Severity-Sensitive Robustness Analysis in Normative Systems

Luca Gasparini, Timothy J. Norman, Martin J. Kollingbaum, and Liang Chen

Department of Computing Science, University of Aberdeen, UK
{l.gasparini, t.j.norman, m.j.kollingbaum, l.chen}@abdn.ac.uk

Abstract. In multi-agent systems, norms specify ideal behaviour. Agents, however, are autonomous, and may fail to comply with the ideal. Contrary to Duty obligations can be used to specify reparational behaviour that mitigates the effects of a violation. In addition to specifying reparational behaviours, it is important to understand how robust a system is against possible violations. Depending on what kind of system property we want to preserve, non-compliance with different norms may be of varying severity. We propose a method for analysing robustness of normative systems, with support for Contrary to Duty obligations. We introduce violation severity as a concept orthogonal to reparational behaviour and specify it by means of a partial order between norms. We use this severity partial order, together with normative specifications, to rank the possible worlds from the most to the least compliant. In this way, we are able to use model checking to analyse robustness to a certain severity, or whether it is possible to achieve a certain goal, without violating any norm of a given severity.

1 Introduction

In multi-agent systems (MAS), a normative system specification consists of a set of constraints (norms) that specify the ideal behaviour of agents. Norms declare how agents should behave within a social context, what they should refrain from doing or what undesirable outcomes to avoid. Sub-ideal behaviour may, however, vary in severity. For example, the consequences of revealing restricted information is undesirable, but less severe than revealing secret information ("restricted" and "secret" being common information security classifications). Implicit in this example is the idea that severity is viewed as a series of levels, or, more accurately, represents a partial ordering over norm violations. This is, we believe, the best way to think of the notion of severity from the perspective of system robustness. We are interested in reasoning about how robust a system may be to some *level* of severity, given some situation, often in which some kind of norm violation is inevitable. The common alternative is to view violation severity in terms of penalties (i.e. anticipated loss of utility). This, however, leaves the way open to significant fallacies in reasoning. Consider, for example, prison terms imposed on individuals in a jurisdiction for certain crimes. Suppose that a typical term for a robbery is 6 months, and a murder 25 years. Should we infer that committing a murder (gently or not!) is equivalent to a series of 50 robberies? Penalties for norm violation are imposed *post hoc*, typically by authorities. Violations may even be excused if the alternative would have been less desirable; e.g. an under-cover policeman choosing between engaging in robberies or commiting a murder to gain trust.

Where norms capture the ideal, agents operate autonomously and, hence, their actual behaviour may violate norms. Norm violations may be accidental, due to unanticipated consequences of activities, or deliberate, for example, in order to achieve a goal that would not be possible otherwise. It is, therefore, important to account for and consider the consequences of violations. One way of addressing this issue is to define Contrary To Duty (CTD) obligations. These are structures that describe what an agent should do when a violation occurs. CTD obligations can be used to define a behaviour that mitigates the effects of a violation. In traditional deontic logic frameworks, CTD obligations often lead to inconsistencies [2]. For this reason, a number of logics have been proposed to capture and correctly reason about CTD obligations [7, 11, 12].

In addition to specifying behaviours that may mitigate the effects of a violation through CTD obligations, it is important to understand how robust a normative system is to potential future violations. For example, we may want to determine if certain desired properties are preserved even if a subset of agents in the system fail to comply with the ideal. Ågotnes et al. [1] introduced the idea of verifying robustness of normative systems. They developed a logic, Norm Compliance CTL (NCCTL), for the definition of robustness-related properties. In their model the transitions between possible worlds of a Kripke structure are divided into those allowed (green) and forbidden (red), according to a normative specification. Using NCCTL it is possible to specify properties such as "if a subset of agents comply with the normative system (i.e. do not activate any forbidden transition), it is guaranteed that a certain (un)desired property will (not) hold". In related work, Kazmierczak et al. [9] developed a model checking tool (NorMC) that enables the verification of a NCCTL property for a specified model. Model checking [3] is a formal verification technique that, given a model specification, and some properties, determines whether these properties hold. Properties can be specified using various temporal logic formalisms, such as Linear Temporal Logic (LTL) and Computation Tree Logic (CTL) (or its extension CTL*).

One open question in robustness analysis of normative systems, however, is how to reason about (non) compliance of CTD obligations. Moreover, we believe that, when analysing the robustness of normative systems, it is important to take into account the severity of violations. Our idea is based on the observation that, if our objective is to preserve certain safety properties of a system, some norms are more important than others. In fact, while a system could accept a number of violations of a certain kind, some properties might cease to hold even with only one (more severe) violation of another kind.

Our aim is to develop methods to reason about the robustness of normative systems, taking into consideration both violation severity and CTD obligations.

We apply model checking to analyse robustness, under different compliance standards. We build upon a preference-based approach to define obligations proposed by van der Torre and Tan [12] and use the preference relation to derive a ranking of the worlds according to their "ideality level"; i.e. according to how compliant these worlds are with the enforced normative system. Moreover we introduce a preference relation between obligations that specifies, for each obligation, how severe its violation would be. The preference relation between worlds is computed in such a way that worlds that violate less severe obligations or fewer obligations of the same severity are preferred. Different ranges of ranking levels are then computed according to the severity of the obligations that are violated in such worlds. This results in a partition of the world-space that is encoded in a model suitable for an off-the-shelf model checker. Further, we discuss how severity ranges are used to query the model checker about robustness-related properties and the feasibility of a given plan if we constrain ourselves to a certain severity range. Before presenting our model, however, we present an example scenario that helps illustrate the motivations behind our research.

