem 2, that Ψ is satisfiable iff $\exists C' \subseteq A : [K \dagger (\eta \upharpoonright C') \models \varphi] \wedge \neg (C \subseteq C')$. ∎

The following sums up some general properties of the concepts we have discussed so far. Here, "sufficient" ("necessary") means "sufficient (necessary) for η in the context of K and φ".

Proposition 1

1. *Every coalition is sufficient if φ is a tautology, and no coalition is sufficient if φ is a contradiction.*
2. *Every coalition is necessary if φ is a contradiction, but only the empty coalition is necessary if φ is a tautology.*
3. *If a coalition is sufficient, then so is any superset of it.*
4. *If a coalition is necessary, then so is any subset of it.*
5. *There might be no sufficient coalitions.*
6. *There is always a necessary coalition: the empty coalition.*
7. *There might be two disjoint sufficient coalitions.*
8. *There might be no non-empty necessary coalitions.*
9. *If C is necessary and C' sufficient, then $C \subseteq C'$.*
10. *If there are two disjoint sufficient coalitions, then there is no non-empty necessary coalition.*

Proof.

1. If φ is a tautology, then each coalition's compliance will ensure it. And if φ is a contradiction, no matter who complies with η, φ will not be achieved.
2. If φ is a contradiction, the condition of C being necessary becomes $\forall C' \subseteq A \top$, which is true. If φ is a tautology, the condition of necessity becomes $\forall C' \subseteq A : C \subseteq C'$, which is only true for $C = \emptyset$.
3. Let $C_1 \subseteq C_2$. Now use (1) and observe that if all extensions $C' \supseteq C_1$ satisfy some property, then also all extensions $C' \supseteq C_2$ satisfy it.
4. Let $C_1 \subseteq C_2$. Now use (2) and observe that if some property of C' implies that $C_2 \subseteq C'$, then that property also implies that $C_1 \subseteq C'$.
5. Take, e.g., a system consisting of a single state with a self-loop and where p is true, and let $\varphi = \mathsf{E} \bigcirc \neg p$. η must be empty, and φ can never be true.
6. Immediate.
7. Take again the system from point 5, and let $\varphi = \mathsf{E} \bigcirc p$. Both $\{a\}$ and $\{b\}$ are sufficient, for any $a \neq b$.
8. Take the system and formula in the previous point.
9. Let C be necessary and C' sufficient. From sufficiency of C' we have that $K \dagger (\eta \restriction C') \models \varphi$, and from necessity of C it follows that $C \subseteq C'$.
10. Immediate from the above point. ∎

Note that point 9 above implies that every necessary coalition is contained in the intersection of all sufficient coalitions. Does the other direction hold, i.e., is the intersection of all sufficient coalitions necessary? In the general case the answer is "no", as the following example illustrates.

Example 4 *Take the system in Figure 2, and let $\varphi = \mathsf{E} \bigcirc \mathsf{A} \bigcirc p$. It is easy to see that:*

- $\{a\}$ *is sufficient;*
- $K \dagger (\eta \restriction \{b\}) \models \varphi$;
- *None of* $\{b\}$, $\{c\}$ *or* $\{b, c\}$ *are sufficient.*

From the first and last point it follows that $\{a\}$ is the intersection of all sufficient coalitions; from the second point it follows that $\{a\}$ is not necessary.

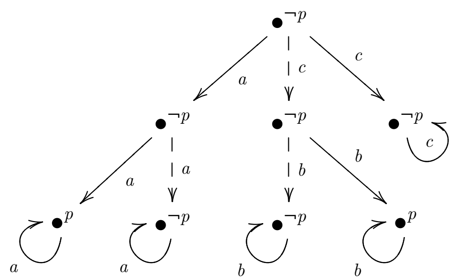

FIG. 2. A normative system. The dashed lines indicate "illegal" transitions. The uppermost state is the single initial state.

However, for universal objectives the greatest necessary coalition is exactly the intersection of the sufficient coalitions:

Lemma 1 *When the objective is a formula in L^u, the intersection of all sufficient coalitions is a necessary coalition.*

PROOF. Let $\varphi \in L^u$ and let $C = \bigcap_{C' \text{ sufficient}} C'$. Assume that $K \dagger (\eta \restriction C_2) \models \varphi$; we must show that $C \subseteq C_2$. From Theorem 1 we have $K \dagger (\eta \restriction C_3) \models \varphi$ for any C_3 such that $C_2 \subseteq C_3$. It follows that C_2 is sufficient. But then $C \subseteq C_2$. ∎

Thus, for the case of universal objectives the necessary coalitions are exactly the subsets of the intersection of the sufficient coalitions. Indeed, in Example 1 we saw that the intersection of the sufficient coalitions, consisting of agents a and b, is a necessary coalition.

3.1 Feasibility of Robust Normative Systems

So far, our technical results have focused on *verifying* robustness properties of normative systems. However, an equally important question is that of *feasibility*. As we noted earlier, feasibility basically asks whether there exists some normative system such that, if this law was imposed (and, implicitly, everybody complies), then the desired effect of the normative system would be achieved. In the context of robustness, we ask whether a normative system is *robustly* feasible. In more detail, we can think about robust feasibility as follows. Suppose we know that some subset C of the overall agent population is "reliable", in that we are confident that C can be relied upon to comply with a normative system. Then instead of asking whether there exists an *arbitrary* normative system η that is effective for our desired objective φ, we can ask whether there exists a normative system η such that C is sufficient for η in the context of φ. We call this property *C-sufficient feasibility*[1]. Formally, this question is as follows:

$$\begin{aligned}&\exists \eta \in N(R) : (K \dagger \eta \models \varphi) \wedge \\ &\quad \forall C' \subseteq A : (C \subseteq C') \Rightarrow [K \dagger (\eta \restriction C') \models \varphi].\end{aligned}$$

[1]It may at first sight seem strange that we consider this problem: why not simply look for a normative system η such that $A(\eta) = C$? Our rationale is that the *worst case* corresponds to only C complying with the normative system; it may well be that we get *better* results if more agents comply.

It turns out that, under standard complexity theoretic assumptions, checking this property is harder than the (co-NP-complete) verification problem.

