Monitoring Norm Violations in Multi-Agent Systems

Nils Bulling
Department of Informatics
Clausthal University of Technology
bulling@in.tu-clausthal.de

Mehdi Dastani
Intelligent Systems Group
Utrecht University
m.m.dastani@uu.nl

Max Knobbout
Intelligent Systems Group
Utrecht University
m.knobbout@students.uu.nl

ABSTRACT

The use of norms is widely accepted as an effective approach to control and regulate the behaviour of agents in multi-agent systems. Existing work on normative multi-agent systems has mainly focussed on how norms can influence the behaviour of agents by assuming that the agents' behaviours are perfectly monitored. In this paper we focus on monitoring mechanisms, propose different types of monitors, provide a logical analysis of monitors, study the relations between monitors and norms to be monitored, and finally explore computational aspects of norm monitoring.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*

General Terms

Theory, Verification, Languages

Keywords

Normative Environment, Monitoring, Normative Systems, Logic and Reasoning

1. INTRODUCTION

In an open multi-agent system the behaviours of individual agents need to be controlled and coordinated in order to ensure the global system objectives. Norms are generally conceived as standards of behaviour that can be used to specify good behaviours and to guide the agents' activities [7]. Norms can be used by an individual agent to decide which actions to perform by reasoning and balancing between on the one hand the importance of its own objectives that may conflict with the norms, and on the other hand the consequence of norm violations [1]. However, norms can also be used by the designer of open multi-agent systems to detect violating behaviours and to impose corresponding sanctions [3]. In this paper we focus on the latter use of norms and assume that individual agents are autonomous in the sense that they decide themselves whether to obey or violate norms and that multi-agent systems have the ability to detect and enforce norms. In particular, we consider norm monitoring as a mechanism that can observe agents' activities, evaluate them with respect to a given set of norms, and signal norm violations. In the rest of this paper, we use the term normative multi-agent systems to refer to open multi-agent systems where agents' behaviours are regulated by means of norms and sanctions.

Existing literature on normative multi-agent systems often assumes that monitors are perfect in the sense that they can fully observe all agents' activities and correctly evaluate them with respect to a given set of norms. For example, consider the scenario where two cars approach a road bottleneck from opposite directions at the same time. Suppose the norm is that cars coming from the right have priority and that cars cannot pass the road bottleneck at the same time. Existing approaches assume that all cars' activities can be monitored, e.g., for each car it can be observed if the car is just before, within, and just after the road bottleneck. However, this is not a realistic assumption as monitors often can only observe a limited set of events or activities. For example, a monitor may only be able to observe that the car coming from the right is just before the road bottleneck or that a car is within the road bottleneck. In general, the notion of norm monitors is neither defined nor analysed such that it is not possible to study whether and to which extent a monitor can be effective in detecting some norm violations. The existing (perfect) monitors are inherently coupled to a set of norms in the sense that for a given set of norms the corresponding monitor can observe all agents' activities and evaluate them with respect to the given set of norms.

In this paper we aim at relaxing these assumptions by providing a logical and computational framework to specify various types of norm monitors. The framework is compositional in the sense that basic monitors can be composed to build more complex and powerful norm monitors. We study formal properties of monitors in order to determine whether a specific monitor can be effective for a specific set of norms. We also consider computational problems of norm monitors such as whether a given monitor can detect violations of a specific set of norms, whether there exists such a monitor, and if there exists a combination of some given monitors to detect the violation of a given set of norms.

2. SYSTEMS, NORMS AND MONITORS

In this section, we introduce the basic class of models that we want to study and define the concept of norms and

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

monitors. Later, in Section 3, we show how norms can be defined by means of temporal logic.

2.1 The Abstract Setting

In this paper, the class of models that we use to study and describe (multi-agent) systems are transition systems. Such systems, which we will formally define shortly, consist of states and transitions between states. The intuition is that such a model merely describes all possible transitions of states that can occur within the system, and does not necessarily state which transitions will occur during run-time. For example, in the agent-based setting, each transition might correspond to an action belonging to an agent.

Definition 1 (Transition system). *A* transition system *is a tuple $\mathfrak{T} = (Q, \rightarrow, q_0)$ where Q is a set of states, $\rightarrow \subseteq Q \times Q$ a binary serial relation over Q, i.e. for all $q \in Q$ there is a state $q' \in Q$ such that $q \rightarrow q'$, and $q_0 \in Q$ a distinguished initial state.*

Since it is often the case that we want to prove or validate certain properties of a system, we add a valuation function that for each state assigns a set of (observable) propositions which are deemed true at this particular state. We call a transition system together with a valuation function an interpreted transition system.

Definition 2 (Interpreted transition system). *Given a set of atomic propositions Props, an* interpreted transition system *is a tuple $\mathfrak{I} = (Q, \rightarrow, q_0, v)$ where $(Q, \rightarrow, q_0)$ is a transition system and $v : Q \rightarrow \mathcal{P}(Props)$ a valuation function associating propositions with states.*

In this paper we will omit the term interpreted if clear from context or not relevant. A transition system gives rise to a set of possible executions, or runs for short.

Definition 3 (Run, $\mathcal{R}$). *Given a transition system $\mathfrak{T} = (Q, \rightarrow, q_0)$ a* run *of the system is defined as an infinite sequence $q_0 q_1 \ldots \in Q^\omega$ such that q_0 is the starting state and $\forall n \in \mathbb{N}_0 : (q_n, q_{n+1}) \in \rightarrow$. For a given run r, we define $r[n]$ $(n \geq 0)$ as the n-th state q_n occurring on the run and $r[n, \infty]$ $(n \geq 0)$ as the postfix of the run starting from position n. The set of all possible runs over $\mathfrak{T}$ is denoted by $\mathcal{R}_{\mathfrak{T}}$. We define runs analogously for interpreted transition systems. In this case, we use the notation $\mathcal{R}_{\mathfrak{I}}$ or simply $\mathcal{R}$ if $\mathfrak{I}$ is clear from context.*

We consider some of the behaviours as desired, others as undesired; hence, a *norm* is a set of desired behaviours which is a subset of all possible behaviours. In the remainder of this section we assume that we are given a transition system that models the set of possible runs $\mathcal{R}$ that can occur.

Definition 4 (Norm, compliance, violation). *A* norm *$\mathcal{N}$ is a subset of system runs, i.e. $\mathcal{N} \subseteq \mathcal{R}$. A run r* complies with a norm *$\mathcal{N}$, for short r* is $\mathcal{N}$-complient, *if and only if $r \in \mathcal{N}$. Otherwise, r* violates *$\mathcal{N}$, for short r* is an $\mathcal{N}$-violation.

It is important to note that we are not interested in how such a norm is acquired (which can be through a deontic specification, emergent behaviour of the system or through a conflict resolution of multiple possible norms), we merely use the fact that once we have a mechanism to decide which trace is "good" and "bad", we can acquire a set of desired traces. A *monitor* observes the behaviours of the system. In this paper we adopt the view that a monitor for each possible run r constructs a candidate set of runs that represents all the possibilities that might have actually occurred. In the ideal case, if r takes place the monitor should observe r and nothing else. However, due to several reasons, e.g. cost limitations, noise or environmental restrictions, it is not always possible to perfectly monitor the behaviours of the system. In the general case, we model a monitor as a function $m : \mathcal{R} \rightarrow \mathcal{P}(\mathcal{R})$. Intuitively, given a run $r \in \mathcal{R}$ and a monitor m, the set $m(r)$ contains all the runs that the monitor considers possible as being the actual run r.