2 Motivating Scenario

We consider as an example a coalition of three agents that includes a patrol boat, a UAV and a helicopter of the sea-guard conducting surveillance of a restricted area. In order for the coalition to achieve its mission, either the helicopter or the UAV needs to monitor the restricted area. If an unauthorized boat is discovered in the restricted area, one of the three agents must intercept the vehicle. The behaviour of the three agents is guided by the normative system specified in Example 1.

Example 1. Sea-Guard.

1. The UAV must monitor the area.
2. If the UAV does not monitor the area, the helicopter must monitor the area.
3. If an unauthorized vehicle is in the area, one of the three agents must intercept the vehicle.
4. If no agent intercepts the vehicle, one of the three agents must send a report to the head-quarters.
5. The UAV must not reveal its location.

There are, in addition to normative constraints, practical constraints that restrict possible solutions to achieve the mission goals. Neither the helicopter nor the UAV can monitor and intercept at the same time, and, by deploying the UAV to intercept the unauthorized vehicle, its position is revealed. It is easy to see that norms 2 and 4 are CTD obligations, describing behaviours that should be performed in order to mitigate the effect of violations of norms 1 and 3 respectively. As discussed before, another way of addressing the issue of non-compliance could be to develop a normative system that is robust to violations. Considering our example, we assume that the objective is to preserve the security property: “no unauthorized vehicle is to enter the restricted area, without being reported”. However it is preferred that unauthorized vehicles be intercepted. We want to be able to specify that, if our main concern is to preserve these two properties, obligation 3 or at least 4 must be always complied with, while we could accept some violations of norm 1 or 5. Moreover, in order for the coalition to be operative, we want to specify that it is more important to guarantee that there is at least one agent monitoring (either the UAV or the helicopter) rather than to avoid revealing the location of the UAV. We address this problem by defining a partial order between norms that specifies for each norm, how severe its violation is. We then want to use the normative and severity specifications to compute a ranking of the possible worlds, according to their level of compliance with the set of norms, giving more importance to violations of more severe obligations. In other words, keeping at the first level the worlds that are fully compliant, we want to give higher ranking to the possible worlds that violate more severe obligations, or more obligations of the same (or incomparable) severity. Our aim is to apply model checking to ask questions such as: is it possible to always intercept or report a boat without going through states that are above a certain severity level; i.e. without violating any norm that is as severe as a given level?

3 Formalization

Given a normative specification, our aim is to compute a preference relation between the possible worlds that reflects the level of compliance of the worlds with a set of norms. We then use this preference relation to build a ranking of possible worlds. Such a ranking can be used to partition the world-space in different severity ranges, and encode them into a model suitable for a model-checker. By doing so, we can verify different properties of a system, taking different assumptions about its level of compliance; i.e. verify how robust our system is against failures to comply with norms.

Our semantics is based on Prohairetic Deontic Logic (PDL)[12], where dyadic (conditional) obligations are represented through a preference relation between worlds. As claimed by van der Torre and Tan, this formalization allows us to correctly represent most of the scenarios involving CTD norms. Note that, like PDL, this is not a conflict-tolerant deontic logic and it requires a conflict-free normative specification. We do not address the problem of checking whether a normative specification might result in some conflicts between two or more norms, but see [13] for an example of an approach to addressing this problem. Together with a set of norms, we declare a strict partial order relation between norms that specifies the relative severity of their violation. A preference relation over possible worlds is computed using both normative and severity specifications.

We define a model $M = \langle W, B, V, OS, R, P_o \rangle$ where:

- $W = \{w_1, \dots, w_i, \dots, w_n\}$ is a set of n possible worlds.

- B is a set of boolean atoms. The set of well formed boolean formulae f is defined as $f ::= b \mid (\neg f) \mid (f \wedge f) \mid (f \vee f) \mid (f \rightarrow f)$, where $b \in B$.
- $V : W \rightarrow 2^B$ is a valuation function that assigns to each world w the set of boolean atoms that hold in w.
- $OS = \{O_1 = \mathbf{O}(\alpha_1 \mid \beta_1)), \ldots, O_m = \mathbf{O}(\alpha_m \mid \beta_m)\}$ is a normative specification, where α_i and β_i are two boolean formulae. $\mathbf{O}(\alpha_i \mid \beta_i)$ represents an obligation to achieve (or maintain) α_i that applies to the worlds where β_i holds.
- $R \subseteq W \times W$ is an accessibility relation, where $(w_i, w_j) \in R$, or alternatively $w_j \in R(w_i)$, if it is possible, from world w_i, to access w_j through a transition. A transition is an event that could lead to a change on the environment; e.g. actions performed by one or more agent, or non-deterministic events. While transitions are not used to compute the ranking of the possible worlds, we need to encode them in the model.
- $P_o \subseteq OS \times OS$ is a partial order over obligations that reflects the relative severity of their violation. Given two obligations O_i and O_j, $(O_i, O_j) \in P_o$ means that a violation of O_i is considered more severe than one of O_j. P_o is a transitive relation, thus, if we consider a graph G, where each node represents an obligation, and each edge a member of P_o, we say that violating O_a is more severe than violating O_b (alternatively $O_a \succ_o O_b$) if and only if the node representing O_b is reachable from O_a through the edges of G.