Theorem 4 *Deciding C-sufficient feasibility is Σ_2^p-complete.*

Proof. We deal with the complement of the problem, which we show to be Π_2^p-complete. The complement problem is that of deciding:

$$\begin{aligned}&\forall \eta \in N(R) : (K \dagger \eta \models \varphi) \Rightarrow \\ &\quad \exists C' \subseteq A : (C \subseteq C') \wedge (K \dagger (\eta \upharpoonright C') \not\models \varphi).\end{aligned}$$

Membership is immediate from the definition of the problem. For hardness, we reduce the problem of determining whether $\mathrm{QBF}_{2,\forall}$ formulae are true [20, p.96]. An instance of $\mathrm{QBF}_{2,\forall}$ is given by a quantified Boolean formula with the following structure:

$$\forall \bar{x}_1 \, \exists \bar{x}_2 \, \chi(\bar{x}_1, \bar{x}_2) \tag{3}$$

in which $\bar{x}_1$ and $\bar{x}_2$ are disjoint sets of Boolean variables, and $\chi(\bar{x}_1, \bar{x}_2)$ is a propositional logic formula (the *matrix*) over these variables. Such a formula is true if for all assignments to Boolean variables $\bar{x}_1$, there exists an assignment to $\bar{x}_2$, such that $\chi(\bar{x}_1, \bar{x}_2)$ is true under the overall assignment. An example of a $\mathrm{QBF}_{2,\forall}$ formula is:

$$\forall x_1 \exists x_2 [(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)] \tag{4}$$

The reduction is related to that of Theorem 2, although slightly more involved. Let $\bar{x} = \{x_1, \dots, x_g\}$ be the universally quantified variables in the input formula, let $\bar{y} = \{y_1, \dots, y_h\}$ be the existentially quantified variables, and let $\chi(\bar{x}, \bar{y})$ be the matrix. We create a Kripke structure with $3(3(g+h)+3)$ states and $g+h$ agents. We create variables corresponding to $\bar{x}$ and $\bar{y}$, and in addition to these, we create a variable *end*. The overall structure is defined to be as shown in Figure 3; note that *end* is true only in the final state of the structure. We set $C = \{1, \dots, g\}$, and create the objective φ to be

$$\varphi \hat{=} (\neg \mathsf{E} \Diamond end) \vee (\neg \chi^*(\bar{x}, \bar{y}))$$

where $\chi^*(\bar{x}, \bar{y})$ is the CTL formula obtained from the propositional formula $\chi(\bar{x}, \bar{y})$ by systematically substituting $(\mathsf{E} \Diamond v)$ for each variable $v \in \bar{x} \cup \bar{y}$. Correctness follows from construction. Since the complement problem is Π_2^p-complete, C-sufficient feasibility is Σ_2^p-complete. ∎

4 k-Robustness

The notions of robustness described above are based on identifying some "critical" coalition, whose compliance is either necessary and/or sufficient for the correct functioning of the overall normative system. In this section, we explore a slightly different notion, whereby we instead *quantify* the extent to which a normative system is resistant to non-compliance. We introduce the notion of *k-robustness*, where $k \in \mathbb{N}$: intuitively, saying that a normative system is k-robust will mean that it remains effective as long as k *arbitrary* agents comply.

As with C-compliance, we can consider k-compliance from the point of view of both sufficiency and necessity. Where $k \geq 1$, we say a normative system η is *k-sufficient* (w.r.t.

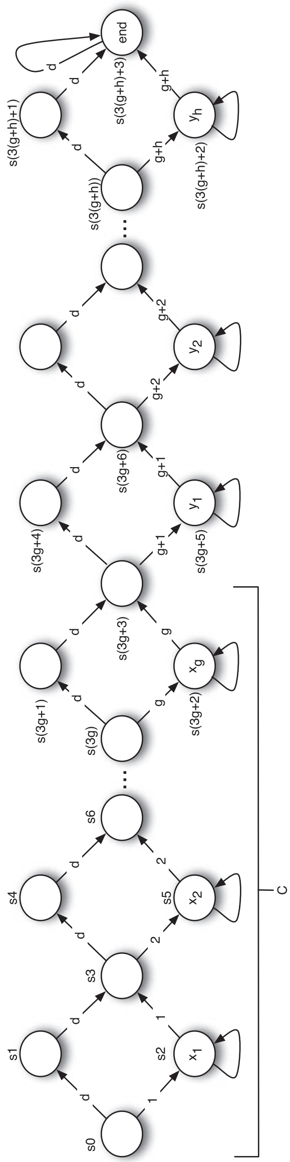

FIG. 3. Illustrating the reduction used in Theorem 4.

some K, φ) if the compliance of *any arbitrary k agents* is sufficient to ensure that the normative system is effective with respect to φ. Formally, this involves checking that:

$$\forall C \subseteq A : (|C| \geq k) \qquad \Rightarrow \qquad (K \dagger (\eta \restriction C)) \models \varphi. \tag{5}$$

As with checking C-sufficiency, checking k-sufficiency is hard.

Theorem 5 *Deciding k-sufficiency is co-*NP*-complete.*

Proof. Membership of co-NP is obvious from the problem definition; for hardness, we reduce TAUT, constructing the Kripke structure, normative system, and objective as in the proof of Theorem 2; and finally, we set $k=0$. The correctness argument is then as in Theorem 2. ■

We define the *resilience* of a normative system η (w.r.t. K, φ) as the largest number of non-compliant agents the system can tolerate. Formally, the resilience is the largest number k, $k \leq n$, such that

$$\forall C \subseteq A : (|C| \leq k) \qquad \Rightarrow \qquad (K \dagger (\eta \upharpoonright A \setminus C)) \models \varphi. \tag{6}$$

where n is the number of agents. It is easy to see that the resilience of η is the largest number k such that η is $(n-k)$-sufficient. Observe that the resilience is *undefined* iff the objective does not hold even if all agents comply to the norm ($K \dagger \eta \not\models \varphi$). Also, note that if φ is a tautology, then the resilience of a normative system is n: all agents may ignore η while φ is still true. It is immediate that computing the resilience of a normative system is co-NP-complete with respect to Turing reductions.

Example 5 *We continue Example 3. While both $\{a,b,c\}$ and $\{a,b,d\}$ are sufficient coalitions, η_1 is not 3-sufficient w.r.t. K_1, φ_1 because not every three-agent coalition is sufficient. It is 4-sufficient (the objective is satisfied if the grand coalition complies). Thus, the resilience is equal to 0.*

Now consider the situation where a has left the computer facility; b, c, d remains. Let $K_1', \eta_1', \varphi_1'$ be the corresponding variants of K_1, η_1 and φ_1. Now, each of $\{b,c\}$, $\{b,d\}$ and $\{c,d\}$ are sufficient. Thus, η_1' is 2-sufficient w.r.t. K_1', φ_1', and the resilience is 1.

We then define k-necessity in the obvious way – η is k-necessary (w.r.t. K, φ) iff:

$$\forall C \subseteq A : (K \dagger (\eta \upharpoonright C)) \models \varphi \qquad \Rightarrow \qquad (|C| \geq k). \tag{7}$$

Theorem 6 *Deciding k-necessity is co-NP-complete.*

Proof. Membership of co-NP is again obvious from the problem definition; for hardness, we reduce SAT to the complement problem, proceeding as in Theorem 3; where l is the number of Boolean variables in the SAT instance, we set $k=l+1$. Correctness of the reduction is then straightforward. ■

We say that η is *k-robust*, $k \geq 1$, if it is both k-sufficient and k-necessary. In other words, η is k-robust if it is effective exactly in the event of non-compliance of any arbitrary coalition of up to $n-k$ agents: η is k-robust iff

$$\forall C \subseteq A : (|C| \leq n-k) \qquad \Leftrightarrow \qquad (K \dagger (\eta \upharpoonright A \setminus C)) \models \varphi.$$

where n is the number of agents. From the results above, it is immediate that checking k-robustness is co-NP-complete.