Definition 5 (Monitor m, R_m). *A* monitor *m over system behaviours $\mathcal{R}$ is a function $m : \mathcal{R} \rightarrow \mathcal{P}(\mathcal{R})$. We define $m(\mathcal{R}') = \bigcup_{r' \in \mathcal{R}'} m(r')$ for a set $\mathcal{R}' \subseteq \mathcal{R}$ of runs. Moreover, we define R_m as the corresponding relation over $\mathcal{R} \times \mathcal{R}$ derived from m, i.e. $rR_m r'$ iff $r' \in m(r)$.*

We can associate properties with monitors. Consider for example the extreme case in which a monitor m given the set $\mathcal{R}$ is defined as $m(r) = \mathcal{R}$ for all $r \in \mathcal{R}$. In this particular case, the monitor cannot distinguish anything; for each run of the system it is possible that any arbitrary run might have actually occurred. The other extreme case which we have already mentioned earlier is the case in which it holds that $m(r) = \{r\}$ for all $r \in \mathcal{R}$. Here we might say that the monitor can distinguish everything; it always observes the actual run. We introduce the following properties of a monitor.

Definition 6 (Properties of a monitor). *Let m be a monitor over system behaviours $\mathcal{R}$. We say that m is*

- broken *iff there exists a run $r \in \mathcal{R}$ such that $m(r) = \emptyset$.*
- correct *iff for all runs $r \in \mathcal{R}$ we have $r \in m(r)$.*
- consistent *iff for all runs $r_1, r_2, r_3 \in \mathcal{R}$ we have if $(r_1 \in m(r_3)$ and $r_2 \in m(r_3))$ then $r_1 \in m(r_2)$ and $r_2 \in m(r_1)$.*
- ideal *iff for all runs $r \in \mathcal{R}$ we have $m(r) = \{r\}$.*

The property of correctness states that a monitor should never exclude the true run from the set of possibilities. For example, an incorrect monitor may observe something that never actually took place (e.g. something like a noisy camera). The property of consistency states that if a monitor cannot differentiate between two runs, then the monitor should neither be able to differentiate these two alternatives from each other. When a monitor is broken, it can be seen as a monitor that makes impossible observations; none of the runs from $\mathcal{R}$ are deemed similar to the run that was observed.

Proposition 1. *Let m be a monitor over system behaviours $\mathcal{R}$. A monitor m is correct iff R_m is reflexive, broken iff R_m is not serial, consistent iff R_m is Euclidian, and ideal iff R_m is the identity relation.*

This proposition holds since the definitions of broken, correct, consistent and ideal directly coincide with the aforementioned properties of binary relations. However, this allows us to use results found in the study of binary relations to properties of monitors. For example, we can see that if a monitor m is both correct and consistent, then R_m is an equivalence relation. Moreover, if a monitor is correct, then it is not broken and if a monitor is ideal, then it is correct. Certainly more relations can be shown, but that is beyond the scope of this paper. We end this section with an example.

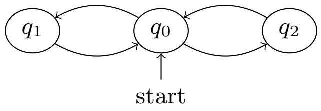

Figure 1: A simple transition system.

Example 1. *Consider the transition system shown in Figure 1, with states $Q = \{q_0, q_1, q_2\}$. Let $\mathcal{N} = \{(q_0\, q_1)^\omega\}$ consist of a single run which only visits q_0 and q_1 in succession. We define the following monitor: $m(r) = \{r' \in \mathcal{R} \mid r[1] = r'[1]\}$. This monitor observes only the first step and partitions the runs into two distinct classes: one class consists of all the runs that initially visit state q_1 and the other one of all the runs that initially visit q_2. This monitor is correct and consistent, but it is not ideal. If r initially visits q_2, then all the runs in the set $m(r)$ are $\mathcal{N}$-violations. However, if r initially visits q_1, exactly 1 run in the set $m(r)$ is not an $\mathcal{N}$-violation while the remaining runs are.*

2.2 Logic-Based Setting

In the following we introduce the *linear-time temporal logic* LTL [6] which can be used to characterize norms. Formulae of LTL are defined by the following grammar where $p \in \mathit{Props}$ denotes an atomic proposition:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\mathcal{U}\varphi \mid \bigcirc\varphi$$

LTL-formulae are interpreted over a run r of a transition system $\mathfrak{I}$. $\bigcirc\varphi$ means that φ holds in the next state of the run and $\varphi\mathcal{U}\psi$ that φ holds along the run until ψ holds. We define $\Diamond\varphi \equiv \top\mathcal{U}\varphi$ (eventually φ) and $\Box\varphi \equiv \neg\Diamond\neg\varphi$ (always φ) as macros, as well as the standard Boolean connectives. We recall the semantics of LTL-formulae:

$\mathfrak{I}, r \models_{LTL} p$ iff $p \in v(r[0])$
$\mathfrak{I}, r \models_{LTL} \neg\varphi$ iff $\mathfrak{I}, r \not\models_{LTL} \varphi$
$\mathfrak{I}, r \models_{LTL} \varphi \vee \psi$ iff $\mathfrak{I}, r \models_{LTL} \varphi$ or $\mathfrak{I}, r \models_{LTL} \psi$
$\mathfrak{I}, r \models_{LTL} \bigcirc\varphi$ iff $\mathfrak{I}, r[1, \infty] \models_{LTL} \varphi$
$\mathfrak{I}, r \models_{LTL} \varphi\mathcal{U}\psi$ iff $i \geq 0$ such that $\mathfrak{I}, r[i, \infty] \models_{LTL} \psi$ and for all $0 \leq k < i$, $\mathfrak{I}, r[k, \infty] \models_{LTL} \varphi$

We also lift the semantic satisfaction relation to sets of runs $R' \subseteq \mathcal{R}$ as follows:

$$\mathfrak{I}, R' \models \varphi \text{ iff } \forall r \in R' : \mathfrak{I}, r \models_{LTL} \varphi.$$

In the following we simply write $\models$ for $\models_{LTL}$ whenever clear from context. A norm $\mathcal{N}$ can be defined by an LTL-formula χ such that $\mathcal{N}$ contains all the runs and only the runs satisfying χ. Note however that in the general case, not all norms can be described in this way.

Definition 7 (χ-norm). *Let χ be an LTL-formula and $\mathfrak{I}$ be an interpreted transition system. The χ-norm in $\mathfrak{I}$ is defined as the set $\mathcal{N}_{\mathfrak{I}}(\chi) = \{r \in \mathcal{R} \mid \mathfrak{I}, r \models \chi\}$. In the following we will often identify $\mathcal{N}(\chi)$ with χ and omit $\mathfrak{I}$ when clear from context.*

Let $\mathfrak{I}$ be a model, χ be a norm, and m be a monitor. Now we are ready to reason about norm violations. As said in the previous section, a run r is an $\mathcal{N}$-violation iff $r \notin \mathcal{N}(\chi)$. In this setting, this is equivalent to saying that r is an $\mathcal{N}$-violation iff $\mathfrak{I}, r \models \neg\chi$. However, this violation may not be detected by monitor m; this heavily depends on the capabilities of the monitor and what it can observe. Intuitively, a norm violation for a run r is detected if in all the possibilities that the monitor considers, a violation occurs, i.e. whenever $m(r) \subseteq \mathcal{R}\backslash\mathcal{N}(\chi)$ holds. This statement is equivalent to saying that a violation is detected if $\mathfrak{I}, m(r) \models \neg\chi$. In the case of a non-ideal monitor the classification is not that simple: some runs in $m(r)$ may violate the norm whereas others do not.

Definition 8 (Detection of norm violation). *Let $\mathfrak{I}$ be an interpreted transition system, $r \in \mathcal{R}$, χ be a norm, and m be a monitor. We say that monitor m on input r returns (or detects a)*

- χ-violation *iff $\mathfrak{I}, m(r) \models \neg\chi$;*
- χ-compliance *iff $\mathfrak{I}, m(r) \models \chi$; and*
- χ-indifference *iff both $\mathfrak{I}, m(r) \not\models \chi$ and $\mathfrak{I}, m(r) \not\models \neg\chi$.*

A run is evaluated with respect to a given monitor and norm. However, if the monitor detects a possible norm violation, the norm may actually not be violated. This can happen if the monitor does not accurately detect the current run.

