Robust Normative Systems*

Thomas Ågotnes[1], Wiebe van der Hoek[2], and Michael Wooldridge[2]

[1] Dept. of Computer Engineering, Bergen University College, Norway
[2] Dept. of Computer Science, University of Liverpool, 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.

1 Introduction

Normative systems, or social laws, have been widely promoted as an approach to coordinating multi-agent systems [11, 12, 6, 8, 1, 2]. 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 [2]. Using the terminology of mechanism design [10, p.179], we try to make

* The content of this paper is also found in a paper appearing in the proceedings of the AAMAS 2008 conference.

Dagstuhl Seminar Proceedings 09121
Normative Multi-Agent Systems
http://drops.dagstuhl.de/opus/volltexte/2009/1897

compliance *incentive 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 [4]. 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. Following an introduction to the technical framework of normative systems, we introduce and investigate three ways of characterising robustness. First, 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 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. Finally, we consider a more general, *logical* approach of characterising robustness, whereby we define a predicate over sets of agents, such that this predicate characterises exactly those sets of agents whose compliance will ensure the success of the normative system. We conclude with a brief discussion, including some pointers to related and future work.

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 based on that of [8, 1, 2], which is in turn descended from [11]. Although our presentation is complete, it is succinct, and readers are referred to [8, 1, 2] for details and discussion.

Kripke Structures: We use *Kripke structures* as our basic semantic model for multi-agent systems [5]. 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; $S^0 \subseteq S$ denotes the *initial states* 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., [11, 8] 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 the properties of individual states S: we use a function $V : S \rightarrow 2^{\Phi}$ to label each state with the Boolean variables 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*; $S^0 \subseteq S$ ($S^0 \neq \emptyset$) is the set of *initial states*; $R \subseteq S \times S$ 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 [5]. 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:

$$\mathsf{A}\Diamond\varphi \equiv \mathsf{A}(\top\,\mathcal{U}\,\varphi) \qquad \mathsf{E}\Diamond\varphi \equiv \mathsf{E}(\top\,\mathcal{U}\,\varphi)$$
$$\mathsf{A}\Box\varphi \equiv \neg\mathsf{E}\Diamond\neg\varphi \qquad \mathsf{E}\Box\varphi \equiv \neg\mathsf{A}\Diamond\neg\varphi$$

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 [5]. We write $K \models \varphi$ if $K, s_0 \models \varphi$ for all $s_0 \in S^0$, 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):

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

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. [8]).

Theorem 1 ([8]). *Suppose $K_1 \sqsubseteq K_2$, and $s \in S$. Then:*

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

Normative Systems: For our purposes, a *normative system* (or “norm”) is simply *a set of constraints on the behaviour of agents in a system* [1]. 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.

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 [8, 1]). 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 $K' = \langle S', S^{0\prime}, R', A', \alpha',\ V'\rangle$ where:

- $S = S'$, $S^0 = S^{0\prime}$, $A = A'$, and $V = V'$;
- $R' = R \setminus \eta$; and

– α' is the restriction of α to R':

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

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 $K \dagger \eta \models \varphi$.

We make use of operators on normative systems which correspond 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., $\eta \upharpoonright C = \{(s,s') : (s,s') \in \eta \;\&\; \alpha(s,s') \in C\}$. Also, $\eta \restriction C$ denotes the normative system that is the same as η except that it only contains the arcs of η that ***do not*** correspond to actions of agents in C: $\eta \restriction C = \{(s,s') : (s,s') \in \eta \;\&\; \alpha(s,s') \notin C\}$.

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:

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

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 fullfil his task, agent a would every now and then like to use resources of type R_1 and R_3 simultaneously. We write $\mathit{Useful}(a) = \{R_1, R_3\}$. Simililary, $\mathit{Useful}(b) = \{R_1, R_2\}$, $\mathit{Useful}(c) = \{R_2, R_3\}$ while $\mathit{Useful}(d) = \{R_2\}$.