Example 6 *We continue Example 5. While $\{a,b\}$ is the largest necessary coalition, η_1 is 3-necessary w.r.t. K_1, φ_1 because at least three agents must comply (in this case, either $\{a,b,c\}$ or $\{a,b,d\}$). It is not k-robust for any k, because it is 4-sufficient but not 3-sufficient, and 3-necessary but not 4-necessary.*

η_1' is both 2-sufficient and 2-necessary w.r.t. K_1', φ_1'. It is thus 2-robust. Thus, the objective will be maintained if and only if at least 2 agents comply.

Example 7 *We continue Example 6. Consider yet another variant: the agents are again all four a, b, c, d, but their needs have changed. Now each agent only needs a PC, i.e., $Useful(a) = Useful(b) = Useful(c) = Useful(d) = \{R_3\}$. Now we have that no singleton coalition is sufficient and every two-agent coalition is sufficient. The system is 2-sufficient, 2-necessary, 2-robust and its resilience is $4-2=2$.*

The following sums up some general properties of the concepts of k-robustness. Here, "k-sufficient" ("k-necessary") means "k-sufficient (k-necessary) in the context of K and φ".

Proposition 2

1. *Any system is 0-necessary.*
2. *If the system is k-sufficient, then C is sufficient for any C such that $|C| \geq k$.*
3. *If C is necessary, then the system is $|C|$-necessary.*
4. *If the system is k-sufficient for $k < n$, then no non-empty coalition is necessary.*
5. *k-robustness is unique: if the system is k-robust and k'-robust, then $k = k'$.*

Proof.

1.-3. Immediate.

4. Let $k < n$ and assume that the system is k-sufficient and that $C \neq \emptyset$ is necessary. Let C' be a coalition such that $|C'| \geq k$. By k-sufficiency, $K \dagger (\eta \upharpoonright C') \models \varphi$, and by necessity of C, $C \subseteq C'$. Since C' was arbitrary, we have that $C \subseteq \bigcap_{|C'| \geq j} C'$. Assume that $a \in C$. Let $|C_1| = k$. $a \in C_1$. Now let $b \in A \setminus C_1$ (b exists because $k < n = |A|$), and let $C_2 = C_1 \setminus \{a\} \cup \{b\}$. $|C_2| = k$, but $a \notin C_2$ which contradicts the assumption that $a \in C$. Thus, C must be empty.
5. If the system is k-robust and k'-robust for $k > k'$ and C' is a coalition of size k', then by k'-sufficiency $(K \dagger (\eta \upharpoonright C)) \models \varphi$ and by k-necessity it follows that $|C| \geq k$ which is not the case. ■

5 A Logic of Compliance

In this section we introduce a logic for that allows us to express properties of normative systems in general and robustness in particular *within the object language*. Formulae φ in the language are interpreted in the context of a Kripke structure K and a normative system η over K; $K, \eta \models \varphi$ means that the combination of the structure K and the normative system η has the property φ. The potential advantages of such a logical language are many-fold. First, it will allow us to formally reason about the logical principles of robustness – such as those mentioned in Propositions 1 and 2. Second, it would allow us *specify* normative systems, "the system should have the property φ", and to make use of standard tools to, e.g.,

- *verify* normative systems: does the normative system η have the property φ? Logically, this is the *model checking* problem, taking K, η, φ as input and checking whether or not $K, \eta \models \varphi$;
- *synthesise* normative systems: construct a normative system with property φ.

Third, it will give us a vocabulary for defining other, perhaps more sophisticated, robustness concepts than the ones we have already discussed. In this section we introduce a logic called

Norm Compliance CTL *(*NCCTL*)*, and show how it can be used to reason about compliance and robustness. In the next section we discuss how it can be used to specify general forms of robustness

We first define the language and semantics of NCCTL, before we show how it can be used to express robustness properties, investigate some logical properties of norm compliance in general and robustness in particular, and discuss relationships to other logics.

5.1 Language and Semantics

The language and semantics of *Norm Compliance* CTL *(*NCCTL*)* are defined as follows. The language extends the CTL language with an operator $\langle P\rangle$ where P is a *coalition predicate*. The intuitive meaning of a formula of the form $\langle P\rangle\varphi$ is that there exists a coalition C satisfying the predicate P, and if C cooperate in complying with the normative system, φ will be true. Coalition predicates were originally introduced in [4] as a way of quantifying over coalitions. A coalition predicate, as the name suggests, is simply a predicate over coalitions: if P is a coalition predicate, then it denotes a set of coalitions – those that satisfy P.

We first introduce the language of coalition predicates (from [4]), before we define the full language of NCCTL. Syntactically, the language of coalition predicates is built from three atomic predicates *subseteq*, *supseteq*, and *geq*, and we derive a stock of other predicate forms from these. Formally, the syntax of coalition predicates over a set of agents A is given by the following grammar:

$$P ::= \mathit{subseteq}(C) \mid \mathit{supseteq}(C) \mid \mathit{geq}(n) \mid \neg P \mid P \vee P$$

where $C \subseteq A$ is a set of agents and $n \in \mathbb{N}$ is a natural number.

The circumstances under which a coalition $C_0 \subseteq A$ satisfies a coalition predicate P are specified by the satisfaction relation "$\models_{cp}$", defined by the following rules:

$C_0 \models_{cp} \mathit{subseteq}(C)$ iff $C_0 \subseteq C$
$C_0 \models_{cp} \mathit{supseteq}(C)$ iff $C_0 \supseteq C$
$C_0 \models_{cp} \mathit{geq}(n)$ iff $|C_0| \geq n$
$C_0 \models_{cp} \neg P$ iff not $C_0 \models_{cp} P$
$C_0 \models_{cp} P_1 \vee P_2$ iff $C_0 \models_{cp} P_1$ or $C_0 \models_{cp} P_2$

We assume the conventional definitions of implication ($\rightarrow$), biconditional ($\leftrightarrow$), and conjunction ($\wedge$) in terms of $\neg$ and $\vee$. We also find it convenient to make use of the derived predicates defined in Table 1.

Note that we could have chosen a smaller base of predicates to work with, deriving the remaining predicates from these. In fact, the *eq* predicate alone would do, in the sense that any predicate P can be expressed solely in terms of *eq*:

$$\models_{cp} P \leftrightarrow \Big(\bigvee_{C \models_{cp} P} \mathit{eq}(C) \Big)$$

However, using only *eq* would not give succinct characterisations – the number of disjuncts in the expression above could be exponential in the number of agents in the system – see the discussion in [4].