Definition 9 (Classification error). *Let $\mathfrak{I}$ be a transition system, χ a norm, and m a monitor. We say that m makes a* χ-classification error *on r iff*

- *m detects a χ-violation on r and r is not a χ-violation (*false negative *classification error); or*
- *m detects a χ-compliance on r and r is a χ-violation (*false positive *classification error).*

Ultimately, we are interested in monitors which detect all norm violations and only those.

Definition 10 (Sound, complete, sufficient). *Let $\mathfrak{I}$ be a transition system, χ a norm and m a monitor. We say that m is*

- χ-sound *in $\mathfrak{I}$ iff for all $r \in \mathcal{R}$ it holds: $\mathfrak{I}, m(r) \models \neg\chi$ implies $\mathfrak{I}, r \models \neg\chi$. In words, a monitor is sound with respect to a norm χ when for every possible run of the system, it holds that whenever a χ-violation is detected it is the case that the run was a χ-violation.*
- χ-complete *in $\mathfrak{I}$ iff for all $r \in \mathcal{R}$ it holds: $\mathfrak{I}, r \models \neg\chi$ implies $\mathfrak{I}, m(r) \models \neg\chi$. In words, a monitor is complete with respect to a norm χ when for every possible run of the system, it holds that whenever the run was a χ-violation it is the case that a χ-violation is detected.*
- χ-sufficient *in $\mathfrak{I}$ if m is χ-sound and χ-complete.*

Again, we omit $\mathfrak{I}$ if the model is clear from context.

To summarize some of these definitions, let us return to our example.

Example 2. *We return to the transition system found in Example 1, but now also assume we have a set of atomic propositions $\mathit{Props} = \{p\}$. Together with the valuation function v defined as $v(q_0) = v(q_1) = \{p\}$ and $v(q_2) = \emptyset$, we get an interpreted transition system $\mathfrak{I}$. The norm can now be specified as an LTL formula $\chi = \Box p$. Let us also consider that we still have the same monitor m, i.e. $m(r) = \{r' \in \mathcal{R} \mid r[1] = r'[1]\}$. It is now easy to verify that this monitor is χ-sound; whenever for a given run r it holds that $\mathfrak{I}, m(r) \models \neg\Box p$ then certainly $\mathfrak{I}, r \models \neg\Box p$. However, this monitor is not χ-complete. In particular, consider the run $r = q_0 q_1 (q_0 q_2)^\omega$. It is clear that for this run it holds that $\mathfrak{I}, r \models \neg\Box p$, however it also holds that $\mathfrak{I}, m(r) \not\models \neg\Box p$; this is because the run $(q_0 q_1)^\omega$ is also in the set $m(r)$ and this run is not a χ-violation. Thus, for this run the monitor detects a χ-indifference, but it should have detected a χ-violation for it to not break the property of χ-completeness.*

3. LTL-BASED MONITORS

We introduce monitors built from LTL-formulae. Firstly, we introduce (basic) *LTL-monitors* which use a single LTL-formula to classify runs of the system. Then, in Section 3.3 we consider how to combine these monitors to obtain more expressive ones. We call the latter type *combined LTL-monitors* and refer to both types as LTL-based monitors.

3.1 Basic LTL-Monitors

In this section we will discuss the topic of using LTL formulas in order to characterize monitors. Given an LTL-formula ξ a *ξ-monitor* can distinguish runs which satisfy ξ from those which do not satisfy ξ. This is a much simpler characterization of a monitor since it only divides the set of runs into two classes; the set of runs that satisfy ξ and the set of runs that do not. Intuitively, if two runs r and r' both satisfy ξ, then it holds that $m(r) = m(r')$. The converse also holds, if they do not satisfy ξ then the observed sets are also the same. We can define this formally as follows.

Definition 11 (ξ-monitor). *Let ξ be an LTL-formula and $\Im$ be a transition system. The* ξ-monitor *over $\Im$ is the function $m_\xi : \mathcal{R} \to \mathcal{P}(\mathcal{R})$ defined as follows: $m_\xi(r) := \{r' \in \mathcal{R} \mid \Im, r \models \xi \text{ iff } \Im, r' \models \xi\}$. Again, we omit $\Im$ if clear from context.*

Let us briefly return to our example to see how we can associate an LTL-formula with a monitor.

Example 3. *Returning to the interpreted transition system found in Example 2, we could have also defined this monitor as a $\bigcirc p$-monitor. Both this monitor and the monitor in the example for any arbitrary run r return exactly the same set $m(r)$.*

3.2 Examples of LTL-Monitors

We assume that we have a set of propositions C denoting the set of state conditions under which something can be monitored and a set of propositional formulae P denoting the state properties which can be verified. Both sets are assumed to be fixed and motivated below. The intuition here is that a monitor cannot always under any circumstances validate any property. E.g. a monitor that measures whether a car is exceeding the speed limit can only do this if the car is in front of the monitor. Moreover, under this condition not any property can be checked; perhaps the monitor can observe the speed, but it cannot observe whether the driver is wearing his/her seatbelt. We assume that every LTL-monitor has the form $\Box(\varphi_c \to \varphi_p)$ such that φ_c (respectively φ_p) is an LTL-formula consisting only of temporal operators in combination with propositional formulae from C (respectively P).

State Monitors. We call a $\Box(\psi_c \to \psi_p)$-monitor a *state monitor* where $\psi_c \in C$ and $\psi_p \in P$. Intuitively, this monitor checks under each occurance of ψ_c whether the propositional formula ψ_p holds. Thus this monitor guarantees that at each moment in time in a run of the system, either the event ψ_c has not occurred, or if it has, the validity of the property ψ_p is guaranteed. The class of all possible state monitors is given by $\mathsf{M}_{state}(C, P)$

Transition Monitors. We call a $\Box((\psi_c \wedge \bigcirc\psi_c') \to (\psi_p \wedge \bigcirc\psi_p'))$-monitor a *transition monitor* where $\psi_c, \psi_c' \in C$ and $\psi_p, \psi_p' \in P$. This monitor checks at any moment in time if the conditions ψ_c and ψ_c' are succinctly met and, if so, whether ψ_p and ψ_p' are succinctly satisfied. The class of all possible transitions monitors is given by $\mathsf{M}_{trans}(C, P)$

Sequence Monitors. We call a $\Box((\bigwedge_{i=0}^{k} \bigcirc^i \psi_c^i) \to (\bigwedge_{i=0}^{k} \bigcirc^i \psi_p^i))$-monitor (where $\psi_c^j \in C, \psi_p^j \in P, 0 \leq j \leq k$, $k \in \mathbb{N}_0$ and $\bigcirc^i := \underbrace{\bigcirc \ldots \bigcirc}_{i\text{-times}}$) a *sequence monitor* of length k. Notice that for $k = 0$ this definition reduces to state monitors and for $k = 1$ to transition monitors. The class of all possible sequences monitors (for arbitrary k) is given by $\mathsf{M}_{seq}(C, P)$.

3.3 Combinations of LTL-Monitors

In Section 3.1 we have introduced monitors that are built from LTL-formulae. As we will show in Section 4, a monitor m_ξ can classify the space of all system behaviours into two classes: one class contains runs satisfying ξ and the other one runs which do not. Moreover, we show that, in case of a sufficient monitor, these (equivalence) classes must *exactly* coincide with the corresponding equivalence classes emerging from the norm (cf. Theorem 1). This also implies that the formula ξ must be very similar to the norm itself; hence, if the norm is a "complicated" (in terms of expressivity) formula, ξ will most probably be one too. Therefore, we propose a combination of monitors. The idea is that we start with simple monitors and combine them to more complex ones without using more complex formulae. Firstly, we formally introduce the concept of combined monitors.