As typical in such models, prohibitions are defined in terms of obligations. Saying that a world that satisfies a is prohibited whenever b holds ($\mathbf{F}(a \mid b)$) is equivalent to saying that there is an obligation to achieve or maintain $\neg a$ whenever b holds ($\mathbf{O}(\neg a \mid b)$). Moreover, we assume that all the worlds that are not explicitly prohibited are permitted. Let α and β be two boolean atoms in B, boolean formulae satisfaction is defined as:

- $w_i \models \alpha$ iff $\alpha \in V(w_i)$.
- $w_i \models \neg\alpha$ iff $\neg(\alpha \in V(w_i))$.
- $w_i \models \alpha \wedge \beta$ iff $(w_i \models \alpha)$ and $(w_i \models \beta)$.

The other boolean operators are defined as usual.

The choice of using a partial order to specify the severity of obligations, rather than defining a fully ordered sequence of obligations, is motivated by the fact that we might have sets of obligations that are not comparable in terms of severity. We believe that our approach well represents most of the real world scenarios, where one violation of a certain norm is less desirable than several other violations, and provides the necessary flexibility to define complex structures.

In the following we define *compliance* of a world with an obligation, and *coherence* of an ordered pair of worlds with an obligation. These two concepts will be used to compute a preference relation between possible worlds, where a world w_i is preferred to w_j if it can be considered less compliant with the normative specification.

Definition 1. *A world* w_i *is* compliant *with an obligation* $O_j = \boldsymbol{O}(\alpha_j \mid \beta_j)$ *if* $w_i \models \neg\beta_j \vee \alpha_j$*: i.e. if the obligation does not apply to* w_i *or it is already satisfied. We denote this by compliant*(w_i, O_j).

Definition 2. *Given an ordered pair of worlds (w_i, w_j) where w_i, $w_j \in W$ and an obligation $O_k \in OS$, we define the following:*

$$\begin{aligned} \mathit{coherent}((w_i, w_j), O_k) &\equiv \mathit{compliant}(w_i, O_k) \wedge \neg \mathit{compliant}(w_j, O_k) \\ \mathit{incoherent}((w_i, w_j), O_k) &\equiv \mathit{compliant}(w_j, O_k) \wedge \neg \mathit{compliant}(w_i, O_k) \end{aligned}$$

We define $P_w \subseteq W \times W$ as a strict partial order that defines a preference relation between worlds. We write $(w_i, w_j) \in P_w$ or alternatively $w_i \succ_w w_j$ if w_i is preferred to w_j according to the normative system specification. P_w is computed from M according to the following rule:

$$\begin{aligned} w_i \succ_w w_j \leftrightarrow \exists\, O_k \in OS \text{ s.t. } (\mathit{coherent}((w_i, w_j), O_k) \wedge \\ (\nexists\, O_l \in OS \text{ s.t. } \neg(O_k \succ_o O_l) \wedge \mathit{incoherent}((w_i, w_j), O_l))) \end{aligned} \tag{1}$$

Informally, we say that w_i is preferable to w_j, if w_i complies with an obligation O_k that is violated by w_j, and all the obligations O_l (if any) that are violated by w_i and for which w_j is compliant, are less severe than O_k. If we assume that all obligations are incomparable in terms of severity, the statement $\neg(O_k \succ_o O_l)$ holds for any pair of obligations and P_w becomes equivalent to the preference relation between worlds defined by van der Torre and Tan [12]. Note, however, that while the preference relation of PDL semantics is reflexive, P_w is a strict one; thus, whenever $\alpha \succ_w \beta$ holds, we can say that α is preferred to β. Formally, if we denote by $\succeq_{PDL}$ the preference relation used to define the semantics for PDL, we have $w_1 \succ_w w_2 \;\equiv\; (w_1 \succeq_{PDL} w_2) \;\wedge\; \neg(w_2 \succeq_{PDL} w_1)$.

The second condition of (1) is needed to avoid the so called "strong preference problem" [12]: considering the two worlds $w_1 \models a \wedge \neg b$ and $w_2 \models \neg a \wedge b$, without such conditions, a normative system with two obligations $O_1 = \mathbf{O}(a \mid \mathit{true})$ and $O_2 = \mathbf{O}(b \mid \mathit{true})$ would result in two conflicting preference relations $w_1 \succ_w w_2$ (according to O_1) and $w_2 \succ_w w_1$ (according to O_2). Introducing our second condition, and assuming that the two obligations are incomparable according to P_w, we have no preference between these worlds. When we specify $O_a \succ_o O_b$ we want to say that a violation of O_a is more severe than a violation of O_b, thus we want to obtain a ranking where w_1 is preferred to w_2. We obtain this by restricting the second part of the equation, introducing the condition $\neg(O_k \succ_o O_l)$. Doing so, we have a preference relation (w_i, w_j) only if it is incoherent with obligations O_l that are less severe than the obligation O_k such that $\mathit{coherent}((w_i, w_j), O_k)$. Considering the previous example, since $O_a \succ_o O_b$, we have only $w_1 \succ_w w_2$.