$eq(C)$	$\hat{=}$	$subseteq(C) \wedge supseteq(C)$
$subset(C)$	$\hat{=}$	$subseteq(C) \wedge \neg eq(C)$
$supset(C)$	$\hat{=}$	$supseteq(C) \wedge \neg eq(C)$
$incl(i)$	$\hat{=}$	$supseteq(\{i\})$
$excl(i)$	$\hat{=}$	$\neg incl(i)$
any	$\hat{=}$	$supseteq(\emptyset)$
$nei(C)$	$\hat{=}$	$\bigvee_{i \in C} incl(i)$
$ei(C)$	$\hat{=}$	$\neg nei(C)$
$gt(n)$	$\hat{=}$	$geq(n+1)$
$lt(n)$	$\hat{=}$	$\neg geq(n)$
$leq(n)$	$\hat{=}$	$lt(n+1)$
$maj(n)$	$\hat{=}$	$geq(\lceil (n+1)/2 \rceil)$
$ceq(n)$	$\hat{=}$	$(geq(n) \wedge leq(n))$

Table 1. Derived coalition predicates.

We can now define the language of NCCTL. Formally, the formulae φ of the language are defined, over a set Φ of primitive propositions and a set A of agents, as follows:

$$\varphi ::= \quad \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \mathsf{E} \bigcirc \varphi \mid \mathsf{E}(\varphi \mathcal{U} \varphi) \mid \mathsf{A} \bigcirc \varphi \mid \mathsf{A}(\varphi \mathcal{U} \varphi) \mid \langle P \rangle \varphi$$

where $p \in \Phi$ and P is a coalition predicate over A. The usual abbreviations are used. An *objective* formula is purely propositional formula (no temporal or cooperation operators); a *non-temporal* formula is a formula without any temporal operators.

The language is interpreted in a triple K, η, s where K is a Kripke structure, η is a normative system over K and s is a state of K. The notion that $K, \eta, s \models \varphi$ is defined as follows. First, the clause for the new operator is as follows:

$$K, \eta, s \models \langle P \rangle \varphi \text{ iff } \exists C \subseteq A (C \models_{cp} P \text{ and } K \dagger (\eta \upharpoonright C), \eta, s \models \varphi) \tag{8}$$

The other clauses are then defined as in the CTL case, carrying the normative system η in the context. For example,

$$K, \eta, s \models \mathsf{A} \bigcirc \varphi \text{ iff } \forall \pi \in \Pi(s) : K, \eta, \pi[1] \models \varphi$$

We write $[P]\varphi$ as shorthand for the dual $\neg\langle P \rangle \neg\varphi$. Observe that:

$$K, \eta, s \models [P]\varphi \Leftrightarrow \text{ for all } C \subseteq A \ (C \models_{cp} P \Rightarrow K \dagger (\eta \upharpoonright C), \eta, s \models \varphi)$$

So $\langle P \rangle \varphi$ means that there is a coalition which satisfies P, and when they comply to the norm η, the objective φ is guaranteed. Similarly, $[P]\varphi$ means that compliance to η of any coalition that satisfies P will guarantee φ. Note how η on the left hand side holds the overall norm, and the $\langle P \rangle$ operator is then used to determine who complies with it. For simplicity of notation, we also write

$$\langle C \rangle \text{ for } \langle eq(C) \rangle$$

and similarly for $[C]$, and sometimes we will drop the braces in the set notation and write, e.g., $\langle a, b \rangle$ for $\langle \{a, b\} \rangle$ when $a, b \in A$ are agents.

As usual, $K,\eta \models \varphi$ means that $K,\eta,s^0 \models \varphi$. We say that a formula φ is valid, denoted $\models \varphi$, when $K,\eta \models \varphi$ for all K and η over K.

Example 8 *Continuing Example 1, we have that*

$$K_1,\eta_1 \models [\mathit{supseteq}(\{a,b,c\})]\varphi_1 \wedge \langle \mathit{eq}(\{a,b,d\})\rangle\varphi_1 \wedge \neg\langle \mathit{eq}(\{b,c,d\})\rangle\varphi_1$$

saying that compliance of all supersets of $\{a,b,c\}$ *to* η *guarantees the objective* φ_1*, whereas compliance of* $\{a,b,d\}$ *would also ensure the objective, contrary to the compliance of* $\{b,c,d\}$ *only.*

5.2 Expressing Robustness Properties

We say that a formula φ *expresses* a property of normative systems if it is satisfied by exactly the normative systems with that property, in other words if for any normative system η over any Kripke structure K it is the case that $K,\eta \models \varphi$ iff η has the property. As a simple example, recall that A is the set of all agents; then $\langle \mathit{eq}(A)\rangle\varphi$ expresses that the norm η is effective for φ (see page 8). Likewise, $\langle \mathit{eq}(A)\rangle\varphi \wedge [\mathit{subset}(A)]\neg\varphi$ would express that η is an effective, but at the same time a vulnerable norm for φ: it only takes one agent's non-compliance to make the objective φ fail. As a final general example, note that $\langle\top\rangle\varphi$ means that there is some coalition whose compliance will ensure the objective φ.

The language of NCCTL can in particular be used to express properties related to robustness. For example, the fact that coalition C are sufficient for η in the context of K and φ is expressed by the following formula.

$$[\mathit{supseteq}(C)]\varphi \tag{9}$$

This says that if any superset of C complies, φ will be true. Conversely, the fact that C are necessary for η in the context of K and φ can be expressed as follows.

$$[\neg \mathit{supseteq}(C)]\neg\varphi \tag{10}$$

This formula says that if any coalition not containing C complies, φ will not be true.

Moving on to k-robustness, the following expresses the fact that η is k-sufficient w.r.t. K and φ.

$$[\mathit{geq}(k)]\varphi \tag{11}$$

The fact that the resilience of a normative system (w.r.t. K,φ) is k can now be expressed as follows (where $n=|A|$ is the number of agents).

$$[\mathit{geq}(n-k)]\varphi \wedge \langle \mathit{ceq}(n-k-1)\rangle\neg\varphi \tag{12}$$

(Compliance of any coalition of at least $n-k$ members will ensure φ, but this is not true for $n-k-1$.)