Definition 12 ($\oplus$, Σ-monitor). *Let $m_1, m_2 : \mathcal{R} \to \mathcal{P}(\mathcal{R})$ be two monitors. We define the monitor $m \oplus m' : \mathcal{R} \to \mathcal{P}(\mathcal{R})$ as follows: $m \oplus m'(r) := m(r) \cap m'(r)$.*

Let Σ be a set of LTL-formulae. The (combined) Σ-monitor *is the function $m_\Sigma : \mathcal{R} \to \mathcal{P}(\mathcal{R})$ with $m_\Sigma(r) = \oplus_{\xi\in\Sigma} m_\xi(r)$.*

The following result is clear from the definition of m_Σ.

Proposition 2. *The operator $\oplus$ is associative and commutative.*

To illustrate the concept of combined monitors, we consider the following example.

Example 4. *Consider a transition system $\Im$ with five states $\{q_0, \ldots, q_4\}$, a set of runs $\mathcal{R}_\Im = \{q_0(q_1)^\omega, q_0(q_2)^\omega, q_0(q_3)^\omega, q_0(q_4)^\omega\}$ and labeling $v(q_0) = v(q_4) = \emptyset$, $v(q_1) = \{r, s\}$, $v(q_2) = \{r\}$, $v(q_3) = \{s\}$. We refer to run $q_0(q_i)^\omega$ with r_i, for $i = 1, \ldots 4$. Suppose the norm is given by $\chi = \bigcirc((r \wedge \neg s) \vee (\neg r \wedge s))$. Consider the two LTL-monitors $m_{\bigcirc r}$ and $m_{\bigcirc s}$. Then, we have that $m_{\bigcirc r}(r_1) = m_{\bigcirc r}(r_2) = \{r_1, r_2\}$ and $m_{\bigcirc r}(r_3) = m_{\bigcirc r}(r_4) = \{r_3, r_4\}$ and $m_{\bigcirc s}(r_1) = m_{\bigcirc s}(r_3) = \{r_1, r_3\}$ and $m_{\bigcirc s}(r_2) = m_{\bigcirc s}(r_4) = \{r_2, r_4\}$. Clearly, none of these monitors is χ-sufficient. Let $\Sigma = \{\bigcirc r, \bigcirc s\}$, then for the monitor m_Σ we have $m_\Sigma(r_i) = \{r_i\}$ for $i = 1, \ldots 4$. That is, the monitor perfectly classifies all runs and is χ-sufficient. We note that the LTL-monitor $m_{\bigcirc r \wedge \bigcirc s}$ is* not *χ-sufficient.*

The next proposition shows that there are combined monitors for which there is no equivalent non-combined LTL-monitor; that is, combined monitors are strictly more expressive.

Proposition 3. *The class of LTL-monitors is not closed under operator $\oplus$.*

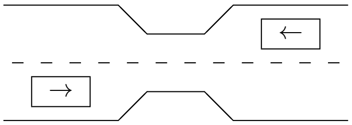

Figure 2: Schematic view to the road example.

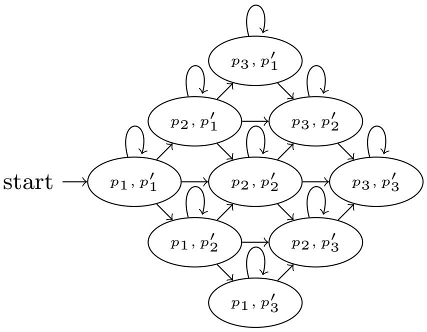

Figure 3: Transition system $\mathfrak{I}$ of the road example.

Although this intuitively already follows from Example 4, the actual proof of this proposition is given in Proposition 6 where we show that the number of classes a monitor partitions the runs in can be strictly higher for combined monitors than for non-combined monitors.

3.4 An Example Scenario

In this example we assume that there are two cars approaching a bottleneck in the road, both coming from opposite directions and want want to reach the other side, cf. Figure 2. However, to avoid the dangerous situation where both cars are on the bottleneck at the same time, there is a norm in place that states that (1) When two cars are on opposite sides of the bottleneck, cars coming from the right have priority over cars coming from the left, and (2) No two cars should simultaneously be on the bottleneck. We can associate with each car three different general locations: (1) Just before the bottleneck; (2) on the bottleneck; and (3) after the bottleneck. We denote this by the propositions (1) p_1, (2) p_2 and (3) p_3 for the car coming from the left, and (1) p_1', (2) p_2' and (3) p_3' for the car coming from the right. Assuming that the cars can either wait or move forward (and not reverse), the transition system is shown in Figure 3. Any traversal of the transition system gives rise to a run. R is the set of all these runs. Given this model, we state the corresponding norm:

$$\chi \equiv \Box((p_1 \wedge p_1') \to \neg \bigcirc p_2) \wedge \Box(\neg p_2 \vee \neg p_2').$$

In the following we consider some monitors. Following Section 3.2, we define set $C = \{p_1, p_2, p_3\}$ and $P = \{p_1', p_2', p_3'\}$ of conditions under which some property can be monitored and of properties to be verified, respectively. The sets C and P allow to express each possible position of the car coming from the left (p_1, p_2, p_3) we can verify each of the positions of the car coming from the right (p_1', p_2', p_3').

Example 5. *Consider the state monitor $m_{\Box(p_3 \to p_3')} \in \mathsf{M}_{state}(C, P)$. This monitor is intuitively understood as a monitor that detects whether the car from the right crosses the bottleneck before the car from the left does. For each run going through the state (p_3, p_1') or (p_3, p_2') it holds that this monitor detects a χ-violation. To see this, each of the runs that satisfy $\neg\Box(p_3 \to p_3')$ goes from (p_1, p_1') to (p_2, p_1') or goes through the state (p_2, p_2'). However, this monitor is not χ-complete (i.e. it does not detect all χ-norm violations): e.g. the run $(p_1, p_1')(p_2, p_2')((p_3, p_3'))^\omega$ is not detected, even though it violates the norm.*

Example 6. *Consider the three monitors m_1, m_2 and m_3 with $m_1 = m_{\Box((p_1\wedge\bigcirc p_2)\to(p_1'\wedge\bigcirc p_1'))}$, $m_2 = m_{\Box((p_1\wedge\bigcirc p_2)\to(p_2'\wedge\bigcirc p_2'))}$ and $m_3 = m_{\Box((p_1\wedge\bigcirc p_2)\to(p_1'\wedge\bigcirc p_2'))}$. We have that $m_1, m_2, m_3 \in \mathsf{M}_{trans}(C, P)$. These monitors are, just like the previously discussed monitor, not χ-complete. However, the monitor obtained by combining these monitors, $m_1 \oplus m_2 \oplus m_3$, is χ-complete and even χ-sufficient. Although not completely trivial, the reasoning behind it is that any run going from (p_1, p_1') to (p_2, p_1') is detected by m_1, any run going from (p_1, p_2') to (p_2, p_2') is detected by m_2 and finally any run going from (p_1, p_1') to (p_2, p_2') by m_3. Moreover, these are exactly all the runs which are χ-violations.*

4. MONITORING AND PROPERTIES

In this section we discuss properties of LTL-based monitors. Our main results are characterization theorems for LTL-monitors (Theorem 1) and for combined LTL-monitors (Theorem 2). They are later used to solve the decision problems whether there are sound and complete monitors. Of utmost importance is that monitors *partition* the set of runs in a system. Therefore we introduce the following notation:

Definition 13. *Given a set X and a function $f : X \to \mathcal{P}(X)$, we say that f is an* equivalence X-classifier *if there is a partition $(X_i)_{i\in I}$ of X (that is, $I \subseteq \mathbb{N}_0$ and each $X_i \subseteq X$, $\cup_{i\in I} X_i = X$, and $X_i \cap X_j = \emptyset$ for $i, j \in I$ and $i \neq j$), and for all $x \in X$, $f(x) = X_i$ whenever $x \in X_i$ for $i \in I$. A* binary X-classifier *is an equivalence X-classifier with only two partitions (note, possibly some of which is the empty set). We let $\mathcal{C}(f)$ denote the set of equivalence classes of f.*

Let us first discuss properties of non-combined LTL-monitors. Some propositions and theorems of these monitors also apply to combined monitors; in these particular cases we refer to the proofs in the next section of combined monitors which are a generalization of the non-combined cases.