Definition 3. *Given a set of possible worlds W and a strict partial order relation P_w on W, we define the ranking of the set as a function $\mathit{ranking}_{(P_w)} : W \to \mathbb{N}$ where:*

- $\mathit{ranking}_{(P_w)}(w_i) = 1$ *if there is no* $(w_j, w_i) \in P_w$.
- $\mathit{ranking}_{(P_w)}(w_i) = \mathit{max}[\mathit{ranking}_{(P_w)}(w_j) : (w_j, w_i) \in P_w] + 1$, *otherwise.*

Dividing the worlds by their ranking, we obtain a partition of the set W, in which states in the same subset can be considered equally compliant. We call the ranking of a possible world w_i according to P_w the *ideality level* of w_i. When verifying robustness properties, we want to reason about what properties hold when we consider only violations of a certain severity. Let $ranking_{(P_w)}(O_i)$ be the world with minimum (more compliant) ranking such that we have a violation of O_i. We can state that all the worlds with ranking lower than $ranking_{(P_w)}(O_i)$ can be considered more compliant than a world that violate O_i, while all the worlds with higher ranking violate obligations that are at least as severe as O_i. We define for each obligation O_i the *severity range of* O_i, alternatively $severity\_O_i$, as the set of worlds that have ranking lower than $ranking_{(P_w)}(O_i)$. Severity ranges can be used to verify how robust a system is to violations of a certain severity, or to verify the feasibility of a certain workflow/plan, restricting ourselves to worlds that violate only obligations that are less severe than a given one.

In the following section, we detail how we compute the strict partial order relation P_w for a model M and, given that, the $ranking_{(P_w)}$ of the possible worlds in W.

4 Normative Ranking of Possible Worlds

In this section, we introduce two algorithms. The first one uses the set of possible worlds, the set of obligations enforced and the severity relation to compute the partial order relation between worlds P_w. The second one computes a ranking of the possible worlds into *ideality levels*, from the best (most compliant) world to the worst (least compliant).

4.1 Computing P_w

Algorithm 1 computes a preference relation P_w that satisfies (Formula 1). In lines 1-6, for each enforced obligation O_1, we loop through all the possible worlds w_1 that are compliant with O_1, and for each of them we create preference relations to all non-compliant worlds w_2. From line 8 to 22 we loop again through all the obligations and remove all the preference relations (w_1, w_2) that are incoherent with the current obligation O_1. Note that, we delete a relation (w_1, w_2) only if we can find no other obligation O_2 that is more severe than the current one ($O_2 \succ_o O_1$) and such that $coherent((w_i, w_j), O_2)$ (variable `to_delete` in lines 11-16). In other words the relation is not removed if it is imposed by a more severe obligation. Recall from the definition of P_o that, in order to check whether $O_2 \succ_o O_1$, we need to check graph reachability over G. Since, in the worst case, we have to perform the reachability test n^2m^2 times, where n is the number of obligations and m the number of worlds, it is convenient to pre-compute the transitive closure of G (e.g. using the Floyd-Warshall algorithm [5]) so that we can test reachability in O(1) time. Applying Algorithm 1 to a set of worlds W, a set of obligations OS and a severity relation P_o, we obtain as output a partial order relation P_w that respects Formula 1. For all the $(w_i, w_j) \in P_w$ we can say that w_i is preferable to w_j according to the normative specification enforced: if we consider the obligations violated in the two worlds, there is at least one

Algorithm 1 Algorithm for computation of preference relation

```
for all $O_1 = \mathbf{O}(a \mid b) \in OS$ do
    for all worlds $w_1$ such that $compliant(w_1, O_1)$ do
        for all worlds $w_2$ such that $\neg compliant(w_2, O_1)$ do
            add the relation $(w1, w2)$ to $P_w$.
        end for
    end for
end for
for all $O_1 = \mathbf{O}(a \mid b) \in OS$ do
    for all worlds $w_1$ such that $\neg compliant(w_1, O_1)$ do
        for all worlds $w_2$ such that $compliant(w_2, O_1)$ do
            boolean $to\_delete$ = true
            for all $O_2 = \mathbf{O}(c \mid d) \in OS$ do
                if $(O_2 \succ_o O_1) \wedge compliant(w_1, O_2) \wedge \neg compliant(w_2, O_2)$ then
                    $to\_delete$ = false
                end if
            end for
            if $to\_delete$ then
                delete $(w_1, w_2)$ from $P_w$
            end if
        end for
    end for
end for
```

obligation violated in w_j that is more severe than all the obligations violated in w_i, or w_j violates more obligations at the highest severity level for which the number of violations is not equal between the two worlds.