Similarly to sufficiency and necessity for C, the following expresses the fact that η is k-necessary w.r.t. K and φ.

$$[\neg \mathit{geq}(k)]\neg\varphi \tag{13}$$

Obviously, k-robustness is expressed by the following.

$$[geq(k)]\varphi \wedge [\neg geq(k)]\neg\varphi \tag{14}$$

For further sufficiency properties, recall that there need not exist sufficient coalitions. The formula

$$\langle eq(A)\rangle\varphi \tag{15}$$

(where A is the set of all agents) expresses the fact that *there exists some sufficient coalition* (in the context of φ). To see this, observe that if there exists a sufficient coalition, the grand coalition is also sufficient (Proposition 1.3) and it follows that $\langle eq(A)\rangle\varphi$ holds. Conversely, if $\langle eq(A)\rangle\varphi$ holds, then A is trivially sufficient (it has no proper supersets). An alternative way to express the existence of a sufficient coalition (in the context of φ) is

$$\langle any\rangle[any]\varphi \tag{16}$$

which is true if there exists a coalition such that if that coalition complies, any additional compliance by any other coalition will ensure φ. Formally, observe that $K,\eta,s \models \langle any\rangle[any]\varphi$ iff there exists a coalition C such that $K \dagger (\eta \restriction C),\eta,s \models [any]\varphi$ iff, by the fact that $(\eta \restriction C) \restriction D = \eta \restriction (C \cup D)$, there exists C such that for any D, $K \dagger (\eta \restriction (C \cup D),\eta,s \models \varphi$ iff there is a C such that for any $C' \supseteq C$ it is the case that $K \dagger (\eta \restriction C'),\eta,s \models \varphi$ which means that there exists some coalition (C) which is sufficient.

Notice that (16) can be generalised to the following:

$$\langle P\rangle[any]\varphi \tag{17}$$

This expresses the fact that there exists a sufficient coalition satisfying P. For example, $\langle ceq(1)\rangle[any]\varphi$ expresses the fact that there exists a sufficient single-agent coalition.

The fact that there exists *two disjoint* sufficient coalitions (in the context of φ) can be expressed by:

$$\bigvee_{C \subseteq A} ([supseteq(C)]\varphi \wedge \langle supseteq(A \setminus C)\rangle[any]\varphi) \tag{18}$$

This formula can be read as follows: there is some coalition C which is sufficient (the first conjunct), and there exists another disjoint coalition D such that if D complies then φ will be true no matter which other agents comply (i.e., D is also sufficient).

Moving back to necessity properties, the fact that there exist non-empty necessary coalitions (recall that the empty coalition is always necessary) is expressed by

$$\neg \bigwedge_{i \in A} \langle \neg supseteq(i)\rangle\varphi \tag{19}$$

These formula schemes clearly demonstrate that the language can be used to specify the robustness properties of normative systems, and hence that logical tools and techniques such as model checking can thus be used to verify and analyse such properties of particular systems. We conclude with a slightly more detailed example.

Example 9 *Going back to examples 1–6, we have that:*

- $K_1, \eta_1 \models [supseteq(a,b,c)]\varphi_1$ *(a, b, c are sufficient)*
- $K_1, \eta_1 \models [supseteq(a,b,d)]\varphi_1$ *(a, b, d are sufficient)*
- $K_1, \eta_1 \models [eq(b)]\varphi_2 \wedge \neg[supseteq(b)]\varphi_2$ *(φ_2 is true if only b complies, but $\{b\}$ is still not sufficient)*
- $K_1, \eta_1 \models [\neg supseteq(a,b)]\neg\varphi_1$ *(a, b are necessary)*
- $K_1, \eta_1 \models \neg[geq(3)]\varphi_1 \wedge [geq(4)]\varphi_1$ *(η_1 is 4-sufficient but not 3-sufficient)*
- $K_1, \eta_1 \models [\neg geq(3)]\neg\varphi_1$ *(η_1 is 3-necessary)*

5.3 Logical Principles of Compliance and Robustness

Let us first look at some general NCCTL validities. The first validity below again shows that for expressiveness we could have taken $\langle C\rangle$ as primary and dispense with the coalition predicates, but the main motivations for including the latter are succinctness and clarity of expressions, as well as enabling expressions which are independent of the number of agents in the system.

Proposition 3 *The following are all valid.*

1. $\langle P\rangle\varphi \leftrightarrow \bigvee\{\langle C\rangle\varphi : C \models_{cp} P\}$
2. $\langle C\rangle\alpha \leftrightarrow \alpha$ — *α an objective formula*
3. $\langle C\rangle\langle D\rangle\varphi \leftrightarrow \langle C\cup D\rangle\varphi$
4. $\langle C\rangle\neg\varphi \leftrightarrow \neg\langle C\rangle\varphi$
5. $\varphi \leftrightarrow \langle\emptyset\rangle\varphi$
6. $\langle C\rangle(\varphi_1\wedge\varphi_2) \leftrightarrow (\langle C\rangle\varphi_1 \wedge \langle C\rangle\varphi_2)$
7. $\langle C\rangle(\varphi_1\vee\varphi_2) \leftrightarrow (\langle C\rangle\varphi_1 \vee \langle C\rangle\varphi_2)$
8. $\langle C\rangle\varphi \rightarrow \langle C'\rangle\varphi$ — *$C\subseteq C'$ and φ universal*
9. $\langle C'\rangle\varphi \rightarrow \langle C\rangle\varphi$ — *$C\subseteq C'$ and φ existential*
10. $\langle C\rangle(\langle C'\rangle\varphi \leftrightarrow \langle C'\setminus C\rangle\varphi)$

Note that a property like $\langle C\rangle\alpha \leftrightarrow \alpha$ (with α objective) cannot be directly generalised to arbitrary predicate modalities giving the formula scheme $\langle P\rangle\alpha \leftrightarrow \alpha$. This is because $\langle P\rangle\alpha$ implies that there is a coalition that satisfies P. We *do* have $\langle P\rangle\top \rightarrow (\langle P\rangle\alpha \leftrightarrow \alpha)$, though. We briefly mention some other properties of $\langle P\rangle$:

Proposition 4 *The following hold.*

1. $\langle P\rangle\top \leftrightarrow \mathsf{A}\,\Box\langle P\rangle\top$ *(properties of coalitions do not depend on the Kripke structure)*
2. $\langle P\rangle\langle Q\rangle\varphi \rightarrow \langle\top\rangle\varphi$ *(two coalitions that together can ensure φ, imply that there is a coalition that can ensure φ)*
3. *If* $\models_{cp} P \rightarrow Q$ *then* $\models \langle P\rangle\varphi \rightarrow \langle Q\rangle\varphi$ *(If C is a coalition satisfying P that is can ensure φ, there is, by assumption, a coalition satisfying Q which can ensure φ)*
4. $\not\models \langle P\rangle\langle P\rangle\varphi \rightarrow \langle P\rangle\varphi$ *(counterexample: suppose 4 agents are needed for compliance, and P denotes coalitions of exactly two members)*

The NCCTL validities discussed above are logical principles of norm compliance in general. Let us move on to principles more explicitly related to robustness. Using the robustness expressions we discussed in Section 5.2 to logically characterise the robustness properties from Propositions 1 and 2, we get the following.

