







# The Complexity of Norm Synthesis and Revision

Davide Dell'Anna<sup>1</sup> , Natasha Alechina<sup>2</sup> , Fabiano Dalpiaz<sup>2</sup> ,  
Mehdi Dastani<sup>2</sup> , Maarten Löffler<sup>2</sup>, and Brian Logan<sup>2,3</sup> 

<sup>1</sup> Delft University of Technology, Delft, The Netherlands  
d.dellanna@tudelft.nl

<sup>2</sup> Utrecht University, Utrecht, The Netherlands  
{n.a.alechina, f.dalpiaz, m.m.dastani, m.loffler, b.s.logan}@uu.nl

<sup>3</sup> University of Aberdeen, Aberdeen, Scotland

**Abstract.** Norms have been widely proposed as a way of coordinating and controlling the activities of agents in a multi-agent system (MAS). A norm specifies the behaviour an agent should follow in order to achieve the objective of the MAS. However, designing norms to achieve a particular system objective can be difficult, particularly when there is no direct link between the language in which the system objective is stated and the language in which the norms can be expressed. In this paper, we consider the problem of synthesising a norm from traces of agent behaviour, where each trace is labelled with whether the behaviour satisfies the system objective. We show that the norm synthesis problem and several related problems are NP-complete.

## 1 Introduction

There has been a considerable amount of work on using norms to coordinate the activities of agents in a multi-agent system (MAS) [11]. Norms can be viewed as standards of behaviour which specify that certain states or sequences of actions in a MAS should occur (obligations) or should not occur (prohibitions) in order for the objective of the MAS to be realized [9]. We focus on conditional norms with deadlines which express behavioral properties [35]. Conditional norms are triggered (detached) in certain states of the MAS and have a temporal dimension specified by a deadline, which is also a state property. The satisfaction or violation of a detached norm depends on whether the behaviour of the agent brings about a specified state before a state in which the deadline condition is true. Conditional norms are implemented in a MAS through enforcement. That is, violation of a norm results in either the behaviour being pre-empted (regimented, [5]), or in the violating agent incurring a sanction, e.g., a fine. See, e.g., [14] for how to determine an appropriate level of sanction.

For many applications it is assumed that the MAS developer will design an appropriate norm to realise the system objective. However, this can be difficult, particularly when the internals of the agents are unknown, e.g., in the case of open MAS [6], and when there is no direct connection between the language in which the system objective is stated and the language in which norms can be expressed. For example, one objective of a traffic system may be to avoid traffic collisions, but ‘not colliding’ is not a property under direct agent control, and prohibition of collisions cannot be stated as a norm.

A poorly designed norm may fail to achieve the system objective, or have undesirable side effects, e.g., the objective is achieved, but the autonomy of the agents is restricted unnecessarily.

The increasing availability of large amounts of system behaviour data [1, 23] introduces the possibility of a new approach to the design of norms, namely the synthesis of norms directly from data collected during the execution of the system. For example, data may show that collisions always happen when an agent’s speed is very high, allowing us to state a norm prohibiting agents from speeding too much. In this paper, we consider the problem of synthesising conditional norms with deadlines from traces of agent behaviour, where each trace is labelled with whether the behaviour satisfies the system objective.

The contributions of this paper are the following.

- We show that synthesising a conditional norm with deadline (i.e., an obligation or a prohibition) that correctly classifies the traces (i.e., the norm is violated on traces where the behaviour does not satisfy the system objective, and is not violated on other traces) is an NP-complete problem.
- We show that analogous complexity results (NP-completeness) also hold for the problem of the synthesis of sets of conditional norms with deadlines.
- We also consider the problem of synthesizing a norm that is “close” to a target norm. This problem is relevant where there is an existing norm that does not achieve the system objective, but which is accepted, e.g., by human users of a system, and we wish a minimal modification that does achieve the objective. We show that the minimal norm revision problem is also NP-complete.

This paper is organized as follows. Section 2 provides the necessary formal background on conditional norms and on traces of agent behaviours. Section 3 discusses the complexity of the problem of synthesising a single conditional norm. Section 4 discusses the complexity of synthesising a set of conditional norms. Section 5 discusses the complexity of the minimal norm revision. Section 6 discusses related work and Sect. 7 presents conclusions and future work.

## 2 Preliminaries

In this section we give formal definitions of the behaviour of agents in the MAS and of conditional norms.

We assume a finite propositional language  $L$  that contains propositions corresponding to properties of states of the MAS. A *state* of the MAS is a propositional assignment. A conjunction of all literals (propositions or their negations) in a state  $s$  will be referred to as a *state description* of  $s$ . For example, for  $L = \{p, q, r\}$ , a possible state description is  $p \wedge \neg q \wedge r$  (a state where  $p$  is true,  $q$  is false, and  $r$  is true).

A propositional formula is a boolean combination of propositional variables. The definition of a propositional formula  $\phi$  being true in a state  $s$  ( $s \models \phi$ ) is the standard classical one. We use  $\top$  for a formula that is true in all states and  $\perp$  for the formula which is false in all states.

A *trace* is a finite sequence of states. We use the notation  $\rho = (s_1, \dots, s_k)$  for a trace consisting of states  $s_1, \dots, s_k$ . For example, a trace could be generated by the actions of all vehicles involved in a traffic accident. We denote the  $i$ -th state in a trace  $\rho$  by  $\rho[i]$ . We assume that the behaviour exhibited by the agents in the MAS is represented by a set of finite traces  $\Gamma$ . We denote by  $S(\Gamma)$  or simply by  $S$  the set of states occurring in traces in  $\Gamma$ . Each subset  $X$  of  $S(\Gamma)$  is definable by a propositional formula  $\phi_X$  (a disjunction of state descriptions of states in  $X$ ). Note that the size of  $\phi_X$  is linear in the size of  $X$  (the sum of sizes of state descriptions of states in  $X$ ). For example, if  $X = \{s_1, s_2\}$  where  $s_1 = p \wedge q \wedge r$  and  $s_2 = \neg p \wedge \neg q \wedge \neg r$ , the description of  $X$  is  $(p \wedge q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$ .  $\Gamma$  is partitioned into two sets  $\Gamma_T$  ('good', or positive, traces) and  $\Gamma_F$  ('bad', or negative, traces). The partition is performed with respect to the system objective, which typically does not correspond directly to the properties expressible in  $L$ . We note that the assumption that each trace describing the behavior of the agents can be labeled as either good or bad is realistic in several contexts and for different kinds of MAS objectives. For example, instances of a process can be deemed as compliant or non-compliant w.r.t. a model [24]; in the traffic domain, traces can be labeled individually w.r.t. the expected travel time or emissions, or based on the occurrence of a collision.