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. $a' = b, b' = c, c' = d$ and $d' = a$;
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 posession 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:

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

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

$$happy(i) = \bigwedge_{R_j \in \mathit{Useful}(i)} i^o_j$$

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:

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

– 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 $K \dagger (\eta \upharpoonright C) \models \varphi$? The answer is no. For suppose the objective is an *existential* property $\eta \in L^e$. Then the fact that $K \dagger (\eta \upharpoonright C) \models \eta$ and $C \subseteq C'$ does not guarantee that $K \dagger (\eta \upharpoonright C') \models \eta$. Intuitively, this is because, if more agents than C comply, then this might eliminate transitions from K, causing the existential property η 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:

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

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, $K_1 \dagger (\eta_1 \upharpoonright \{a, b, c\}) \models \neg\varphi_2$, 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.*

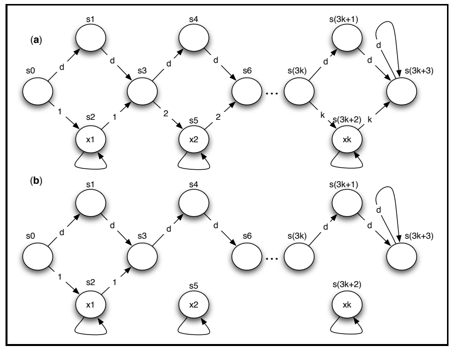

Fig. 1. Illustrating the reduction used in Theorem 2: (a) the Kripke structure produced in the reduction; (b) how the construction corresponds to a valuation: if only agent 1 defects, then the Kripke structure we obtain corresponds to a valuation in which x_1 is true (a state in which x_1 is true is 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).

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 $S^0 = \{s_0\}$ be the singleton initial state set. 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 $K \dagger (\eta \upharpoonright C) \models \mu$; since $\mu \in L^u$, the fact that $K \dagger (\eta \upharpoonright C') \models \mu$ 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:

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

The following example illustrates necessity.

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. *There might be no sufficient coalitions.*
2. *There is always a necessary coalition: the empty coalition.*
3. *There might be two disjoint sufficient coalitions.*
4. *There might be no non-empty necessary coalitions.*
5. *If C is necessary and C' sufficient, then $C \subseteq C'$.*
6. *If there are two disjoint sufficient coalitions, then there is no non-empty necessary coalition.*

Proof.

1. Trivial. 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.
2. Immediate.

3. Take again the system from the first point, and let $\varphi = \mathsf{E}\bigcirc p$. Both $\{a\}$ and $\{b\}$ are sufficient, for any $a \neq b$.
4. Take the system and formula in the previous point.
5. 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 $C \subseteq C'$.
6. Immediate from the above point.

Note that point 5 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 sufficent coalitions; from the second point it follows that $\{a\}$ is not necessary.

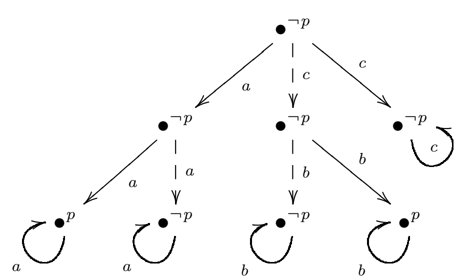

Fig. 2. A normative system. The dashed lines indicate “illegal” transitions. The uppermost state is the single inital state.

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 Examples 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 focussed 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*[3]. Formally, this question is as follows:

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

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 $\text{QBF}_{2,\forall}$ formulae are true [9, p.96]. An instance of $\text{QBF}_{2,\forall}$ is given by a quantified Boolean formula with the following structure:

$$\forall \bar{x}_1 \; \exists \bar{x}_2 \; \chi(\bar{x}_1, \bar{x}_2) \tag{1}$$

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 $\text{QBF}_{2,\forall}$ formula is:

$$\forall x_1 \exists x_2 [(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)] \tag{2}$$

The reduction is related to that of Theorem 2, although slightly more involved. Let $\bar{x} = \{x_1, \ldots, x_g\}$ be the universally quantified variables in the input formula, let $\bar{y} =$

[3] 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.

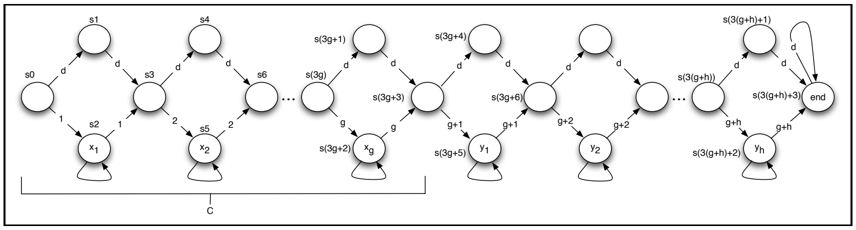

Fig. 3. Illustrating the reduction used in Theorem 4.