4.2 Computing the Ranking

Once we have computed P_w, we can rank the worlds according to Definition 3, obtaining a ranking where the more compliant worlds are in a lower position. To do so, we extend the topological sorting algorithm developed by Kahn [8], computing the ranking while sorting the worlds in a linear extension of the partial order. The original topological sorting algorithm performs, at each iteration, the following steps: firstly, it takes all the nodes with indegree equal to 0 (i.e. no incoming edges) and inserts these nodes at the end of an ordered list called `no_incoming`; then it takes the first element of the `no_incoming` list, inserts it at the end of the list `ordered_list` (where the resulting ordered list is stored), and deletes all its outgoing edges from the `relations` list. We observe that a node w_i is inserted into the `no_incoming` list when the last node w_l such that $(w_l, w_i) \in P_w$ has been deleted. Since topological sorting deletes nodes in an order that respects the partial order (and thus the ranking), all the previously deleted nodes have ranking lower than or equal to that of w_l. It follows that $ranking_{(P_w)}(w_i)$ must be equal to $ranking_{(P_w)}(w_l) + 1$. Every time we add a node to the `no_incoming` list, we assign to the node a ranking equal to the ranking of the last node we removed from the graph incremented by 1.

We now apply the algorithms proposed in Sections 4.1 and 4.2 to the running example of Section 2, firstly assuming all the obligations to be equivalent in terms of severity, and after that, specifying the severity relations between obligations.

5 Detailed Example

The norms summarized in Example 1 can be formalized as in Table 1, where the proposition m_u stands for "The UAV is monitoring the restricted area", m_h for "the helicopter is monitoring the restricted area", *rep* stands for "the unauthorized vehicle has been reported" and r_u for "the location of the UAV has been revealed". We use a single variable *rep* instead one variable for each agent who might send a report in order to limit the possible worlds space and make our presentation more compact. The variables i_u, i_b and i_h represent the UAV, the boat and the helicopter respectively intercepting the unauthorised boat. In formalizing the normative system, we take the assumption that an unauthorized vehicle is entering the restricted area. This is the reason why norm O_3 is unconditionally active. We do this in order to simplify the example. It is possible, however, to add a variable *boat* to the model and modify the normative specification accordingly. Norm 3, for example, would become $\mathbf{O}(i_u \vee i_b \vee i_h \mid boat)$.

Table 1. Norms formalization.

Id	Norm
O_1	$\mathbf{O}(m_u \mid \top)$
O_2	$\mathbf{O}(m_h \mid \neg m_u)$
O_3	$\mathbf{O}(i_u \vee i_b \vee i_h \mid \top)$
O_4	$\mathbf{O}(rep \mid \neg(i_u \vee i_b \vee i_h))$
O_5	$\mathbf{O}(\neg r_u \mid \top)$

Considering all the possible values for the boolean variables m_u, m_h, *rep*, r_u, i_u, i_b and i_h, we compute the list of possible worlds (Table 3). In listing all the possible worlds, we do not consider all those that do not satisfy the constraints of Table 2. While constraints 3, 4 and 5 are causal constraints that allow us to capture only the subset of worlds that are meaningful in a real scenario, constraints 1, 2 and 6 should be encoded as norms; these are standard operating procedures in the scenario that guide an optimal allocation of resources. We declared them as causal constraints, again to simplify our scenario. Constraint 1 says that either the UAV or the helicopter, but not both, can monitor the area at a certain instant of time. In the same way, constraint 2 says that no more than one agent will be deployed at each instant of time. Constraint 3 states that if the UAV is deployed for interception, then its position will be revealed. Constraints 4 and 5 state that both UAV and helicopter are not able to monitor the area while intercepting targets. Constraint 6 allows us not to consider the worlds in which an unauthorized boat is both reported and intercepted.

Table 2. Sea guard scenario: possible worlds constraints

Id	**Constraint**
1	$\neg m_u \lor \neg m_h$
2	$(\neg i_b \land (\neg i_u \lor \neg i_h)) \lor (\neg i_u \land \neg i_h)$
3	$i_u \rightarrow r_u$
4	$\neg m_u \lor \neg i_u$
5	$\neg m_h \lor \neg i_h$
6	$\neg rep \lor \neg(i_u \ \lor \ i_b \ \lor \ i_h)$

Table 3. Possible worlds for the sea guard example.

Id	**World**						
	...						
w_3	$\neg i_h$	rep	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
w_9	$\neg i_h$	$\neg rep$	i_b	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
w_{13}	i_h	$\neg rep$	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
w_{16}	$\neg i_h$	$\neg rep$	$\neg i_b$	m_h	r_u	$\neg m_u$	i_u
w_{22}	$\neg i_h$	$\neg rep$	$\neg i_b$	$\neg m_h$	r_u	$\neg m_u$	$\neg i_u$
	...						