The problem we wish to solve is how to generate a conditional norm which is expressed using propositions from  $L$ , and that is obeyed on traces in  $\Gamma_T$  and violated on traces in  $\Gamma_F$ .

**Definition 1 (Conditional Norm).** A conditional norm (over  $L$ ) is a tuple  $(\phi_C, Z(\phi_Z), \phi_D)$ , where  $\phi_C$ ,  $\phi_Z$  and  $\phi_D$  are propositional formulas over  $L$ , and  $Z \in \{P, O\}$  indicates whether the norm is a prohibition ( $P$ ) or an obligation ( $O$ ).

We refer to  $\phi_C$  as the (detachment) condition of the norm, and  $\phi_D$  as the deadline.  $\phi_Z$  characterizes a state that is prohibited (resp. obligated) to occur after a state where the condition of the norm  $\phi_C$  holds, and before a state where the deadline of the norm  $\phi_D$  holds. We define the conditions for violation of norms formally below.

**Definition 2 (Violation of Prohibition).** A conditional prohibition  $(\phi_C, P(\phi_P), \phi_D)$  is violated on a trace  $(s_1, s_2, \dots, s_m)$  if there are  $i, j$  with  $1 \leq i \leq j \leq m$  such that  $\phi_C$  is true at  $s_i$ ,  $\phi_P$  is true at  $s_j$ , and there is no  $k$  with  $i < k < j$  such that  $\phi_D$  is true at  $s_k$ .

In other words, a conditional prohibition is violated on a trace if the states in the trace exhibit a pattern of the following type: a state where the norm is detached (orange in Fig. 1) is followed by a number of states (possibly none) where neither the prohibition is violated nor the deadline is reached (the yellow states), after which there is a state where the deadline is still not reached but the prohibition is violated (the blue state). Note that the state where the prohibition is violated may be the same state where the norm is detached (not shown in Fig. 1, which considers the case where the three types of states are distinct).

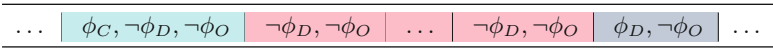
**Definition 3 (Violation of Obligation).** A conditional obligation  $(\phi_C, O(\phi_O), \phi_D)$  is violated on a trace  $(s_1, s_2, \dots, s_m)$  if there are  $i, j$  with  $1 \leq i \leq j \leq m$  such that  $\phi_C$



**Fig. 1.** Example of violation of a prohibition (Color figure online)

is true at  $s_i$ ,  $\phi_D$  is true at  $s_m$ , and there is no  $k$  with  $i \leq k \leq j$  such that  $\phi_O$  is true at  $s_k$ .

In other words, a conditional obligation is violated on a trace if the states in the trace exhibit a pattern of the following type: a state where the norm is detached (light blue in Fig. 2) is followed by a number of states (possibly none) where neither the obligation is satisfied nor the deadline is reached (the pink states), after which there is a state where the obligation is still not satisfied but the deadline is reached (the gray state). Note that, as in the case of conditional prohibitions, the state where the obligation is violated (the gray state) may be the same state where the norm is detached (not shown in Fig. 2, which considers the case where the three types of states are distinct).



**Fig. 2.** Example of violation of an obligation (Color figure online)

Note that the violation of a conditional prohibition or obligation does not distinguish between a single or multiple violations, i.e., a trace violates a norm if at least one violation occurs.

A conditional norm is obeyed on a trace if it is not violated on that trace. Violation conditions of conditional norms can be expressed in Linear Time Temporal Logic (LTL) and evaluated on finite traces in linear time [4].

*Example 1.* Consider the following simple example. Let  $L = \{p, q, r\}$  be a language where  $p$  means that a vehicle is on a particular stretch of a street,  $q$  means that it is a large goods vehicle, and  $r$  means that its speed exceeds 15 mph. The  $p$  stretch is a steep incline with a blind corner, and heavy vehicles sometimes crash into a barrier at the bottom of the street. The system objective is that such crashes are avoided. An example set of positive and negative traces is given below.

$$\begin{aligned} \Gamma_T = \{ & \rho_1 = (s_1 = p \wedge q \wedge \neg r, s_1 = p \wedge q \wedge \neg r, s_2 = \neg p \wedge q \wedge \neg r), \\ & \rho_2 = (s_3 = \neg p \wedge \neg q \wedge \neg r, s_4 = p \wedge \neg q \wedge r, s_4 = p \wedge \neg q \wedge r), \\ & \rho_3 = (s_5 = \neg p \wedge q \wedge r, s_5 = \neg p \wedge q \wedge r) \} \end{aligned}$$

$$\begin{aligned} \Gamma_F = \{ & \rho_4 = (s_1 = p \wedge q \wedge \neg r, s_6 = p \wedge q \wedge r, s_2 = \neg p \wedge q \wedge \neg r), \\ & \rho_5 = (s_1 = p \wedge q \wedge \neg r, s_6 = p \wedge q \wedge r) \} \end{aligned}$$

Intuitively, positive traces involve only slowly driving trucks when  $p$  is true, and arbitrary speeds otherwise. The following conditional prohibition is violated on all negative traces:  $(p \wedge q, P(p \wedge q \wedge r), \neg p)$ .<sup>1</sup>

### 3 Complexity of Norm Synthesis

Given a set of agent behaviour traces  $\Gamma$  partitioned into  $\Gamma_T$  and  $\Gamma_F$ , we wish to synthesize a norm that correctly classifies each trace (that is, the norm is violated on all traces in  $\Gamma_F$ , and is not violated on any trace in  $\Gamma_T$ ). Clearly, this is not always possible; two sets of traces may not be distinguishable by a single conditional norm (or even by a set of conditional norms). For example:

$$\Gamma_T = \{(s_1, s_2, s_3)\}, \Gamma_F = \{(s_1, s_1, s_2, s_3)\}$$

cannot be distinguished by a conditional norm.

#### 3.1 Prohibition Synthesis

We first define formally the decision problem we call *prohibition synthesis*.

**Definition 4.** *The prohibition synthesis problem is the following decision problem:*

**Instance** *A finite set of propositions  $L$ ; a finite set of finite traces  $\Gamma$  partitioned into  $\Gamma_T$  and  $\Gamma_F$ , each trace given as a sequence of state descriptions over  $L$ .*

**Question** *Are there three propositional formulas  $\phi_C$ ,  $\phi_P$ , and  $\phi_D$  over  $L$  such that*

**Neg** *every trace in  $\Gamma_F$  violates  $(\phi_C, P(\phi_P), \phi_D)$*

**Pos** *no trace in  $\Gamma_T$  violates  $(\phi_C, P(\phi_P), \phi_D)$*

The correspondence between sets of states and formulas over  $L$  allows us to restate the prohibition synthesis problem as follows: given a set of positive traces  $\Gamma_T$  and negative traces  $\Gamma_F$ , find three sets of states  $X_C, X_P, X_D$  such that:

**Neg** For every trace  $\rho \in \Gamma_F$ , there exists  $i$  and  $j$  with  $i \leq j$  such that  $\rho[i] \in X_C$ ,  $\rho[j] \in X_P$ , and there is no  $k$  with  $i < k < j$  such that  $\rho[k] \in X_D$ .