$\{y_1, \ldots, y_h\}$ 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 $C = \{1, \ldots, g\}$, and create the objective φ to be

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

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. 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:

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

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, $k < n$, such that

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

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 ($K \dagger \eta \not\models \varphi$). 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 wrt. 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 wrt. 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:

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

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

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

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 wrt. 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 wrt. 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., $\mathit{Useful}(a) = \mathit{Useful}(b) = \mathit{Useful}(c) = \mathit{Useful}(d) = \{R_3\}$. 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 Logical Characterisation of Robustness

We have thus far seen two different ways in which we might want to consider robustness: try to identify some "lynchpin" coalition, 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. 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. So, how are we to characterise such properties? The idea we adopt is to characterise the robustness by means of a *coalition predicate*. Coalition predicates were originally introduced in [3] 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.

$eq(C)$	$\hat{=}\ subseteq(C) \wedge supseteq(C)$
$subset(C)$	$\hat{=}\ subseteq(C) \wedge \neg eq(C)$
$supset(C)$	$\hat{=}\ supseteq(C) \wedge \neg eq(C)$
$incl(i)$	$\hat{=}\ supseteq(\{i\})$
$excl(i)$	$\hat{=}\ \neg incl(i)$
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 first introduce the language of coalition predicates (from [3]), and then show how this language can be used to characterise robustness properties. 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[4]. Formally, the syntax of coalition predicates is given by the following grammar:

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

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:

$C_0 \models_{cp} subseteq(C)$ iff $C_0 \subseteq C$
$C_0 \models_{cp} supseteq(C)$ iff $C_0 \supseteq C$
$C_0 \models_{cp} geq(n)$ 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.

Now, given a Kripke structure K, normative system η, objective φ, and coalition predicate P, we say that P *characterises the robustness of* η iff the compliance of any coalition satisfying P is sufficient to ensure that η is effective (w.r.t. K, φ). More formally, P characterises the robustness of η w.r.t. K and φ iff:

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

Now, consider the following simple coalition predicate.

$$supseteq(C) \tag{3}$$

[4] In fact, we could choose a smaller base of predicates to work with, deriving the remaining predicates from these, but the definitions would not be succinct; see the discussion in [3].

Expanding out the semantics, we have that (3) characterises the robustness of a normative system η w.r.t. K, φ iff:

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

In other words, (3) expresses that 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 a given coalition predicate P characterises robustness in the way described above. Since we can use P-characterisation to express necessary and sufficient coalitions, we have the following.

Corollary 2. *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 expresses the robustness property expressed in this argument.

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

The coalition predicate any expresses the fact 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 $\neg any$ expresses the fact that the normative system will fail w.r.t. its objective irrespective of who complies with it.

6 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 [6] 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 [7]. 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

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

In words: $\pi' \in \Delta^1_s$ 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 $\Delta^1_s(\pi)$, and π itself, φ is true. So, $\blacktriangle\varphi$ is true in a η-complient 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 [7], 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 [7] is quite different from ours: whereas [7] focusses on the number of illegal transitions, we are concerned with the number of compliant agents, or compliant coalitions.

References

1. T. Ågotnes, W. van der Hoek, J. A. Rodriguez-Aguilar, C. Sierra, and M. Wooldridge. On the logic of normative systems. In *Proc. of the Twentieth Inter. Joint Conf. on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.
2. T. Ågotnes, W. van der Hoek, and M. Wooldridge. Normative system games. In *Proc. of the Sixth Intern. Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, Honolulu, Hawaii, 2007.
3. T. Ågotnes, W. van der Hoek, and M. Wooldridge. Quantified coalition logic. In *Proc. of the Twentieth Intern. Joint Conf. on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.
4. R. Axelrod. An evolutionary approach to norms. *American Political Science Review*, 80(4):1095–1110, 1986.
5. E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume B: Formal Models and Semantics*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, 1990.
6. D. Fitoussi and M. Tennenholtz. Choosing social laws for multi-agent systems: Minimality and simplicity. *Artificial Intelligence*, 119(1-2):61–101, 2000.
7. T. French, C. McCabe-Dansted, and M. Reynolds. A temporal logic of robustness. In B. Konev and F. Wolter, editors, *Frontiers of Combining Systems*, volume 4720 of *LNCS*, pages 193–205, 2007.
8. 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.
9. D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, pages 67–161. Elsevier Science Publishers B.V.: Amsterdam, 1990.

10. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press: Cambridge, MA, 1994.
11. 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.
12. 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*, pages 597–618. The MIT Press, 1996.