Corollary 2

1. $\models \langle eq(Ag)\rangle\varphi \leftrightarrow \langle any\rangle[any]\varphi$, *for any* φ *(two equivalent expressions of sufficiency, see Section 5.2)*
2. $\models [subseteq(C)]\varphi \wedge \neg[supseteq(C)]\neg\varphi$, *for any* C *when* φ *is a tautology (Proposition 1.1)*
3. $\models [\neg subseteq(C)]\neg\varphi \wedge \neg[\neg supseteq(D)]\varphi$, *for any* C *and* $D \neq \emptyset$ *when* φ *is a contradiction (Proposition 1.2)*
4. $\models [supseteq(C)]\varphi \rightarrow [supseteq(D)]\varphi$, *for any* φ *and* $C \subseteq D$ *(Proposition 1.3)*
5. $\models [\neg supseteq(C)]\neg\varphi \rightarrow [\neg supseteq(D)]\neg\varphi$, *for any* φ *and* $D \subseteq C$ *(Proposition 1.4)*
6. $\not\models [eq(Ag)]\varphi$, *for some* φ *(Proposition 1.5)*
7. $\models [\neg supseteq(\emptyset)]\neg\varphi$, *for any* φ *(Proposition 1.6)*
8. $\not\models \neg\bigvee_{C\subseteq A}([supseteq(C)]\varphi \wedge \langle supseteq(A\setminus C)\rangle[any]\varphi)$, *for some* φ *(Proposition 1.7)*
9. $\not\models \bigwedge_{i\in A}\langle \neg supseteq(i)\rangle\varphi$, *for some* φ *(Proposition 1.8)*
10. *Two equivalent ways to express Proposition 1.9:*
    (a) $\models [\neg supseteq(C)]\neg\varphi \rightarrow \neg[supseteq(C')]\varphi$, *for any* C *and* C *such that* $C \not\subseteq C'$ *and any* φ
    (b) $\models [\neg supseteq(C)]\neg\varphi \rightarrow [\neg supseteq(C)]\neg[any]\varphi$, *for any* φ*. The formula says that if* C *is necessary, then any* C' *which does not contain* C *is not sufficient.*
11. $\models \bigvee_{C\subseteq A}([supseteq(C)]\varphi \wedge \langle supseteq(A\setminus C)\rangle[any]\varphi) \rightarrow \bigwedge_{i\in A}\langle\neg supseteq(i)\rangle\varphi$, *for any* φ *(Proposition 1.10)*
12. $\models [\neg geq(0)]\neg\varphi$, *for any* φ *(Proposition 2.1)*
13. $\models [geq(k)]\varphi \rightarrow [geq(k)][any]\varphi$, *for any* φ *(Proposition 2.6)*
14. $\models [\neg supseteq(C)]\neg\varphi \rightarrow [\neg geq(|C|)]\neg\varphi$, *for any* φ *(Proposition 2.7)*
15. $\models [geq(k)]\varphi \rightarrow \bigwedge_{i\in A}\langle\neg supseteq(i)\rangle\varphi$, *for any* φ *and* $k < n$ *(Proposition 2.8)*
16. $\models ([geq(k)] \wedge [\neg geq(k)]\neg\varphi) \rightarrow \neg([geq(k')]\varphi \wedge [\neg geq(k')]\neg\varphi)$, *for any* φ *and* $k \neq k'$ *(Proposition 2.9)*

5.4 Temporal Compliance

There is one feature of NCCTL that we have not addressed much yet, viz. the nesting of compliance operators in the scope of temporal operators. Since NCCTL is motivated by the desire to have a way to reason about robustness directly in the object language, this feature did not come to the fore yet. But in NCCTL one can express properties such as "in order to guarantee φ, it is sufficient that in the first k execution steps of the system, coalition C complies, while after that also D has to comply". We might formalise this in a formula $\langle C\rangle \bigcirc^k \langle D\rangle\varphi$ (where $\bigcirc^k$ indicates k repetitions of $\bigcirc$). Note that in our framework, we can only reason about 'accumulated compliance', i.e., the compliance of a coalition in a nested occurrence of $\langle\rangle$ always is assumed to be added to the compliance already induced by the earlier occurrences of $\langle\rangle$. This property of "irrevocable" compliance is further discussed in the next section.

5.5 Relationships to Other Logics

Another logic interpreted in the context of the type of normative systems we have considered here is *Normative Temporal Logic* NTL [2]. The main constructs of NTL are of the form $P_\eta \bigcirc\varphi$ and $O_\eta \bigcirc\varphi$ (tense operators $\Diamond$, $\Box$, $\mathcal{U}$ can also be used in place of $\bigcirc$), where η denotes a normative system, meaning that $\bigcirc\varphi$ is *permitted*, resp. *obligatory*, in the context of η. It is not possible to refer to agents or coalitions in the object language, which is a key feature of NCCTL.

Let us move on to logics with similar *languages*, possibly interpreted over different structures. If we want to discuss to what extent such logics are similar to NCCTL we need to look at the possibility of mappings between the semantic structures. A key notion here is validity: if the validities of the logics differ, then we can conclude that we cannot view structures of one type as structures of the other type satisfying the same formulae.

The language of NCCTL is obtained by combining the language of CTL with that of *Quantified Coalition Logic (*QCL*)* [4]. QCL extends Coalition Logic (CL) [23] with coalition predicates. The key construct of CL is of the (now familiar) form $\langle C\rangle\varphi$[2], where C is a coalition. A well known combination of coalition modalities $\langle C\rangle$[3] and branching time tense modalities is *Alternating-time Temporal Logic* ATL [8], having expressions such as $\langle C\rangle\Diamond\varphi$ – C can ensure that φ will be true sometime in the future. A variant even closer to NCCTL is ATL* which unlike (ATL) does not require that every tense modality is preceded by a coalition modality. The interpretation, however, is radically different from NCCTL. CL and ATL formulae are interpreted in Kripke structures where a strategic game form or, equivalently, an effectivity function, is associated with each state. $\langle C\rangle\varphi$ is true in a state iff each member in C can choose an action such that no matter how the other agents (outside C) act, φ will be true. The two main differences between these logics and NCCTL are that, first, in the former coalitional ability is defined by "there are some actions for C such that no matter what the other agents do, ..." (so-called α effectivity) while in the latter it is defined by the simpler "there are some actions for C such that ...", and, second, that in the latter, unlike the former, the meaning of $\langle C\rangle$ is defined by *model updates*. These two differences give different logical properties. Take the following formula:

$$\langle C\rangle(\langle C'\rangle\varphi \leftrightarrow \langle C' \setminus C\rangle\varphi) \tag{20}$$

It is valid in NCCTL, but not in CL, QCL or ATL*. Conversely, the formula

$$(\langle a\rangle \mathsf{E} \bigcirc p \wedge \langle b\rangle \mathsf{E} \bigcirc p) \rightarrow \langle a, b\rangle \mathsf{E} \bigcirc p \tag{21}$$

is valid in CL, QCL and ATL* (it is an instance of *super-additivity*), but not in NCCTL.