**Pos** For every trace  $\rho \in \Gamma_T$ , if for some  $i$  and  $j$ ,  $i \leq j$ ,  $\rho[i] \in X_C$ ,  $\rho[j] \in X_P$ , then there exists  $k$  such that  $i < k < j$  and  $\rho[k] \in X_D$ .

**Theorem 1.** *The prohibition synthesis problem is NP-complete.*

*Proof.* The prohibition synthesis problem is clearly in NP (a non-deterministic Turing machine can guess the three sets and check in polynomial time that they satisfy the conditions). To prove that it is NP-hard, we reduce 3SAT (satisfiability of a set of clauses with 3 literals) to prohibition synthesis.

<sup>1</sup> Clearly, alternative definitions of norms are also possible. For example, since trucks do not cease being trucks while driving along the street, we can also state the prohibition as  $(p \wedge q, P(r), \neg p)$ , or we can prohibit a truck driving fast on  $p$ :  $(\top, P(p \wedge q \wedge r), \perp)$ .

3SAT is an NP-complete problem. An instance of 3SAT is a set of clauses, where each clause is a disjunction of at most 3 literals, for example,  $\{(x_1 \vee x_2 \vee \neg x_3), (\neg x_1 \vee \neg x_2 \vee x_4)\}$ . The question is whether the set of clauses is satisfiable, that is, whether there is an assignment of truth values 0 and 1 to the propositional variables that makes all the clauses true; in other words, is there an assignment such that each clause contains at least one true literal. In the example above, assigning 0 to  $x_3$  and 1 to  $x_4$ , and, for example, 0 to  $x_1$  and to  $x_2$ , makes both clauses true.

To start the reduction from 3SAT to prohibition synthesis, suppose an instance of 3SAT is given; that is, we have a set of clauses  $C_1, \dots, C_n$  over variables  $x_1, \dots, x_m$ . We generate an instance of the prohibition synthesis problem such that it has a solution if, and only if,  $C_1, \dots, C_n$  are satisfiable (each clause contains at least one true literal). We construct the corresponding instance of the prohibition synthesis problem as follows. The set of states in the prohibition synthesis problem consists of two states  $s$  and  $t$  ( $s$  and  $t$  are a technical device; intuitively they serve as the detachment condition and the violation of the prohibition), and for each variable  $x_i$ , we need two states  $u_i$  and  $v_i$ . When we ‘translate’ a clause into a trace, we insert  $u_i$  into the trace if  $x_i$  occurs positively in the clause, and  $v_i$  if it occurs negatively. Intuitively,  $u_i$  in  $X_D$  will be a proxy for ‘ $x_i$  should be assigned 1’, and  $v_i$  in  $X_D$  will be a proxy for ‘ $x_i$  should be assigned 0’. We give the rest of the construction below. Comments in square brackets explain the intuition for each step in the construction.

The set of negative traces  $\Gamma_F$  contains:

- a two state trace  $(s, t)$  [together with  $s, t \notin X_C \cap X_P$  below, this forces  $s \in X_C$  and  $t \in X_P$ ];
- for every variable  $x_i$  in the input, a trace  $(s, v_i, t, s, u_i, t)$  [this ensures that either  $v_i$  or  $u_i$  are not in  $X_D$ ].

The set of positive traces  $\Gamma_T$  contains:

- a single state trace  $(s)$  [so  $s$  cannot be in  $X_C \cap X_P$ ];
- $(t)$  [so  $t$  cannot be in  $X_C \cap X_P$ ];
- for every variable  $x_i$  in the input:  $(s, v_i, u_i, t)$  [this means that either  $v_i$  or  $u_i$  are in  $X_D$ ];  $(v_i)$ ;  $(u_i)$ ;  $(v_i, t)$ ;  $(u_i, t)$ ;  $(s, v_i)$ ;  $(s, u_i)$ ;
- for every pair of variables  $x_i, x_j$  in the input:  $(v_i, u_j)$ ;  $(u_j, v_i)$  [this together with preceding traces ensures that  $v_i$  and  $u_i$  are not in  $X_C$  or  $X_P$ ];
- for each clause  $C$  in the input over variables  $x_j, x_k, x_l$ :  $(s, z_j, z_k, z_l, t)$  where  $z_i$  is  $u_i$  if  $x_i$  occurs in  $C$  positively, and  $v_i$  if it occurs negatively.

It is easy to see that the reduction from the 3SAT instance to the prohibition synthesis instance is polynomial in the number  $m$  of variables (quadratic) and in the number  $n$  of clauses (linear).

We claim that there exists an assignment  $f$  of truth values 0, 1 to  $x_1, \dots, x_m$  such that all the clauses  $C_1, \dots, C_n$  are true if, and only if, there is a solution to the prohibition synthesis problem above, where  $X_C = \{s\}$ ,  $X_P = \{t\}$ , and for every  $i$ ,  $u_i \in X_D$  iff  $f(x_i) = 1$  and  $v_i \in X_D$  iff  $f(x_i) = 0$ .

**‘only if’ direction:** Assume that an assignment  $f$  that makes  $C_1, \dots, C_n$  true exists. Let  $X_C = \{s\}$  and  $X_P = \{t\}$ . For every  $i$ , place  $u_i$  in  $X_D$  if  $f(x_i) = 1$  and  $v_i \in X_d$  if

$f(x_i) = 0$ . This produces a solution to the prohibition synthesis problem because:  $s, t$  satisfies **Neg**; for every  $i$ , either  $u_i$  or  $v_i$  are not in  $X_D$ , so  $s, v_i, t, s, u_i, t$  satisfies **Neg**. Positive traces satisfy **Pos**: either  $s$  followed by  $t$  does not occur on a trace, or  $u_i, v_i$  occur between  $s$  and  $t$  and one of them is in  $X_D$ , or (from the clause encoding) one of the literals in the clause is true, so for positive  $x_i$  it means that  $u_i$  is in  $X_D$  and **Pos** is satisfied, or for negative  $\neg x_i$  it means that  $v_i$  is in  $X_D$  and again **Pos** is satisfied.