Using Algorithm 1, we compute the preference relation P_w. For example, we have $w_3 \succ_w w_{22}$ because $coherent((w_3, w_{22}), O_4)$ (same for O_5) and there is no obligation O_i such that $incoherent((w_3, w_{22}), O_i)$. We apply Algorithm 1 (Section 4.2) to the preference relation in order to compute a ranking that satisfies Definition 3. As a result, we obtain an ordered sequence of worlds, with a numeric value that represents their ranking. Part of this ranking is shown in Table 4. Worlds w_9 and w_{13}, with $ranking_{(P_w)}(w_9) = ranking_{(P_w)}(w_{13}) = 1$, are the only two possible worlds that are compliant with all the obligations, while world w_{22}, with $ranking_{(P_w)}(w_{22}) = 6$ is the only world that violates all 5 obligations. Since all the violations are considered equally severe, the ranking depends only on the number of possible violations. For example, w_3 has ranking equal to 2 because it violates only norm O_3, while w_{16} has ranking equal to 3

Table 4. Ranking without considering severity specification

R	**Id**	**World**						
1	w_9	$\neg i_h$	$\neg rep$	i_b	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
1	w_{13}	i_h	$\neg rep$	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
2	w_3	$\neg i_h$	rep	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
	...							
3	w_{16}	$\neg i_h$	$\neg rep$	$\neg i_b$	m_h	r_u	$\neg m_u$	i_u
	...							
6	w_{22}	$\neg i_h$	$\neg rep$	$\neg i_b$	$\neg m_h$	r_u	$\neg m_u$	$\neg i_u$

because the UAV is deployed for interception, and thus both norms O_1 and O_5 are violated.

As stated in Section 2, since our main objective is to preserve the properties $i_u \vee i_h \vee i_b$ and, whenever $i_u \vee i_h \vee i_b$ does not hold, to preserve *rep*, we want to be able to specify that violations of O_3 or O_4 are more severe than other violations. Moreover, since we want to specify that having someone monitoring the area is more important than not revealing the UAV location, we want to say that violations of O_2 are more severe than violations of O_1 and O_5. In other words, observing again worlds w_3 and w_{16}, we want to specify that w_3 is to be considered worse than w_{16}, even if fewer obligations are violated, because the unauthorized boat is not intercepted. To obtain a ranking that respects these two properties, we need to specify a partial order between violations and compute P_w and $ranking_{(P_w)}$ accordingly. Figure 1 shows the graph representing the severity relation in our example. The graph G is built according to the definition of P_o. Each node represents an obligation, while an arrow from an obligation O_i to another obligation O_j represents the relation $O_i \succ_o O_j$. Note that, from the transitivity property of the partial order, since O_3 and O_4 are both preferred to O_2, and O_2 is preferred to O_1 and O_5, we also have that O_3 and O_4 are preferred to O_1 and O_5.

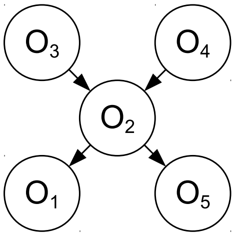

Fig. 1. Sea-guard example: severity partial order between norms

The resulting partial order P_w is just a refinement of the previous one: i.e. all the relations computed without considering obligation severity are still valid when considering any severity specifications. Compared to the preference relation obtained without considering the severity specification, we have, for example, that $w_{16} \succ_w w_3$. This is because $coherent((w_{16}, w_3), O_3)$ and there is no obligation O_i more severe than O_3 such that $incoherent((w_{16}, w_3), O_i)$ (violations of O_1 and O_5 are both considered less severe than violations of O_3).

The resulting ranking, computed according to P_w, is shown in Table 5. Even considering the severity of obligations, the most and least compliant worlds remain the same as in Table 4; i.e. the ones that comply with all the norms and the ones that violate all of them. Our purpose is to query the model checker in order to check what properties hold under different severity ranges; i.e. if

Table 5. Ranking considering severity of norms

R	Id	World						
1	w_9	$\neg i_h$	$\neg rep$	i_b	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
1	w_{13}	i_h	$\neg rep$	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
	$\dots$							
3	w_{16}	$\neg i_h$	$\neg rep$	$\neg i_b$	m_h	r_u	$\neg m_u$	i_u
	$\dots$							
6	w_3	$\neg i_h$	rep	$\neg i_b$	$\neg m_h$	$\neg r_u$	m_u	$\neg i_u$
	$\dots$							
15	w_{22}	$\neg i_h$	$\neg rep$	$\neg i_b$	$\neg m_h$	r_u	$\neg m_u$	$\neg i_u$

we restrict the set of reachable worlds to the ones that violate obligations with severity lower than a certain threshold. Recall that P_w is calculated such that worlds that violate more severe obligations have a lower ranking. Looking at our example, with $ranking_{(P_w)} \geq 6$, we have all the worlds for which O_3 is violated, and with $ranking_{(P_w)} \geq 10$ all those that violate O_4. At first sight it would seem that O_3 is preferred to O_4, but this happens because O_4 is a CTD obligation, active only in the case of violation of O_3. With $ranking_{(P_w)} \geq 4$ we have worlds that violate O_2 or more severe obligations. With $ranking_{(P_w)} \geq 2$ we have worlds that violate O_1, O_5, or more severe obligations. Our approach for severity-sensitive robustness verification is to use these values to label different severity ranges, each of them associated with an obligation, and use these labels to write queries for the model checker.

In the following section, we show how we can do so by using the PRISM [10] model checker. We encode the ranking in a PRISM model and show what kind of properties we can check using Computation Tree Logic (CTL) [3].