There is in fact a variant of ATL using model update semantics – *Irrevocable* ATL *(*IATL*)* [6] and its variant IATL*. The formula (20) *is* in fact valid in IATL* (in a sense it can be seen as an axiom of the model update semantics of coalitional ability). However, the formula

$$(\langle a\rangle\varphi \wedge \langle a\rangle\psi) \rightarrow \langle a\rangle(\varphi \wedge \psi) \tag{22}$$

where a is an agent is valid in NCCTL, but not in IATL* (nor in CL, QCL or ATL*). And the formula (21), non-valid in NCCTL, is valid also in IATL*.

In summary, NCCTL has features from ATL* (combines coalition operators with branching time temporal operators), IATL (model update semantics) and QCL (coalition predicates), but differs from all of them in that ability is not taken to be α effectivity ("no matter what

[2] In [23] $[C]$ is used where we use $\langle C\rangle$.
[3] In [8] $\langle\langle C\rangle\rangle$ is used where we use $\langle C\rangle$.

all other agents do...") and that ability has a concrete interpretation which is implicit in the Kripke structure, namely the ability to refrain from taking actions. One logic with a coalitional ability operator, with an interpretation closer to that of the coalitional ability operator of NCCTL, is *Group Announcement Logic* GAL [7]. In GAL, $\langle C\rangle\varphi$ means that C can make a *public announcement* such that φ will be true. Formally, this is defined as: for each agent $i \in C$, there exists a formula φ_i known by i (agents can have incomplete information), and in the model *updated* by the formula $\bigwedge_{i\in C} K_i\varphi_i$ (K_i is a knowledge operator for i) φ is true. This interpretation is more similar to the interpretation of NCCTL because, first, the same simple form of ability is used, "there are some actions for C" rather than "there are some actions for C such that no matter what the other agents do...", and, second, that the semantics are defined using permanent model updates. Indeed, it tempting to view the NCCTL $\langle C\rangle$ operator as a public announcement of a normative statement, where C make the (truthful) announcement "we will behave!". However, the two logics turn out to be quite different. While the two languages share the coalition modalities, they differ in that GAL has knowledge modalities (as well as announcement modalities $\langle\varphi\rangle$), while NCCTL has temporals. When restricted to only coalition modalities and atomic propositions, both logics collapse to propositional logic. Looking at schemata, we have that (22) is not valid in GAL (but valid in NCCTL). Conversely, the schema

$$\varphi \rightarrow \langle C\rangle\varphi \tag{23}$$

is valid in GAL, but not in NCCTL.

The relationship to the $nC+$ action language [25] is discussed in Section 7.

6 Logical Characterisations of General Robustness

We have thus far seen two different ways in which we might want to consider robustness: try to identify some "lynchpin" coalition (Section 3) or try to "quantify" the robustness of the normative system in terms of the number of agents whose compliance is required to make the normative system effective (Section 4). Often, however, robustness properties will not take either of these forms. For example, here is an argument about robustness that one might typically see: "the system will not overheat as long as at least one sensor works and either one of the relief valves is working or the automatic shutdown is working". Clearly, such an argument does not fit any of the types of robustness property that we have seen so far. The logic NCCTL introduced in Section 5 can be used to characterise such properties.

Let P be a predicate, and take the following formula:

$$\varphi_P \equiv [P]\varphi \wedge [\neg P]\neg\varphi \tag{24}$$

Given a Kripke structure K, normative system η, objective φ, and coalition predicate P, observe that

$$K, \eta \models \varphi_P$$

iff

$$\forall C \subseteq A: \qquad (C \models_{cp} P) \qquad \Leftrightarrow \qquad ((K \dagger (\eta \restriction C)) \models \varphi).$$

i.e., the coalitions satisfying P are exactly those whose compliance with η is effective (w.r.t. K, φ). Thus, we say that P *characterises the robustness of* η (w.r.t. K) iff the formula (24) holds in K, η.

As a simple example, consider the following simple coalition predicate.

$$supseteq(C) \tag{25}$$

We have that (25) characterises the robustness of a normative system η w.r.t. K, φ iff:

$$\forall C' \subseteq A: \qquad (C \subseteq C') \qquad \Leftrightarrow \qquad ((K \dagger (\eta \restriction C)) \models \varphi),$$

in other words, iff C are necessary and sufficient. As another simple example, the predicate $geq(k)$ characterises the robustness of η iff η is k-robust.

The decision problem of *P-characterisation* is that of checking whether for a given coalition predicate P, Kripke structure K and normative system η over K, $K, \eta \models [P]\varphi \wedge [\neg P]\neg\varphi$. Since we can use P-characterisation to express necessary and sufficient coalitions, we have the following.

Corollary 3 *Deciding P-characterisation is co-*NP*-complete.*

Notice that P-characterisation is fully expressive with respect to robustness properties, in that *any* robustness property can be characterised with a coalition predicate of the form:

$$eq(C_1) \vee eq(C_2) \vee \cdots \vee eq(C_u).$$

for some $u \in \mathbb{N}$. In the worst case, of course, we may need a coalition predicate where u may be exponential in the number of agents.

Let us consider some example coalition predicates, and what they say about robustness. Recall the informal example we used in the introduction to this section. Let S be a set of sensors, let R be the set of relief valves, and let a be the automatic shutdown system. Then the following coalition predicate characterises the robustness property expressed in that argument.

$$nei(S) \wedge (nei(R) \vee incl(a))$$

The coalition predicate *any* characterises the property that the normative system is trivial, in the sense that it is robust against any deviation (in which case it is unnecessary, since the objective will hold of the original system). The coalition predicate $\neg any$ characterises the property that the normative system will fail w.r.t. its objective irrespective of who complies with it.

7 Conclusions

We have investigated three types of robustness: necessary and/or sufficient coalitions; the number of non-compliant agents that can be tolerated; and, more generally, a logical characterisation of robustness.

Fitoussi and Tennenholz [15] formulate two criteria when choosing between different social laws. *Simplicity* tries to minimise, for each agent, the differences between states in terms of the allowed actions. The idea behind *minimality* is to reduce the number of forbidden actions that are not necessary to achieve the objective. Obviously, these two criteria typically conflict: one may sacrifice one in favour of the other. One would expect that there is a trade-off between minimality and robustness, and that minimality of η would coincide with the grand coalition A being necessary for it. This match is not perfect, however: first of all, if the latter condition holds, there still may be more transitions forbidden for A than necessary to guarantee the objective φ. Secondly, it might be that not all agents in A are constrained by η. But what we *do* have is that a minimal norm η must have $A(\eta)$ (the agents involved in it) as a necessary coalition.