**‘if’ direction:** Assume there is a solution to the prohibition synthesis problem. It is clear (see the comments in square brackets above) that it has to be of the form  $X_C = \{s\}$ ,  $X_P = \{t\}$  and  $X_D$  containing some  $u_i$ s and  $v_i$ s. In particular, since  $(s, v_i, u_i, t)$  is a positive trace, for every  $i$  either  $u_i$  or  $v_i$  must not be in  $X_D$ . Set  $f(x_i)$  to be 1 if  $u_i$  is in  $X_D$  and 0 otherwise. Then each clause  $C = \{\sim x_j, \sim x_k, \sim x_l\}$  (where  $\sim x_j$  denotes  $x_j$  if it occurs positively or  $\neg x_j$  if it occurs negatively) is satisfied by  $f$  since for every clause there will be one literal which is true. This is because  $(s, z_j, z_k, z_l, t)$  is a positive trace, and either for some positive literal  $x_i, u_i$  is in  $X_D$ , or for some negative literal  $\neg x_i, v_i$  is in  $X_D$ , so  $u_i$  is not in  $X_D$ , so  $f(\neg x_i) = 1$ .

### 3.2 Obligation Synthesis

We now consider the *obligation synthesis problem*.

**Definition 5.** *The obligation synthesis problem is the following decision problem:*

**Instance** *A finite set of propositions  $L$ , a finite set  $\Gamma$  of finite traces partitioned into  $\Gamma_T$  and  $\Gamma_F$ , where each trace is given as a sequence of state descriptions.*

**Question** *Are there three propositional formulas  $\phi_C, \phi_O$ , and  $\phi_D$  over  $L$  such that*

**Neg** *every trace in  $\Gamma_F$  violates  $(\phi_C, O(\phi_O), \phi_D)$*

**Pos** *no trace in  $\Gamma_T$  violates  $(\phi_C, O(\phi_O), \phi_D)$*

Analogously to the prohibition synthesis problem, the obligation synthesis problem can be equivalently restated in terms of states: are there three sets of states  $X_C, X_O$  and  $X_D$  such that:

**Neg** For every trace  $\rho \in \Gamma_F$ , there exist  $i$  and  $j$  with  $i \leq j$  such that  $\rho[i] \in X_C$ ,  $\rho[j] \in X_D$ , and there is no  $k$  with  $i \leq k \leq j$  such that  $\rho[k] \in X_O$

**Pos** For every trace  $\rho \in \Gamma_T$ , if for some  $i$  and  $j$ ,  $i \leq j$ ,  $\rho[i] \in X_C$ ,  $\rho[j] \in X_D$ , then there exists  $k$  such that  $i \leq k \leq j$  and  $\rho[k] \in X_O$ .

**Theorem 2.** *The obligation synthesis problem is NP-complete.*

*Proof.* The obligation synthesis problem is clearly in NP. To prove that it is NP-hard, we again use a reduction from the 3SAT problem.

As before, consider a set of clauses  $C_1, \dots, C_n$  over variables  $x_1, \dots, x_m$ , which is an instance of 3SAT. We generate an instance of the obligation synthesis problem such that it has a solution iff  $C_1, \dots, C_n$  are satisfiable. The idea of the reduction is similar to that for prohibitions. We use two auxiliary states  $s$  and  $t$ , intuitively to serve as the detachment condition and the deadline, and make sure that neither of them is also the obligation, but now instead of inserting a deadline between  $s$  and  $t$  in positive traces, we insert an obligation. We want to make some subset of  $\{v_i : i \in [1, \dots, m]\} \cup \{u_i :$

$i \in [1, \dots, m]$  to be the obligation ( $X_O$ ), so that exactly one of  $v_i, u_i$  for each  $i$  is in  $X_O$ . Then  $u_i \in X_O$  can encode that  $x_i$  is true, and  $v_i \in X_O$  that  $x_i$  is false, and we can make the encoding work by creating a positive trace corresponding to each clause so that at least one of the literals in the clause should be true.

The set of negative traces contains:

- a 2 state trace  $(s, t)$  [this forces either  $s \in X_C \cap \overline{X_D} \cap \overline{X_O}$ ,  $t \in X_D \cap \overline{X_C} \cap \overline{X_O}$ , or  $s \in X_C \cap X_D \cap \overline{X_O}$ , or  $t \in X_C \cap X_D \cap \overline{X_O}$ . To rule out the latter two possibilities, we require below that  $s$  and  $t$  on their own are positive traces.]
- for every variable  $x_i$  in the input, a trace  $(s, v_i, t, s, u_i, t)$  [this ensures that either  $v_i$  or  $u_i$  are not in  $X_O$ , because there is one  $(s, \dots, t)$  sub-trace that does not contain a state from  $X_O$ ].

The set of positive traces contains:

- a one state trace  $(s)$  [so  $s$  cannot be in  $X_C \cap X_D \cap \overline{X_O}$ ]
- a one state trace  $(t)$  [so  $t$  cannot be in  $X_C \cap X_D \cap \overline{X_O}$ ]
- for every variable  $x_i$  in the input, a trace  $(s, v_i, u_i, t)$  [this ensures that either  $v_i$  or  $u_i$  are in  $X_O$ ]
- for each clause  $C$  in the input over variables  $x_j, x_k, x_l$ , a trace  $(s, z_j, z_k, z_l, t)$  where  $z_i$  is  $u_i$  if  $x_i$  occurs in  $C$  positively, and  $v_i$  if it occurs negatively.

The reduction is linear in the number of variables and clauses.

We claim that there exists an assignment  $f$  of 0, 1 to  $x_1, \dots, x_m$  satisfying  $C_1, \dots, C_n$  if, and only if, there is a solution to the obligation synthesis problem above where  $s \in X_C$ ,  $t \in X_D$ , and for every  $i$ ,  $u_i \in X_O$  iff  $f(x_i) = 1$  and  $v_i \in X_O$  iff  $f(x_i) = 0$ . The proof of this claim is analogous to that of Theorem 1.

Assume that an assignment  $f$  satisfying  $C_1, \dots, C_n$  exists. Let  $X_C = \{s\}$  and  $X_D = \{t\}$ . For every  $i$ , place  $u_i$  in  $X_O$  iff  $f(x_i) = 1$  and  $v_i \in X_O$  iff  $f(x_i) = 0$ . It is easy to check that this is a solution to the obligation synthesis problem.