6 Checking Robustness

By ranking the worlds according to their ideality level we obtain a partition of the set of possible worlds. In order to verify properties about the system under different ideality levels, we can encode this partition into a model suitable for a model checker. This enables us to use model checking to ask questions such as what properties hold in each ideality level or what behaviours are feasible if we constrain ourselves to a subset of the ideality levels. In Fig. 2, we encoded the ranking of Table 5 into a model suitable for PRISM.

In Lines 14-16, we define all the variables of our model. From line 3, we write a formula for each level L_1 to L_{15}. These are boolean formulae that return `true` if and only if the model is in the given ideality level. In lines 7-8, we use the if-else construct of the PRISM modelling language to write a formula that returns an integer value corresponding, at each instant of time, to the current *ideality_level*. In lines 10-11, we define, for each obligation O_i, a boolean formula that is true if the system is currently in the severity range of O_i. Lines 13-24 describe the PRISM module, with its variables and transitions. When defining

```
dtmc

formula L1 = (!i_h & !rep &  i_b & !m_h & !r_u &  m_u & !i_u) |
  (i_h & !rep & !i_b & !m_h & !r_u &  m_u & !i_u );
...

formula ideality_level = (L1)?(1):((L2)?(2):((L3)?(3):((L4)?(4)
  :((L5)?(5):((L6)?(6):((L7)?(7): ... ))))))))))))))));

formula severity_O3 = ideality_level < 6;
...

module M1
  i_h: bool init false; i_b: bool init true; i_u: bool init false;
  rep: bool init false; m_h: bool init false; m_u: bool init true;
  r_u: bool init true;

  true -> (1.0):(r_u'= !r_u)&(i_u'=(r_u)?(false):(i_u));

  true -> (1.0):(i_h'=!i_h)&(i_u'=(!i_h)?(false):(i_u))
    &(i_b'=(!i_h)?(false):(i_b))&(m_h'=(!i_h)?(false):
    (m_h))&(rep'=(!i_h)?(false):(rep));
  ...
endmodule
```

Fig. 2. PRISM model for boat example

the possible transition, we can specify a probability p of occurrence and a guard. Each transition is fired with probability p when the guard holds. In this model we assume that all transitions have the same probability and we define the transitions so that only the possible worlds in Table 3 are reachable.

By encoding the ideality levels and severity ranges as PRISM formulae, we can use them to specify CTL properties as we do with standard variables. For example, we can ask whether it is possible to reach a world where an unauthorized boat is neither intercepted nor reported, if we restrict the world-space to the severity range defined by O_5 (Formula 2). The operator $\mathbf{E}\ \phi$ asks whether there exists a path (execution) such that the property ϕ holds. $\alpha\ \mathbf{U}\ \beta$ is a path formula that is true for a path where we can find a world t such that β holds in t and α holds in all the preceding worlds.

$$\mathbf{E}\ [\mathit{severity\_O5}\ \mathbf{U}\ (\neg(i\_u \vee i\_h \vee i\_b \vee rep) \wedge \mathit{severity\_O5})] \tag{2}$$

Considering a slightly extended model for our scenario where we can have more than one unauthorized vehicle to intercept at the same time, we could ask whether it is possible to intercept four vehicles avoiding a state where no vehicles are monitoring the area. In a similar way, if we are modelling a scenario involving collaborating workflows and some normative constraints, we could ask the model checker if it is possible to complete a specified workflow without reaching a state that is above a certain severity range (Formula 3).

$$\mathbf{E}\ [(severity\_O5)\ \mathbf{U}\ (goal\_state \wedge (severity\_O5))] \tag{3}$$

7 Discussion and Conclusions

In this paper, we have proposed a method for verifying the robustness of a normative system. This is done by partitioning the set of predictable possible worlds according to their level of compliance. We encode the partition in a model suitable for the PRISM model checker so that a world satisfies the property L_i (Lines 3-5 of Fig. 2) if it is in the ith level of the ranking. In this way, we are able to use model checking to verify what properties of interest hold at each level. We derive our ranking by computing a preference relation P_w between possible worlds that reflects the given normative specification. We then divide the worlds into different compliance levels so that if $w_i \succ_w w_j$, w_j will be in a higher level than w_i. To do so, we propose an algorithm inspired by the topological sorting algorithm [8]. Computing the preference relation is based on the semantics of Prohairetic Deontic Logic which captures most of the traditional contrary to duty benchmarks. In order to represent different levels of severity for obligation violations, we introduce a partial order over obligations. We say that an obligation $O_k \succ_o O_l$ if a violation of O_k is considered more severe than a violation of O_l. The preference relation between possible worlds is computed so that $w_i \succ_w w_j$ holds if and only if w_j violates more severe obligations or more obligations at the same level of severity.

Compared to the approach by Ågotnes et al. [1], where the set of transitions is divided into allowed and forbidden, we use a more fine-grained partitioning, dividing states according to different levels of ideality. This allows us to verify properties of worlds that are compliant only with some norms and also to represent CTD obligations. In the field of fault tolerant systems, deontic logic is also used to distinguish between correct and faulty behaviours of a system. CTD-like obligations represent behaviours that are meant to repair a fault in the system. French et al. [6] proposed a logic, called RoCTL*, for the specification and verification of robustness properties. RoCTL* enables quantification over the number of failures and the verification of properties such as "it is guaranteed that, with fewer than n violations, a certain property will hold". However there is no distinction between different kinds of violation and no means to specify different severity levels for them.