4.1 Properties: Non-Combined LTL-Monitors

We first show that an LTL-monitor is a binary equivalence classifier. This means that a monitor partitions the set into two classes; one class that satisfies the formula associated with the monitor and one class that satisfies the negation of this formula.

Proposition 4. *For each transition system $\mathfrak{I}$ and LTL-formula ξ, m_ξ is a binary equivalence $\mathcal{R}$-classifier; hence, $|\mathcal{C}(m_\xi)| = 2$.*

Proof. This is proven in the next section where we consider the more general case of arbitrarily combined monitors, i.e. this proof is a special case of Proposition 6 for monitor $m_{\{\xi\}}$. □

Moreover, we are able to show that an LTL-monitor is always correct, consistent, not broken and χ-sound for any LTL-norm χ and transition system $\mathfrak{I}$.

Proposition 5. *Let χ be an LTL-norm and $\mathfrak{I}$ be a transition system. The monitor m_ξ is correct, consistent and not broken. Moreover, each LTL-monitor m_ξ is χ-sound over $\mathfrak{I}$.*

Proof. Again, this is proven in the next section where we consider the more general case. This proof is a special case of Proposition 7 for monitor $m_{\{\xi\}}$. □

Since χ-soundness trivially holds for an LTL monitor, proving that a monitor is χ-sufficient amounts to checking whether this monitor is χ-complete.

Corollary 1. *A χ-complete monitor m_ξ is already χ-sound and χ-sufficient.*

The next proposition characterizes exactly when a monitor is sufficient for a norm.

Theorem 1 (Characterization of LTL-monitors). *Let m_ξ be an LTL-monitor, χ an LTL-norm, and $\mathfrak{I}$ be a transition system. Then, the following statements are equivalent:*

(a) m_ξ is χ-sufficient in $\mathfrak{I}$.
(b) if $\neg\chi \wedge \xi$ is satisfiable on $\mathfrak{I}$ then $\mathfrak{I} \models \xi \rightarrow \neg\chi$; and
if $\neg\chi \wedge \neg\xi$ is satisfiable on $\mathfrak{I}$ then $\mathfrak{I} \models \neg\xi \rightarrow \neg\chi$.
(c) $\mathfrak{I} \models \neg\chi$ or $\mathfrak{I} \models \chi$ or $\mathfrak{I} \models \neg\xi \leftrightarrow \chi$ or $\mathfrak{I} \models \xi \leftrightarrow \chi$.
(d) $\mathcal{N}_{\mathfrak{I}}(\chi) = \bigcup\{X \in \mathcal{C}(m_\xi) \mid \mathfrak{I}, X \models \chi\}$.

Proof. This theorem is a special case of Theorem 2 for $m_{\{\xi\}}$. To see this we note that $\mathsf{Con}(\Sigma) = \{\xi, \neg\xi\}$ (we refer to Section 4.2 for the notation). For (c) we observe that for $\hat{\Sigma} = \emptyset$ (where $\hat{\Sigma}$ is defined in the theorem) we get $\mathfrak{I} \models \bot \leftrightarrow \neg\chi$ which is equivalent to $\mathfrak{I} \models \chi$. For $\hat{\Sigma} = \mathsf{Con}(\Sigma)$, we get $\mathfrak{I} \models (\xi \vee \neg\xi) \leftrightarrow \neg\chi$ which is equivalent to $\mathfrak{I} \models \neg\chi$. □

4.2 Properties: combined LTL-monitors

In this section we discuss properties of the more general case of combined monitors. In what follows, if not said otherwise, we assume that $\Sigma = \{\xi_1, \ldots, \xi_n\}$ where each ξ_i, for $1 \leq i \leq n$ is an LTL-formula. Before we present the characterization theorem for combined monitors we need some additional notation and lemmata. We use $\mathsf{Con}(\Sigma)$ to denote the set of all conjunctions of formulae from Σ. Each conjunction from $\mathsf{Con}(\Sigma)$, denoted $\hat{\xi}$, contains for each formulae $\xi \in \Sigma$ either ξ itself or $\neg\xi$. Moreover, we assume an order on Σ and lift it to conjunctions from $\mathsf{Con}(\Sigma)$ in the natural way to have a well-defined representative.

Definition 14. *Let $X = \{\xi_1, \ldots, \xi_n\}$ be a finite set of formulae. We define $\mathsf{Con}(\emptyset) := \{\bot\}$ and $\mathsf{Con}(X) := \{\xi'_1 \wedge \cdots \wedge \xi'_n \mid \xi'_i \in \{\xi_i, \neg\xi_i\}$ for $i = 1, \ldots, n\}$.*

We can now prove that any combination of LTL-monitors is an equivalence classifier.

Proposition 6. *For each interpreted transition system $\mathfrak{I}$ and finite set of LTL-formulae Σ, the monitor m_Σ is an equivalence classifier with $|\mathcal{C}(m_\Sigma)| \leq 2^{|\Sigma|}$.*

Proof. Given $\hat{\xi} \in \mathsf{Con}(\Sigma)$, let $X_{\hat{\xi}} = \{r \in \mathcal{R} \mid \mathfrak{I}, r \models \hat{\xi}\}$. Clearly it holds that $\bigcup_{\hat{\xi} \in \mathsf{Con}(\Sigma)} X_{\hat{\xi}} = \mathcal{R}$ and given $\hat{\xi}, \hat{\xi}' \in \mathsf{Con}(\Sigma)$ if $\mathfrak{I} \models \hat{\xi} \leftrightarrow \hat{\xi}'$ then we have $X_{\hat{\xi}} = X_{\hat{\xi}'}$, and if $\mathfrak{I} \not\models \hat{\xi} \leftrightarrow \hat{\xi}'$ then $X_{\hat{\xi}} \cap X_{\hat{\xi}'} = \emptyset$; thus $\{X_{\hat{\xi}} \subseteq \mathcal{R} \mid \hat{\xi} \in \mathsf{Con}(\Sigma)\}$ forms a partition of $\mathcal{R}$. Now let $r \in X_{\hat{\xi}'}$ for an arbitrary $\hat{\xi}' = \xi'_1 \wedge ... \wedge \xi'_n$. Then, for all $r' \in \mathcal{R}$ we have $r' \in m_\Sigma(r)$ iff $\mathfrak{I}, r' \models \xi'_1$ and ... and $\mathfrak{I}, r' \models \xi'_n$ iff $\mathfrak{I}, r' \models \hat{\xi}'$ iff $r' \in X_{\hat{\xi}'}$. This shows that $m_\xi(r) = X_{\hat{\xi}'}$. The number of (unique) elements in $\mathsf{Con}(\Sigma)$ is at most $2^{|\Sigma|}$, thus we have $|\mathcal{C}(m_\Sigma)| \leq 2^{|\Sigma|}$. □

This proposition highlights the fact that for a transition system $\mathfrak{I}$ and monitor m_Σ, if it holds for $\xi, \xi' \in \Sigma$ that $\xi \neq \xi'$, and $\mathfrak{I} \models \xi \leftrightarrow \xi'$ or $\mathfrak{I} \models \neg\xi \leftrightarrow \xi'$, then $m_{\Sigma\setminus\{\xi\}}$ and $m_{\Sigma\setminus\{\xi'\}}$ are equivalent to m_Σ (i.e. for every $r \in \mathcal{R}$ it holds that $m_\Sigma(r) = m_{\Sigma\setminus\{\xi\}}(r) = m_{\Sigma\setminus\{\xi'\}}(r)$). This leads us to the following definition, which will be used in the proof of Theorem 2.