Assume there is a solution to the obligation synthesis problem. It is clear (see the comments in brackets above) that any solution should satisfy  $s \in X_C \cap \overline{X_D} \cap \overline{X_O}$  and  $t \in X_D \cap \overline{X_C} \cap \overline{X_O}$ . Since  $(s, v_i, t, s, u_i, t)$  is a negative trace for every  $i$ , this means that it contains an unsatisfied conditional obligation. This means that for every  $i$ , either  $v_i$  or  $u_i$  is not in  $X_O$ . Since  $(s, v_i, u_i, t)$  is a positive trace, then in any solution, for every  $i$ , either  $u_i$  or  $v_i$  has to be in  $X_O$ . Hence we can use the membership in  $X_O$  to produce a boolean valuation of variables  $x_i$  (1 if  $u_i \in X_O$ , and 0 if  $v_i \in X_O$ ). Since for every clause  $C = \{\sim x_j, \sim x_k, \sim x_l\}$ , the trace  $(s, z_j, z_k, z_l, t)$  (where  $z_i$  is  $v_i$  if  $\sim x_i = \neg x_i$ , and  $u_i$  if  $\sim x_i = x_i$ ) is a positive trace, at least one of  $z_i$  is in  $X_O$ . This means that the valuation based on the membership in  $X_O$  satisfies all the clauses (since at least one literal in each clause will evaluate to 1).

## 4 Complexity of Synthesising a Set of Norms

In this section, we consider the problem of synthesising a set of norms. To motivate the problem, we first give an example where classifying positive and negative traces correctly requires more than one norm.



*Example 2.* Let the language  $L$  be  $\{p, q_1, q_2, r_1, r_2\}$  where, for the sake of intuition,  $p$  denotes a particular kind of customer who needs to be greeted in a particular way ( $r_1$ ) before they pass the greeter ( $q_1$ ) and  $\neg p$  is all other customers who need to be greeted in a different way ( $r_2$ ), before  $q_2$ .

$$\begin{aligned} \Gamma_T = \{ & \rho_1 = (s_1 = p \wedge \neg q_1 \wedge \neg q_2 \wedge \neg r_1 \wedge \neg r_2, s_2 = p \wedge \neg q_1 \wedge \neg q_2 \wedge r_1 \wedge \neg r_2, \\ & s_3 = \neg p \wedge q_1 \wedge \neg q_2 \wedge \neg r_1 \wedge \neg r_2, s_4 = \neg p \wedge \neg q_1 \wedge \neg q_2 \wedge \neg r_1 \wedge r_2) \} \\ \Gamma_F = \{ & \rho_2 = (s_1 = p \wedge \neg q_1 \wedge \neg q_2 \wedge \neg r_1 \wedge \neg r_2), \\ & \rho_3 = (s_3 = \neg p \wedge q_1 \wedge \neg q_2 \wedge \neg r_1 \wedge \neg r_2) \} \end{aligned}$$

This example can only be solved by two norms, because a trace consisting only of state  $s_1$  is a violation ( $\rho_2$ ), but  $s_1$  alone cannot be prohibited because the trace  $\rho_1 = (s_1, s_2, s_3, s_4)$  is in  $\Gamma_T$ . So, the trace  $(s_1)$  must be ruled out by an obligation: after  $s_1$ , there should be  $s_2$  or  $s_3$  or  $s_4$ . From trace  $\rho_4$  in  $\Gamma_F$ ,  $(s_3)$  is a violation, so  $s_3$  must either be prohibited, or it must be ruled out by an obligation, that is, after  $s_3$ ,  $s_4$  should happen. If  $s_3$  is prohibited, then  $\rho_1 = (s_1, s_2, s_3, s_4)$  would be a violation, but it isn't. So after  $s_3$ ,  $s_4$  should happen. Therefore two obligations are required, for example,  $(p, O(r_1), q_1)$  and  $(\neg p, O(r_2), q_2)$ .

Similarly, two or more prohibitions may be required if different things are prohibited in different contexts.

If there is a set of norms separating  $\Gamma_T$  and  $\Gamma_F$ , then its size is trivially bounded by the number of all different non-equivalent norms given the language  $L$ . Since  $L$  is finite, there are  $2 \times 3 \times 2^{2^{|L|}}$  possible conditional norms (there are  $2^{|L|}$  state descriptions,  $2^{2^{|L|}}$  possible formulas in disjunctive normal form that can be parts of the norm, 3 positions on which they can occur, and 2 types of conditional norms). There are  $O(2^{2^{2^{|L|}}})$  sets of non-equivalent norms. However, it is possible to produce a much better bound on the maximal size of the set of norms correctly classifying  $\Gamma_T$  and  $\Gamma_F$  than a triple exponential in  $|L|$ .

**Theorem 3.** *If it is possible to correctly classify  $\Gamma_T$  and  $\Gamma_F$  by a set  $N$  of norms, then this can be done by a set of norms of size at most  $|\Gamma_F|$ .*

*Proof.* First, observe that we do not need more than one norm to exclude each trace in  $\Gamma_F$ . So we need to have at most  $|\Gamma_F|$  norms. Second, if a set  $N$  of norms is not violated on any of  $\Gamma_T$  traces, then no norm from  $N' \subseteq N$  is violated on a  $\Gamma_T$  trace.

**Definition 6.** *The multiple conditional norm synthesis problem is the following decision problem:*

**Instance** *A finite set of propositions  $L$ ; an integer  $m$ ; a finite set of finite traces  $\Gamma$  partitioned into  $\Gamma_T$  and  $\Gamma_F$ , each trace given as a sequence of state descriptions over  $L$ .*

**Question** *is there a set  $N$  of conditional prohibitions and obligations over  $L$  with  $|N| \leq m$  such that*

**Neg** *every trace in  $\Gamma_F$  violates one of the norms in  $N$*

**Pos** no trace in  $\Gamma_T$  violates any of the norms in  $N$ .

**Theorem 4.** *The problem of synthesising a set of conditional prohibitions or conditional obligations is NP-complete.*

*Proof.* For membership in NP, observe that it is possible to guess a set  $m \leq |\Gamma_F|$  norms and check in polynomial time that they correctly classify the traces.

Hardness follows from the NP-hardness parts of Theorems 1 and 2.

## 5 Complexity of Minimal Revision

In this section, we consider the problems of (minimally) revising conditional prohibitions and obligations. These problems are relevant when there is an existing norm that does not achieve the system objective, and we wish a minimal modification of the existing norm that does achieve the objective.

Assume we are given a set of traces and a conditional norm  $(\phi_C, Z(\phi_Z), \phi_D)$ , (where  $Z \in \{P, O\}$ ) and need to change it in a minimal way so that it classifies the traces correctly. The editing distance between conditional norms can be defined in various ways, e.g., for formulas  $\phi_C, \phi_Z, \phi_D$  in disjunctive normal form, this could be the sum of the numbers of added and removed disjuncts for all three formulas. Note that the set of non-equivalent propositional formulas built from the set  $L$  is finite, and so is the number of possible different conditional prohibitions or obligations. Regardless of how the distance between different conditional norms is defined, for a fixed set of propositional variables  $L$  there is a maximal editing distance  $\max(L)$  between any two norms using formulas over  $L$ .