The reader might argue that the introduction of violation severity does not increase the expressiveness of our model. In fact, given a desired ranking of worlds RA, it is always possible to define a normative system that uses only CTD norms, and that would result in a ranking that is equivalent to RA. Denoting by L_i the boolean expression that identifies all the worlds at the i-th level, in the following we show how it is possible to define such a normative system.

- $\mathbf{O}(L_1 \mid true)$
- $\mathbf{O}(L_2 \mid \neg L_1)$
- $\mathbf{O}(L_3 \mid \neg L_1 \wedge \neg L_2)$
- $\ldots$
- $\mathbf{O}(L_n \mid \wedge_{i=1}^{n-1} \neg L_i)$

However, in order to do so, it would be necessary to know in advance the desired ranking of worlds and this is not always trivial. Moreover our approach enables a much more straightforward and natural formalization for the same normative system.

The PRISM model checker supports Probabilistic CTL (PCTL), an extension of CTL, that enables the expression of properties involving probabilities about events. In our model (Fig. 2) we assumed that all the transitions occur with the same probability (lines 19-27). By introducing probability values for violations of norms, we could ask the PRISM model checker to compute the likelihood that some undesired situations will happen. An interesting application of our model, together with the probabilistic features of PRISM, would be to compute the conditional probability that a certain (un)-desired property will hold, given that we restrict our model to be inside a certain severity range. A method to evaluate the risk of violating a certain norm in the context of electronic contracts has been proposed by Fagundes et al. [4].

Currently, we are only able to analyse static scenarios with a well defined configuration and there is no support for the representation of complex workflows where the goal of the coalition, or some environmental constraints could change at run-time. We can start to address this problem by verifying the robustness of the system in some representative worst case scenarios. For example, considering an extension of our scenario, it would be interesting to verify what is the maximum number of boats we can intercept at the same time and what norms we need to violate in order to intercept all of them.

Another limitation of our model is given by the fact that we define the severity preference between elements of the set of norms, OS, rather than subsets of OS. There may be situations in which the violation of two or more norms taken individually would have moderate severity but, when combined, would have more severe consequences. We plan to generalise our model in this manner and explore the consequences of using a relation $P_o \subseteq 2^{OS} \times 2^{OS}$ computationally and in modelling real-world scenarios.

Another way to improve our model would be to allow the expression of obligations that must be fulfilled before a certain deadline occurs. Such obligations would be violated only when the deadline has expired. We are also exploring the possibility of specifying different compliance assumptions for different autonomous agents in a system, in the same way that Ågotnes et al. do in NCCTL [1]. It would be interesting, for example, to be able to ask the model checker about properties of the system in the event of different subsets of agents remaining in different severity ranges.

Acknowledgments This research was sponsored by Selex ES.

References

[1] Ågotnes, T., Van der Hoek, W., Wooldridge, M.: Robust normative systems and a logic of norm compliance. Logic Journal of IGPL 18(1), 4–30 (2010)

[2] Chisholm, R.M.: Contrary-to-duty imperatives and deontic logic. Analysis 24(2), 33–36 (1963)

[3] Clarke, E.M., Grumberg, O., Peled, D.: Model checking. The MIT press (1999)

[4] Fagundes, M.S., Ossowski, S., Luck, M., Miles, S.: Using normative Markov decision processes for evaluating electronic contracts. AI Communications 25(1), 1–17 (2012)

[5] Floyd, R.W.: Algorithm 97: Shortest path. Communications of the ACM p. 345 (1962)

[6] French, T., Mc Cabe-Dansted, J.C., Reynolds, M.: A temporal logic of robustness. In: Frontiers of Combining Systems, pp. 193–205. Springer (2007)

[7] Gabbay, D.M.: Reactive Kripke models and contrary to duty obligations. Part A: Semantics. Journal of Applied Logic 11(1), 103 – 136 (2012)

[8] Kahn, A.B.: Topological sorting of large networks. Communications of the ACM 5(11), 558–562 (1962)

[9] Kazmierczak, P., Pedersen, T., Ågotnes, T.: NORMC: A Norm Compliance Temporal Logic Model Checker. In: Kersting, K., Toussaint, M. (eds.) Frontiers in Artificial Intelligence and Applications. pp. 168–179. IOS Press (2012)

[10] Kwiatkowska, M., Norman, G., Parker, D.: Advances and challenges of probabilistic model checking. In: Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing. pp. 1691–1698. IEEE (2010)

[11] Prakken, H., Sergot, M.: Contrary-to-duty obligations. Studia Logica 57(1), 91–115 (1996)

[12] van der Torre, L., Tan, Y.H.: Contrary-to-duty reasoning with preference-based dyadic obligations. Annals of Mathematics and Artificial Intelligence 27(1-4), 49–78 (1999)

[13] Vasconcelos, W.W., Kollingbaum, M.J., Norman, T.J.: Normative conflict resolution in multi-agent systems. Autonomous Agents and Multi-Agent Systems 19(2), 124–152 (2009)