Definition 15. *Given a transition system $\mathfrak{I}$ and set of formulae X, let $X_{\mathfrak{I}} \subseteq X$ be the largest subset such that for all $\xi \in X_{\mathfrak{I}}$ there is no $\xi' \in X_{\mathfrak{I}}$ with $\xi \neq \xi'$ such that $\mathfrak{I} \models \xi \leftrightarrow \xi'$ or $\mathfrak{I} \models \neg\xi \leftrightarrow \xi'$. In general, such a subset is not unique; in this case we choose one arbitrarily.*

Again we note that m_Σ and $m_{\Sigma_{\mathfrak{I}}}$ are exactly the same monitors, so we use the sets Σ and $\Sigma_{\mathfrak{I}}$ interchangeably whenever the transition system $\mathfrak{I}$ is fixed. Analogously to Proposition 5 and Corollary 1, we get the following results:

Proposition 7. *Let χ be an LTL-norm and $\mathfrak{I}$ be a transition system. The combined monitor m_Σ is correct, consistent and not broken. Moreover, each combined LTL-monitor m_Σ is χ-sound over $\mathfrak{I}$.*

Proof. That a Σ-monitor is correct, consistent and not broken follows from the fact that m_Σ is an equivalence classifier, cf. Prop. 6. For the soundness, we assume the contrary for the sake of contradiction. Then, there is a run r such that $\mathfrak{I}, m_\Sigma(r) \models \neg\chi$ and $\mathfrak{I}, r \models \chi$. But by correctness we have that $r \in m_\xi(r)$ and hence $\mathfrak{I}, r \models \neg\chi$. Contradiction. □

Corollary 2. *A χ-complete monitor m_Σ is already χ-sound and χ-sufficient.*

Again note that this corollary trivially holds, since any combination of LTL-monitors is already χ-sound. The following two lemmata are needed to prove Theorem 2.

Lemma 1. *Let $\mathfrak{I}$ be a transition system, $X_{\mathfrak{I}}$ be a finite set of formulae as defined in Definition 15 and let $\hat{U} \subseteq \mathsf{Con}(X_{\mathfrak{I}})$. Then, the following formula is valid over $\mathfrak{I}$:*
$\mathfrak{I} \models \bigwedge_{\hat{\xi} \in \hat{U}} \neg\hat{\xi} \leftrightarrow \bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \hat{\xi}$.

Proof. We prove this by syntactic rewriting. In the following, we use the fact that (1) $\bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})} \hat{\xi} = \top$ and (2) given $\hat{\xi}, \hat{\xi}' \in \mathsf{Con}(X_{\mathfrak{I}})$ with $\hat{\xi} \neq \hat{\xi}'$, it holds that $\hat{\xi} \wedge \hat{\xi}' = \bot$. The reason for the latter is because we can always find a $\xi_i \in X$ which is true in $\hat{\xi}$ and false in $\hat{\xi}'$, or vice versa. In the following we omit mentioning $\mathfrak{I}$: $\bigwedge_{\hat{\xi} \in \hat{U}} \neg\hat{\xi} \leftrightarrow \bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \hat{\xi} \Leftrightarrow \left(\neg \bigvee_{\hat{\xi} \in \hat{U}} \hat{\xi} \vee \neg \bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \hat{\xi}\right) \wedge \left(\bigvee_{\hat{\xi} \in \hat{U}} \hat{\xi} \vee \bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \hat{\xi}\right) \Leftrightarrow \left(\bigwedge_{\hat{\xi} \in \hat{U}} \neg\hat{\xi} \vee \bigwedge_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \neg\hat{\xi}\right) \wedge \left(\bigvee_{\hat{\xi} \in \mathsf{Con}(X_{\mathfrak{I}})} \hat{\xi}\right) \Leftrightarrow \left(\bigwedge_{\hat{\xi} \in \hat{U}} \bigwedge_{\hat{\xi}' \in \mathsf{Con}(X_{\mathfrak{I}})\setminus\hat{U}} \left(\neg\hat{\xi} \vee \neg\hat{\xi}'\right)\right) \Leftrightarrow \top$ □

Let us consider the combined LTL-monitor $m_{\Sigma_{\mathfrak{I}}}$ and a run $r \in \mathcal{R}$. For each monitor m_ξ with $\xi \in \Sigma_{\mathfrak{I}}$, r defines one of two equivalence classes (cf. Proposition 4). Hence, the run r does also uniquely define an equivalence class of the monitor $m_{\Sigma_{\mathfrak{I}}}$ which corresponds to the intersection of all uniquely defined equivalence classes of each m_ξ. Formally, we have:

Lemma 2. *For each run $r \in \mathcal{R}_{\mathfrak{I}}$ there is a unique $\hat{\xi} \in \mathsf{Con}(\Sigma_{\mathfrak{I}})$ with $\mathfrak{I}, r \models \hat{\xi}$. We denote this formula by $\hat{\xi}(r)$.*

Proof. Clearly, for each $\xi \in \Sigma_{\mathfrak{I}}$ either $\mathfrak{I}, r \models \xi$ or $\mathfrak{I}, r \models \neg\xi$. Then, $\hat{\xi}(r) = \xi'_1 \wedge \cdots \wedge \xi'_n$ with $\xi'_i \in \{\xi_i, \neg\xi_i\}$ and $\xi'_i = \xi_i$ iff $\mathfrak{I}, r \models \xi_i$. It is also clear that there can be no other formula $\hat{\xi}' \in \mathsf{Con}(\Sigma_{\mathfrak{I}})$ with $\mathfrak{I}, r \models \hat{\xi}'$. □

Finally, we turn to our main result about combined LTL-monitors. It is a generalization of Theorem 1 (cf. the proof of that theorem). Of particular interest is characterization (c). It says that a monitor is χ-sufficient if there is a set $\hat{\Sigma}$, where each $\hat{\xi} \in \hat{\Sigma}$ characterizes one equivalence class of $m_{\Sigma_\mathfrak{I}}$, such that the disjunction of all these formulae is true iff the norm is violated. In other words, the union of all these equivalence classes must contain *exactly* the runs violating the norm. Moreover, characterization (b) is useful to devise a decision procedure (cf. Proposition 10).

Theorem 2 (Charact. of combined LTL-monitors). *Let $\Sigma = \{\xi_1, \ldots, \xi_n\}$ be a non-empty and finite set of consistent LTL-formulae, χ be an LTL-norm, and $\mathfrak{I}$ be a transition system. Then, the following statements are equivalent:*

(a) m_Σ is χ-sufficient over $\mathfrak{I}$.

(b) for all $\hat{\xi} \in \mathsf{Con}(\Sigma)$, if $\neg\chi \wedge \hat{\xi}$ is satisfiable on $\mathfrak{I}$ then $\mathfrak{I} \models \hat{\xi} \to \neg\chi$.

(c) $\mathfrak{I} \models (\bigvee_{\hat{\xi}\in\hat{\Sigma}} \hat{\xi}) \leftrightarrow \neg\chi$ where $\hat{\Sigma} = \{\hat{\xi} \in \mathsf{Con}(\Sigma) \mid \mathfrak{I} \models \chi \vee \neg\hat{\xi}\}$.