### 5.1 Complexity of Minimal Prohibition Revision

Given some distance measure  $dist$  defined for any two conditional prohibitions  $\alpha_1$  and  $\alpha_2$  over  $L$ , the decision problem for minimal prohibition revision can be stated as:

**Definition 7.** *The (decision form) of the minimal prohibition revision problem is as follows:*

**Instance** *A finite set of propositions  $L$ ; a number  $m$ ; a finite set  $\Gamma$  of finite traces partitioned into  $\Gamma_T$  and  $\Gamma_F$ ; a conditional prohibition  $(\phi_C, P(\phi_P), \phi_D)$  over  $L$ .*

**Question** *Are there three propositional formulas  $\phi'_C, \phi'_P,$  and  $\phi'_D$  over  $L$  such that*

**Dist**  $dist((\phi_C, P(\phi_P), \phi_D), (\phi'_C, P(\phi'_P), \phi'_D)) \leq m$

**Neg** every trace in  $\Gamma_F$  violates  $(\phi'_C, P(\phi'_P), \phi'_D)$

**Pos** no trace in  $\Gamma_T$  violates  $(\phi'_C, P(\phi'_P), \phi'_D)$

**Theorem 5.** *Let  $dist(\alpha_1, \alpha_2)$  be computable in time polynomial in the size of  $\alpha_1$  and  $\alpha_2$ , and the range of  $dist$  over norms built over propositions from  $L$  be bounded by  $\max(L)$ . Then the minimal prohibition revision problem is NP-complete.*

*Proof.* The membership in NP follows from the fact that a solution can be guessed and checked in polynomial time.

NP-hardness is by reduction from the prohibition synthesis problem. Note that if a solution to the prohibition synthesis problem exists, it will be at most at distance  $\max(L)$  from the input norm. So to solve the prohibition synthesis problem, we can ask for a solution to the minimal prohibition revision problem with  $m = \max(L)$ .

## 5.2 Complexity of Minimal Obligation Revision

Given some distance measure  $dist$  defined for any two conditional obligations  $\alpha_1$  and  $\alpha_2$  over  $L$ , the decision problem for minimal obligation revision can be stated as:

**Definition 8.** *The (decision form) of the minimal obligation revision problem is as follows:*

**Instance** *A finite set of propositions  $L$ ; a number  $m$ ; a finite set  $\Gamma$  of finite traces partitioned into  $\Gamma_T$  and  $\Gamma_F$ ; a conditional obligation  $(\phi_C, O(\phi_O), \phi_D)$  over  $L$ .*

**Question** *Are there three propositional formulas  $\phi'_C$ ,  $\phi'_O$ , and  $\phi'_D$  over  $L$  such that*

**Dist**  *$dist((\phi_C, O(\phi_O), \phi_D), (\phi'_C, O(\phi'_O), \phi'_D)) \leq m$*

**Neg** *every trace in  $\Gamma_F$  violates  $(\phi'_C, O(\phi'_O), \phi'_D)$*

**Pos** *no trace in  $\Gamma_T$  violates  $(\phi'_C, O(\phi'_O), \phi'_D)$*

**Theorem 6.** *Let  $dist(\alpha_1, \alpha_2)$  be computable in time polynomial in the size of  $\alpha_1$  and  $\alpha_2$ , and the range of  $dist$  over norms built over propositions from  $L$  be bounded by  $max(L)$ . Then the minimal obligation revision problem is NP-complete.*

*Proof.* The membership in NP follows from the fact that a solution can be guessed and checked in polynomial time. Analogously to the minimal prohibition revision problem, NP-hardness is by reduction from the obligation synthesis problem; if a solution to the obligation synthesis problem exists, it will be at most at distance  $max(L)$  from the input norm. So to solve the obligation synthesis problem, we can ask for a solution to the minimal obligation revision problem with  $m = max(L)$ .

## 6 Related Work

There has been a considerable amount of work on the automated synthesis of norms. In this section, we briefly review some of the main approaches, focussing on work that is most closely related to our approach.

We first review ‘offline’ approaches, in which norms are synthesised at design time. Shoham and Tennenholtz [34] (see also [16]), consider the problem of synthesising a *social law* that constrains the behaviour of the agents in a MAS so as to ensure that agents in a *focal* state are always able to reach another focal state no matter what the other agents in the system do. They show that synthesising a useful social law is NP-complete. Van der Hoek *et al.* [18] recast the problem of synthesising a social law as an ATL model checking problem. The authors show that the problem of whether there exists a social law satisfying an objective expressed as an arbitrary ATL formula (feasibility) is NP-complete, while for objectives expressed as propositional formulae, feasibility (and synthesis) is decidable in polynomial time. Bulling and Dastani [10] consider norm synthesis for LTL objectives. In their approach, agents are assumed to have LTL-defined preferences with numerical values, and the aim of the synthesis is to produce a norm that enforces the objective for some Nash equilibrium. The problems they consider are weak and strong implementation, and norm-based mechanism design. A norm weakly implements a normative behaviour function if there exists a Nash equilibrium that satisfies the LTL formula. A norm strongly implements a normative behaviour function iff all

Nash equilibria satisfy the formula. Weak implementation is  $\Sigma_2^P$ -complete in the size of the CGS, preferences, objective and norm. The strong implementation problem can be solved by a deterministic polynomial-time oracle Turing machine that can make two non-adaptive queries to an oracle in  $\Sigma_2^P$  and is both  $\Sigma_2^P$ -hard and  $\Pi_2^P$ -hard. Weak implementation existence is  $\Sigma_2^P$ -complete. Strong implementation existence is  $\Sigma_3^P$ -complete. In [19], the synthesis of dynamic prohibitions (that is, prohibitions corresponding to Mealy machines) for CTL objectives is shown to be EXPTIME-complete. In [32], the synthesis of dynamic norms for LTL objectives and Nash equilibria is shown to be 2EXPTIME-complete when considering the existence of a Nash equilibrium satisfying the objective, and in 3EXPTIME for enforcing all Nash equilibria to satisfy the objective. Other work on norm synthesis using logical specifications of objectives includes [2, 36]. Alechina *et al.* [4] introduce the concept of norm approximation in the context of imperfect monitors. A conditional norm is synthesized to approximate an ‘ideal’ norm in order to maximize the number of violations that an imperfect monitor can detect. We assume, however, perfectly monitorable norms, and we aim at synthesizing norms that are better aligned with the MAS objectives by using execution data. In contrast to the approach we present here, these approaches assume a complete model the agents’ behaviour is available, e.g., in the form of a transition system or a Kripke structure.