Recently, French *et al.* proposed a temporal logic of robustness [16]. A brief description of the main ideas, using our formalisms, is as follows. Let η be a norm. A path π complies with η if for no $n \in \mathbb{N}$, $(\pi[n], \pi[n+1]) \in \eta$, i.e., no step in π is forbidden. Let $O\varphi$ mean that φ is obligatory: it is true in s if for all η-compliant s-paths, φ holds. $P\varphi$ (φ is permitted) is $\neg O \neg \varphi$. Given an s-path π, let

$$\begin{array}{rcl} \Delta_s^1(\pi) & = & \{\pi' \mid \pi' \text{ is } s\text{-path }, \exists j \in \mathbb{N} \forall i < j \pi(i) = \pi'(i) \ \& \\ & & \pi'[j+1]\pi'[j+2]\ldots \text{ complies with } \eta\} \end{array}$$

In words: $\pi' \in \Delta_s^1$ if it is like π up to some point j, in j it may do an illegal step, but from then on complies with the norm. French *et al.* then define an operator $\blacktriangle\varphi$ ('robustly, φ') which is true on a path π, if for all paths in $\Delta_s^1(\pi)$, and π itself, φ is true. So, $\blacktriangle\varphi$ is true in a η-compliant path, if it is true in all paths that have at most one η-forbidden transition. This is a way of bringing robustness in to the object language. However, note that in [16], there is no notion of *agency*: only the system can deviate from or comply with a norm. If φ is a universal formula, then $K, s_0 \models P\blacktriangle\varphi$ would imply (in our framework) that there is a single agent i such that $A \setminus \{i\}$ is sufficient for $\mathsf{E}\varphi$, given K and η. Although it seems a good idea for future work to incorporate such 'deontic-like' operators in the object language, even the semantics of [16] is quite different from ours: whereas [16] focuses on the number of illegal transitions, we are concerned with the number of compliant agents, or compliant coalitions.

Sergot and Craven [25] extend the action language $C+$ [18], which is used to define labelled transition systems, into $nC+$, with constructs for defining "illegal" – or, in their terminology, "red" as opposed to "green" – transitions and states. The transition models are a little more sophisticated than the ones used in the current paper, in that both transitions and states are labelled as "illegal" or "permitted" – and the interplay between the two is studied. However, $C+$ and $nC+$ are not formal logics, but rather "languages for defining specific instances of labelled transitions systems [...and ...] other languages – we refer to them as 'query languages' – can then be interpreted on these structures" [25, p.223]. NCCTL is, of course, one such language, and the interpretation of NCCTL formulae against $nC+$ specified models is an interesting opportunity for future work.

Other future work w.r.t. NCCTL include a more complete study of meta-logical properties such as axiomatisation and the complexity of key decision problems. In Section 5.4 we briefly touched upon the issue of more complex temporal notions of compliance. As further discussed above, the model update semantics for the compliance operator means that compliance can never be revoked. However, it might be useful to be able to express properties such as "if C complies for the three next steps, and then D (but not necessarily C) complies for one step, the goal will be true". This could possibly be achieved by extending NCCTL with explicit "*release*" operators for compliance, similar to release operators for strategic commitment [1, 13].

References

[1] T. Ågotnes, V. Goranko, and W. Jamroga. Strategic commitment and release in logics for multi-agent systems. In *Proceedings of the 8th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2008)*, Amsterdam, July 2008.

[2] T. Ågotnes, W. van der Hoek, J. A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.

[3] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, Honolulu, Hawaii, 2007.

[4] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Quantified coalition logic. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.

[5] T. Ågotnes, W van der Hoek, and M. Wooldridge. Robust normative systems. In *The Seventh International Conference on Agents and Multi Agent Systems (AAMAS)*, pp. 665–673, Estoril, Portugal, 2008.

[6] T. Ågotnes, V. Goranko, and W. Jamroga. Alternating-time temporal logics with irrevocable strategies. In D. Samet, editor, *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XI)*, pp. 15–24, Brussels, Belgium, June 2007. Presses Universitaires de Louvain/ACM DL.

[7] T. Ågotnes and H. van Ditmarsch. Coalitions and announcements. In Padgham, Parkes, Muller, and Parsons, editors, *Proceedings of the Seventh International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 673–680, Estoril, Portugal, May 2008. IFAMAAS/ACM DL.

[8] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002.

[9] L. Åqvist. Deontic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume II Extensions of Classical Logic*. Kluwer, 1994.

[10] R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80(4):1095–1110, 1986.

[11] G. Boella, L. Torre, and H. Verhagen. Introduction to normative multiagent systems. *Comput. Math. Organ. Theory*, 12(2-3):71–79, 2006.

[12] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. In G. Boella, L. van der Torre, and H. Verhagen, editors, *Normative Multi-agent Systems*, number 07122 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

[13] T. Brihaye, A. Da Costa, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. Technical Report LSV-08-14, ENS Cachan, 2008.

[14] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume B: Formal Models and Semantics*, pp. 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1990.

[15] D. Fitoussi and M. Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1-2):61–101, 2000.

[16] T. French, C. McCabe-Dansted, and M. Reynolds. A temporal logic of robustness. In B. Konev and F. Wolter, editors, *Frontiers of Combining Systems*, *LNCS*, Vol. 4720, pp. 193–205, 2007.

[17] J. P. Gibbs. Norms: The problem of definition and classification. *The American Journal of Sociology*, 70:586–594, 1965.

[18] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain and H. Turner. Nonmonotonic causal theories. *Artificial Intelligence*, 153:2004, 2004.

[19] W. van der Hoek, M. Roberts, and M. Wooldridge. Social laws in alternating time: Effectiveness, feasibility, and synthesis. *Synthese*, 156(1):1–19, May 2007.

[20] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, pp. 67–161. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1990.

[21] J.-J. Ch. Meyer and R. J. Wieringa. Deontic logic: A concise overview. In J.-J. Ch. Meyer and R. J. Wieringa, editors, *Deontic Logic in Computer Science — Normative System Specification*, pp. 3–16. John Wiley & Sons, 1993.

[22] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press: Cambridge, MA, 1994.

[23] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[24] M. Sergot. Modelling unreliable and untrustworthy agent behaviour. In *Monitoring, Security, and Rescue Techniques in Multiagent Systems*, pp. 161–177. Springer, 2005.

[25] M. Sergot and R. Craven. The deontic component of action language *nc*+. In L. Goble and J.-J.C. Meyer, editors, *Deontic Logic and Artificial Normative Systems*, *LNCS*, Vol. 4048, pp. 222–237, 2006.

[26] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Diego, CA, 1992.

[27] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: Off-line design. In P. E. Agre and S. J. Rosenschein, editors, *Computational Theories of Interaction and Agency*, pp. 597–618. The MIT Press: Cambridge, MA, 1996.

[28] G. Therborn. Back to norms! on the scope and dynamics of norms and normative action. *Current Sociology*, 50:863–880, 2002.

[29] G. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.