(d) $\mathcal{N}_\mathfrak{I}(\chi) = \bigcup\{X \in \mathcal{C}(m_\Sigma) \mid \mathfrak{I}, X \models \chi\}$.

Proof. We prove (a)⇒ (b) ⇒(c) ⇒(d)⇒ (a).

__(a) ⇒ (b):__ Suppose (a) holds, i.e. $\forall r \in \mathcal{R} : \mathfrak{I}, r \models \neg\chi$ implies $\mathfrak{I}, m_\Sigma(r) \models \neg\chi$. Moreover, let $\mathfrak{I}, r \models \neg\chi \wedge \hat{\xi}$ for some r and $\mathfrak{I} \not\models \hat{\xi} \to \neg\chi$; i.e. $\mathfrak{I}, r' \models \hat{\xi} \wedge \chi$ for some r'. However, then $r' \in m_\Sigma(r)$; and hence, $\mathfrak{I}, m_\Sigma(r) \not\models \neg\chi$. Contradiction.

__(b) ⇒ (c):__ Let Σ^+ (resp. Σ^-) consist of all formulae $\hat{\xi} \in \mathsf{Con}(\Sigma_\mathfrak{I})$ with $\neg\chi \wedge \hat{\xi}$ satisfiable (resp. not satisfiable) over $\mathfrak{I}$. By (b) we have that $\mathfrak{I} \models \hat{\xi} \to \neg\chi$ for all $\hat{\xi} \in \Sigma^+$ and $\mathfrak{I} \models \neg\chi \to \neg\hat{\xi}$ for all $\hat{\xi} \in \Sigma^-$. This implies that $\mathfrak{I} \models (\bigvee_{\hat{\xi}\in\Sigma^+} \hat{\xi}) \to \neg\chi$ and $\mathfrak{I} \models \neg\chi \to (\bigwedge_{\hat{\xi}\in\Sigma^-} \neg\hat{\xi})$. Moreover, we have that $\mathsf{Con}(\Sigma_\mathfrak{I}) = \Sigma^+ \uplus \Sigma^-$. Then, by Lemma 1 we obtain $\mathfrak{I} \models \bigwedge_{\hat{\xi}\in\Sigma^-} \neg\hat{\xi} \to \neg\chi$ from $\mathfrak{I} \models (\bigvee_{\hat{\xi}\in\Sigma^+} \hat{\xi}) \to \neg\chi$ and thus $\mathfrak{I} \models \bigwedge_{\hat{\xi}\in\Sigma^-} \neg\hat{\xi} \leftrightarrow \neg\chi$.

__(c) ⇒ (d):__ Let $\mathfrak{I} \models (\bigvee_{\hat{\xi}\in\hat{\Sigma}_\mathfrak{I}} \hat{\xi}) \leftrightarrow \neg\chi$. "⊆": Let $r \in \mathcal{N}(\chi)$. Because $r \in m_\Sigma(r)$ by Prop. 7 it remains to show that $\mathfrak{I}, m_\Sigma(r) \models \chi$. We have that $m_\Sigma(r) = \{r' \in \mathcal{R}_\mathfrak{I} \mid \mathfrak{I}, r' \models \hat{\xi}(r)\}$ where $\hat{\xi}(r)$ is the unique formula from Lemma 2, clearly $\hat{\xi}(r) \leftrightarrow \hat{\xi}(r')$ for any $r' \in m_\Sigma(r)$. Now suppose that there is a run $r' \in m_\Sigma(r)$ with $\mathfrak{I}, r' \models \neg\chi$. Then, we must have $\hat{\xi}(r') \in \hat{\Sigma}$ which implies that $\mathfrak{I}, r \models \neg\chi$. Contradiction.

"⊇": If $r \in \bigcup\{X \in \mathcal{C}(m_\Sigma) \mid \mathfrak{I}, X \models \chi\}$ then clearly $\mathfrak{I}, r \models \chi$ and hence $r \in \mathcal{N}_\mathfrak{I}(\chi)$.

__(d) ⇒ (a):__ Suppose $\mathcal{N}_\mathfrak{I}(\chi) = \bigcup\{X \in \mathcal{C}(m_\Sigma) \mid \mathfrak{I}, X \models \chi\}$. By Proposition 7 we need to show that if $\mathfrak{I}, r \models \neg\chi$ then $\mathfrak{I}, m_\Sigma(r) \models \neg\chi$, for all $r \in \mathcal{R}_\mathfrak{I}$. Suppose $\mathfrak{I}, r \models \neg\chi$ holds and $\mathfrak{I}, m_\Sigma(r) \not\models \neg\chi$. Then, there is an $r' \in m_\Sigma(r)$ with $\mathfrak{I}, r' \models \chi$. By (d) there is a class $X \in \mathcal{C}(m_\Sigma)$ with $r' \in X$ and $\mathfrak{I}, X \models \chi$. However, since $r' \in m_\Sigma(r)$ iff $\hat{\xi}(r') \leftrightarrow \hat{\xi}(r)$ we must have $r \in X$. Contradiction. □

5. COMPUTATIONAL PROBLEMS

Given a norm the system designer would like to construct a monitor which is sound and complete for the norm, or in other words, detect all norm violations. Of course, in this setting it is interesting how difficult it is to construct such a monitor. In this section we present our preliminary results on computational complexity issues regarding norm violation detection. For a given interpreted transition system $\mathfrak{I}$ and LTL-norm χ we consider the following problems:

1. Does there exist an LTL-monitor m_ξ which is χ-sufficient over $\mathfrak{I}$?
2. Is a given monitor m (pure LTL-based or combined) χ-sufficient over $\mathfrak{I}$?
3. Is there a monitor in M, where M is a set of monitors, which is χ-sufficient over $\mathfrak{I}$?
4. Can we combine monitors $m_1, \ldots, m_k \in M$ in such a way that $m_1 \oplus \ldots \oplus m_k$ is χ-sufficient?

By Proposition 5 the first question is trivial: every $m_{\neg\xi}$-monitor is ξ-sound and complete. More interesting are the three remaining questions which we answer in the following. We define the size of a (interpreted) transition system as the number of states and transitions in $\mathfrak{I}$.

Results for LTL-Monitors. According to Corollary 1 it is enough to check wether a monitor is χ-complete which implies sufficiency. The characterization theorem for LTL-based monitors (cf. Theorem 1) provides us with a polynomial-space decision procedure for the second question.

Proposition 8. *Let $\mathfrak{I}$ be a transition system, m_ξ an LTL-monitor, and χ an LTL-norm. The problem whether m_ξ is χ-sufficient over $\mathfrak{I}$ is* **PSPACE***-complete in the length of $\mathfrak{I}$, ξ, and χ.*

Proof. Membership can be checked in **PSPACE** by Theorem 1(c) using standard LTL model checking.