Morales *et al.* present LION [28], an algorithm for the synthesis of liberal normative systems, i.e., norms that place as few constraints as possible on the actions of agents. To guide the synthesis process, LION makes use of a normative network: a graph structure that characterises the generalisation relationship between different norms, which is used to synthesise more general, that is, more liberal, norms when possible. The norms synthesised by LION are so-called action-based norms, which prohibit agents to perform actions in certain states [5]. In our work, we focus on the problem of revising conditional norms with deadlines, which are behaviour-based, or path-based, norms, prohibiting (or obliging) agents from exhibiting certain behaviours. While both our work and LION synthesise norms to avoid undesirable system states, in our work we focus on the problem of synthesising norms from data collected during the execution of the system (i.e., traces of agent behaviour), while in [28], the synthesis considers properties of the normative system (e.g., liberality) which are independent of the behaviour of the agents in the MAS. We consider the liberality of norms an interesting possible extension of our work that could be integrated as a criterion when selecting a new norm among possible revisions. Christelis *et al.* [12] present an EXPTIME algorithm based on AI planning for synthesising state-based prohibitions that set preconditions to the actions the agents can perform in a regimentation setting. In our work, we do not assume that norms can be regimented.

Another strand of work focuses on the ‘online’ synthesis of norms, where norms emerge from the interactions of agents in a decentralised way, e.g., [3, 33]. Unlike our approach, such approaches typically assume that the agents are cooperative, and/or that some minimal standards of behaviour can be assumed. However, cooperation between agents cannot be always assumed, particularly in open MAS.

Closer to our work are online approaches that use agents’ behaviour to guide centralised norm synthesis. For example, Morales *et al.* [27] present algorithms for the online synthesis of compact *action-based* norms when the behaviour of agents leads to

undesired system states. In contrast, we consider conditional norms with deadlines that regulate patterns of behaviour. In other work, Morales *et al.* have used game theoretic concepts to guide norm synthesis [29,30]. Their control loop includes game recognition, payoff learning, and norm replication. Their approach to norm synthesis makes use of evolutionary processes to determine, off-line and via simulation, effective and evolutionary stable norms, which are then enforced at run-time. However, while the resulting norms are evolutionary stable, their approach requires sufficient knowledge about the agents, their goals and the environment in which they operate, to permit simulation of their interactions. In our work, instead, we focus on a setting where the only labeled traces of agent behaviors are available. Miralles *et al.* [26] present a framework for the adaptation of MAS regulations at runtime. They consider norms expressed via norm patterns (i.e., IF-THEN rules associated with constraints on the operators and on the values that the norm components can take). The authors describe an adaptation mechanism based on Case Based Reasoning. Adaptation is performed at runtime individually by a number of assistant agents and then, via a voting mechanism, a final adaptation is approved. The decision on how to adapt norms is taken based on similar previously seen cases. On similar lines, Dell'Anna *et al.* [14] propose a framework for the runtime selection of alternative norms based on runtime data and for the revision of the sanctions of norms based on the knowledge of agents preferences. Unlike these approaches, we do not assume knowledge of the agents' internals, e.g., their preferences [14] or their reasoning and communication capabilities [26]. Corapi *et al.* [13] and Athakravi *et al.* [7] discuss the application of Inductive Logic Programming (ILP) [15] to norm synthesis and norm revision. In their work, the desired properties of the system are described through use cases (event traces associated to a desired outcome state), and ILP is used to revise the current norms so to satisfy the use-cases. In their approach, norms and desired outcome are strictly coupled: the desired outcomes of execution traces are expressed in the same language of the norms and, therefore, are directly enforceable. In our approach we consider MAS objectives that cannot be directly enforced, and we use norms as a means to achieve such objectives (e.g., a speed limit norm is a means to achieve vehicles' safety, but it is not possible to directly enforce safety on vehicles: "no accidents should occur" is not directly enforceable on drivers). In our work, the only knowledge of the MAS objectives available to the revision mechanism, is a given boolean labeling of the execution traces. The causal relation between norms and MAS objectives is not given. Because we do not assume that the underlying causal structure of the domain is known to our revision mechanism, we are unable to generate provably correct norm revisions as in ILP-based approaches like those of Corapi *et al.* [13] and related ones (e.g., [21,31]). ILP-based approaches and our approach can therefore be seen as representing different trade-offs between the amount of background knowledge assumed about the possible causes of norm violations, and the guarantees that can be given regarding a particular (candidate) revision. Mahmoud *et al.* [25] propose an algorithm for mining regulative norms that identifies recommendations, obligations, and prohibitions by analyzing events that trigger rewards and penalties. They focus on agents joining an open MAS who have to learn the unstated norms; we, instead, study how to alter existing norms from the point of view of a centralized authority.

Finally, our work is influenced by research on norm change, including logics for norm change [8, 22], the study of the legal effects of norm change, analyzed and formalized by [17], and the contextualization of norms [20], which studies how to refine norms to make them suitable for specific contexts. In our framework, this corresponds to modifying the detachment condition and the deadline of the norms.

## 7 Conclusions

We considered the problem of synthesising and minimally revising norms to achieve a system objective from labelled traces of agent behaviour in a multi-agent system (MAS). We considered a setting where the internals of the agents in the MAS are unknown and where norms are expressed in a different language from that of the system objective that they intend to bring about. In such setting, explicit knowledge about the relationship between the enforced norms, the agents' behavior and the MAS objective is not given, and the norm synthesis and revision rely on traces of agent behaviour labeled as positive or negative, depending on whether each satisfies or not the system objective. We showed that the problems of norm synthesis and minimal revision are NP-complete. In future work, we plan to investigate the synthesis of approximate norms (i.e., norms that do not classify all traces perfectly), and more tractable heuristic approaches to norm synthesis and revision where, for instance, only a bounded number of candidate revisions of a norm are synthesized based on the available data and the semantics of conditional norms, and the most accurate norm (i.e., the norm with highest accuracy w.r.t. the labeled traces) is selected.