We reduce LTL model checking, a **PSPACE**-complete problem [8], to our problem. Let $\mathfrak{I}$ be a transition system and φ be an LTL-formula. Moreover, let p be a fresh proposition neither occurring in $\mathfrak{I}$ nor in φ and let $\mathfrak{I}_p$ be a copy of $\mathfrak{I}$ in which all states are additionally labeled p. Then, we have that ($\star$) $\mathfrak{I} \models \varphi$ iff $\mathfrak{I}_p \models \varphi$. Moreover, we have that ($\star\star$) $\mathfrak{I} \not\models \Box p$ and $\mathfrak{I}_p \models \Box p$. Let $\mathfrak{I}'$ be the disjoint union of $\mathfrak{I}$ and $\mathfrak{I}_p$ together with a new state q_0 which is only connected to all other states of (the copies of) $\mathfrak{I}$ and $\mathfrak{I}_p$. Now, we define $\chi \equiv \bigcirc(\varphi \vee \Box p)$ and $\xi = \top$. By Theorem 1(c) we have that m_ξ is χ-sufficient over $\mathfrak{I}'$ iff $\mathfrak{I}' \models \bigcirc(\neg\varphi \wedge \neg\Box p)$ or $\mathfrak{I}' \models \bigcirc(\varphi \vee \Box p)$. By ($\star$) and ($\star\star$) the left part of the disjunction (i.e. $\mathfrak{I}' \models \bigcirc(\neg\varphi \wedge \neg\Box p)$) cannot be true; hence, we must show that $\mathfrak{I}' \models \bigcirc(\varphi \vee \Box p)$. By construction of $\mathfrak{I}'$ this is the case iff $\mathfrak{I}' \models \bigcirc\varphi$ iff $\mathfrak{I} \models \varphi$. This gives the following polynomial-time reduction: $\mathfrak{I} \models^{LTL} \varphi$ iff $m_\top$ is $\bigcirc(\varphi \vee \Box p)$-sufficient over $\mathfrak{I}'$. □

The *third problem* is an easy extension of the first:

Proposition 9. *Let $\mathfrak{I}$ be a transition system, M a finite set of LTL-monitors, and χ an LTL-norm. The problem whether some $m \in M$ is χ-sufficient over $\mathfrak{I}$ is* **PSPACE***-complete in the length of $\mathfrak{I}$, M, and χ.*

Proof. We call the procedure from Prop. 8 $|M|$-times. Hardness is shown in the very same way as in Prop. 8 with $M = \{m_\top\}$. □

More interesting is the case, when class M of monitors is not given explicitly but by some compact description, like the syntactic descriptions introduced in Section 3.2. We leave this for future research.

Results for Combined LTL-Monitors. Now we turn to combined monitors. Again, by Corollary 2 it is enough to check wether a monitor is χ-complete which implies sufficiency. We consider the *second* problem, i.e., whether a given combined monitor is sound and complete:

Proposition 10. *Let $\mathfrak{I}$ be a transition system, m_Σ a combined LTL-monitor, and χ an LTL-norm. The problem whether m_Σ is χ-sufficient over $\mathfrak{I}$ is* **PSPACE***-complete in the length of $\mathfrak{I}$, Σ, and χ.*

Proof. To show that the problem is in **PSPACE** we use Theorem 2(b). The following procedure runs in nondeterministic polynomial space. This is sufficient for the result since **PSPACE** is closed under complement and nondeterminism. We guess a formula $\hat{\xi} \in \mathsf{Con}(\Sigma)$ and check whether $\neg\chi \wedge \hat{\xi}$ is satisfiable (this can be done in polynomial space by checking $\mathfrak{I} \not\models \neg(\neg\chi \wedge \hat{\xi})$) and whether $\mathfrak{I} \not\models \hat{\xi} \rightarrow \neg\chi$ (again, this can be done in polynomial space). If the answer is yes; then, m_Σ is not χ-sufficient over $\mathfrak{I}$.

Hardness obviously transfers. □

Again, the *fourth problem* can be solved by applying the previous one:

Proposition 11. *Let $\mathfrak{I}$ be a transition system, Σ be a non-empty and finite set of LTL-formulae, and χ an LTL-norm. The problem whether there is a non-empty subset $\Sigma' \subseteq \Sigma$ such that $m_{\Sigma'}$ is χ-sufficient over $\mathfrak{I}$ is* **PSPACE***-complete in the length of $\mathfrak{I}$, Σ, and χ.*

Proof. Firstly, we guess a subset $\Sigma' \subseteq \Sigma$ in polynomial space and apply Proposition 10. Hardness is shown in the very same way as in Prop. 8 with $M = \{m_\top\}$. □

6. RELATED WORK AND CONCLUSIONS

Our work can contribute to the research on monitoring program executions that aim at observing run-time behavior of programs, e.g. [4, 9, 5]. Examples of such monitors are debuggers, tracers, profilers and demons. Our analysis can be extended and applied to study run-time monitors and their properties. For example, in [4] a formal framework for run-time monitors is provided that allows specification, implementation, and reasoning about monitors. It proposes a systematic approach to integrate monitors in the standard semantics of a programming language such that the execution of a program based on the derived monitoring semantics provides monitoring information without changing the program behaviour. In contrast to our framework, the mentioned work does not study properties of monitors and their relations with norms, while our approach allows reasoning about monitors, norms, and program behaviours.

Another line of work focuses on monitoring as a runtime verification problem [9, 2]. The main problem formulated in these works is how to monitor certain temporal logic properties at run-time without storing an entire execution trace and as such, they consider finite traces of possibly increasing size. Since some LTL formulae can never be falsified on finite prefixes of using an infinite-trace semantics (such as until), they use extensions of LTL to reason about finite (timed) traces.

Our work is also related to studies that aim at monitoring constraints (closely related to norms) on databases over the course of time, e.g. [5]. The main aim of this work is to check the integrity constraints on a database with minimum knowledge about its history and possible future. Our work can be seen as a more general framework for analysing monitors with respect to a set of constraints that needs to be monitored. Monitoring constraints in our framework can be modelled by special types of monitors that have restricted knowledge about the past and future.

In this paper we developed a basic framework in which monitors can be studied and analysed. We proposed different types of monitors, provided a logical analysis of monitors, studied the relations between monitors and norms to be monitored, and explored some computational aspects of norm monitoring. The language on which these monitors and norms were built was LTL. As it turned out, combining these simple monitors allowed us to construct complex monitors with vastly growing reasoning capabilities. Moreover, showing that such a monitor is sufficient for a given norm (a violation is always correctly detected) still lies within the complexity bounds of LTL model checking itself.

For future work we plan to consider syntactic restrictions of specific classes of monitors which allow for efficient synthesis. Also, more classes of monitors can be considered, i.e. non-binary or non-correct monitors. With respect to the latter, an interesting study would be to use our framework to detect faulty monitors within a system. Finally, we plan to add *costs* to the framework and to study optimality properties. Finally, we would like to note that although our setting is defined over infinite runs it can also be given over finite runs. In the case of LTL monitors one can use a finite-trace semantics; in particular, a three-valued semantics similar to [2] would be interesting in the context of monitoring norm violations. We leave a detailed study for future research.

7. REFERENCES

[1] N. Alechina, M. Dastani, and B. Logan. Programming norm-aware agents. In *Proceedings AAMAS-12*, 2012.

[2] A. Bauer, M. Leucker, and C. Schallhart. Runtime verification for ltl and tltl. *ACM Trans. Softw. Eng. Methodol.*, 20(4):14:1–14:64, Sept. 2011.

[3] M. Dastani, D. Grossi, and J.-J. Meyer. A logic for normative multi-agent programs. *International Journal of Logic and Computation, special issue on Normative Multiagent Systems*, Published online on 14 September 14 2011.

[4] A. Kishon, C. Consel, and P. Hudak. Monitoring semantics: a formal framework for specifying, implementing and reasoning about execution monitors. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 338–352, June 1991.

[5] U. W. Lipeck and G. Saake. Monitoring dynamic integrity constraints based on temporal logic. *Information Systems*, 12(3):255–269, 1987.

[6] A. Pnueli. The temporal logic of programs. In *Proceedings of FOCS*, pages 46–57, 1977.

[7] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings AAAI-92*, San Diego, CA, 1992.

[8] A. P. Sistla and E. M. Clarke. The complexity of propositional linear temporal logics. *J. ACM*, 32(3):733–749, 1985.

[9] P. Thati and G. Rosu. Monitoring algorithms for metric temporal logic specifications. *Electr. Notes Theor. Comput. Sci.*, 113:145–162, 2005.