## References

1. van der Aalst, W.M.P.: *Process Mining - Data Science in Action*, 2nd edn. Springer, Heidelberg (2016). <https://doi.org/10.1007/978-3-662-49851-4>
2. Ågotnes, T., van der Hoek, W., Rodríguez-Aguilar, J.A., Sierra, C., Wooldridge, M.J.: On the logic of normative systems. In: Veloso, M.M. (ed.) *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 1175–1180 (2007)
3. Airiau, S., Sen, S., Villatoro, D.: Emergence of conventions through social learning. *Auton. Agents Multi-Agent Syst.* **28**(5), 779–804 (2014)
4. Alechina, N., Dastani, M., Logan, B.: Norm approximation for imperfect monitors. In: *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2014*, pp. 117–124 (2014)
5. Alechina, N., Logan, B., Dastani, M.: Modeling norm specification and verification in multiagent systems. *FLAP* **5**(2), 457–490 (2018). <https://www.collegepublications.co.uk/downloads/ifcolog00022.pdf>
6. Artikis, A., Pitt, J.: A formal model of open agent societies. In: *Proceedings of the Fifth International Conference on Autonomous Agents, AGENTS 2001*, pp. 192–193 (2001)
7. Athakravi, D., Corapi, D., Russo, A., Vos, M.D., Padget, J.A., Satoh, K.: Handling change in normative specifications. In: *International Conference on Autonomous Agents and Multi-agent Systems, AAMAS 2012*, pp. 1369–1370 (2012)
8. Aucher, G., Grossi, D., Herzig, A., Lorini, E.: Dynamic context logic. In: *Proceedings of LORI*, pp. 15–26 (2009)

9. Boella, G., van der Torre, L.W.N.: Regulative and constitutive norms in normative multiagent systems. In: *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, Whistler, Canada, 2–5 June 2004, pp. 255–266 (2004)
10. Bulling, N., Dastani, M.: Norm-based mechanism design. *Artif. Intell.* **239**, 97–142 (2016)
11. Chopra, A., van der Torre, L., Verhagen, H., Villata, S. (eds.): *Handbook of Multiagent Systems*. College Publications, London (2018)
12. Christelis, G., Rovatsos, M.: Automated norm synthesis in an agent-based planning environment. In: *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 161–168. International Foundation for Autonomous Agents and Multiagent Systems (2009)
13. Corapi, D., Russo, A., Vos, M.D., Padget, J.A., Satoh, K.: Normative design using inductive learning. *Theory Pract. Log. Program.* **11**(4–5), 783–799 (2011)
14. Dell'Anna, D., Dastani, M., Dalpiaz, F.: Runtime revision of sanctions in normative multiagent systems. *Auton. Agents Multi-Agent Syst.* **34**(2), 1–54 (2020). <https://doi.org/10.1007/s10458-020-09465-8>
15. Dzeroski, S., Lavrac, N.: *Inductive logic programming: techniques and applications* (1994)
16. Fitoussi, D., Tennenholtz, M.: Choosing social laws for multi-agent systems: minimality and simplicity. *Artif. Intell.* **119**(1), 61–101 (2000)
17. Governatori, G., Rotolo, A.: Changing legal systems: legal abrogations and annulments in defeasible logic. *Log. J. IGPL* **18**(1), 157–194 (2010). <https://doi.org/10.1093/jigpal/jzp075>
18. van der Hoek, W., Roberts, M., Wooldridge, M.J.: Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese* **156**(1), 1–19 (2007)
19. Huang, X., Ruan, J., Chen, Q., Su, K.: Normative multiagent systems: the dynamic generalization. In: Kambhampati, S. (ed.) *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pp. 1123–1129 (2016)
20. Jiang, J., Aldewereld, H., Dignum, V., Tan, Y.: Norm contextualization. In: *Coordination, Organizations, Institutions, and Norms in Agent Systems VIII - 14th International Workshop, COIN 2012, Held Co-located with AAMAS 2012, Valencia, Spain, 5 June 2012, Revised Selected Papers*, pp. 141–157 (2012)
21. Katzouris, N., Artikis, A., Paliouras, G.: Incremental learning of event definitions with inductive logic programming. *Mach. Learn.* **100**(2), 555–585 (2015)
22. Knobbout, M., Dastani, M., Meyer, J.C.: A dynamic logic of norm change. In: *Proceedings of ECAI*, pp. 886–894 (2016)
23. Lorenz, R., Senoner, J., Sihm, W., Netland, T.: Using process mining to improve productivity in make-to-stock manufacturing. *Int. J. Prod. Res.*, 1–12 (2021)
24. Loreti, D., Chesani, F., Ciampolini, A., Mello, P.: Generating synthetic positive and negative business process traces through abduction. *Knowl. Inf. Syst.* **62**(2), 813–839 (2020)
25. Mahmoud, M.A., Ahmad, M.S., Yusoff, M.Z.M., Mostafa, S.A.: A regulative norms mining algorithm for complex adaptive system. In: Ghazali, R., Deris, M.M., Nawi, N.M., Abawajy, J.H. (eds.) *SCDM 2018. AISC*, vol. 700, pp. 213–224. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-72550-5\\_21](https://doi.org/10.1007/978-3-319-72550-5_21)
26. Miralles, J.C., López-Sánchez, M., Salamó, M., Avila, P., Rodríguez-Aguilar, J.A.: Robust regulation adaptation in multi-agent systems. *TAAS* **8**(3), 13:1–13:27 (2013)
27. Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J.A., Vasconcelos, W., Wooldridge, M.: Online automated synthesis of compact normative systems. *ACM Trans. Auton. Adapt. Syst. (TAAS)* **10**(1), 1–33 (2015)
28. Morales, J., López-Sánchez, M., Rodríguez-Aguilar, J.A., Wooldridge, M., Vasconcelos, W.: Synthesising liberal normative systems. In: *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 433–441. International Foundation for Autonomous Agents and Multiagent Systems (2015)

29. Morales, J., Wooldridge, M., Rodríguez-Aguilar, J.A., López-Sánchez, M.: Evolutionary synthesis of stable normative systems. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, 8–12 May 2017, pp. 1646–1648 (2017)
30. Morales, J., Wooldridge, M., Rodríguez-Aguilar, J.A., López-Sánchez, M.: Off-line synthesis of evolutionarily stable normative systems. *Auton. Agents Multi-Agent Syst.* **32**(5), 635–671 (2018). <https://doi.org/10.1007/s10458-018-9390-3>
31. Muggleton, S.H., Lin, D., Pahlavi, N., Tamaddoni-Nezhad, A.: Meta-interpretive learning: application to grammatical inference. *Mach. Learn.* **94**(1), 25–49 (2014)
32. Perelli, G.: Enforcing equilibria in multi-agent systems. In: Agmon, N., Taylor, M.E., Elkind, E., Veloso, M. (eds.) Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), pp. 188–196 (2019)
33. Savarimuthu, B.T.R., Cranefield, S.: Norm creation, spreading and emergence: a survey of simulation models of norms in multi-agent systems. *Multiagent Grid Syst.* **7**(1), 21–54 (2011)
34. Shoham, Y., Tennenholtz, M.: On social laws for artificial agent societies: off-line design. *Artif. Intell.* **73**(1–2), 231–252 (1995)
35. Tinnemeier, N., Dastani, M., Meyer, J.J., van der Torre, L.: Programming normative artifacts with declarative obligations and prohibitions. In: Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2009), pp. 69–78 (2009)
36. Wooldridge, M., van der Hoek, W.: On obligations and normative ability: towards a logical analysis of the social contract. *J. Appl. Logic* **3**(3), 396–420 (2005)