Automatic Synthesis of Dynamic Norms for Multi-Agent Systems

Natasha Alechina[1], **Giuseppe De Giacomo**[2], **Brian Logan**[1,3], **Giuseppe Perelli**[2]
[1]Utrecht University
[2]Sapienza University of Rome
[3]University of Aberdeen
{n.a.alechina, b.s.logan}@uu.nl, perelli@di.uniroma1.it, degiacomo@diag.uniroma1.it

Abstract

Norms have been widely proposed to coordinate and regulate behaviour in multi-agent systems (MAS). We consider the problem of synthesising and revising the set of norms in a normative MAS to satisfy a design objective expressed in Alternating Time Temporal Logic (ATL*). ATL* is a well-established language for strategic reasoning, which allows the specification of norms that constrain the strategic behaviour of agents. We focus on dynamic norms, that is, norms corresponding to Mealy machines, that allow us to place different constraints on the agents' behaviour depending on the state of the norm and the state of the underlying MAS. We show that synthesising dynamic norms is $(k+1)$-EXPTIME where k is the alternation depth of quantifiers in the ATL* specification. Note that for typical cases of interest, k is either 1 or 2. We also study the problem of removing existing norms to satisfy a new objective, which we show to be 2EXPTIME-complete.

1 Introduction

Norms have been widely proposed as a way of coordinating and regulating behaviour in multi-agent systems (MAS) (Chopra et al. 2018). Intuitively, a *norm* expresses a pattern of desired or undesired behaviour. Examples of norms include traffic laws stating how an agent should drive, hygiene rules introduced during a pandemic, and social conventions, such as greeting someone the first time you see them each day. Norms may be generated by the designer or administrator of a MAS (prescriptive norms), e.g., traffic laws, or emerge spontaneously from interactions between agents (emergent norms) (Haynes et al. 2017). In what follows, we focus on prescriptive norms designed by the developer or administrator of a normative MAS, rather than norms that emerge spontaneously.

Norms may be implemented in a MAS through regimentation or enforcement (Grossi, Aldewereld, and Dignum 2006). A *regimented* norm is impossible to violate due to the design of the MAS. For example, only authorised users can login to the system. *Enforcement* imposes a sanction on an agent when a norm is violated, e.g., a fine or social disapproval. In the interests of brevity, we focus on regimented norms, however our approach can be easily modified to produce norms where sanctions are imposed when the norm is violated.

Rather than modelling norms as forbidding some actions in a given state of the environment (static norms), we model them as automata (Mealy machines), as this allows us to take into account the relevant history of the system. In general, it is not reasonable to assume that the entire history of a MAS is recorded in the state of the environment. For example, we may consider a norm that forbids greeting someone the agent already met on the same day, or making more than n support requests in a fixed period of time, or performing an experiment before ethical approval has been obtained, etc. We refer to such automata-based norms as *dynamic norms*. Such dynamic norms were introduced in (Huang et al. 2016; Perelli 2019).

We focus on the problem of the *automated synthesis of dynamic norms*: given a multi-agent system and an objective (a formula in ATL*), is there a dynamic norm that, when implemented in the MAS, ensures the objective is satisfied? We also consider a problem of removing dynamic norms in order to satisfy ATL* objectives. There has been considerable work on norm synthesis for static norms, e.g., (Morales et al. 2015; Morales et al. 2018; Bulling and Dastani 2016). For example, Bulling and Dastani (2016) 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. There has been less work on the synthesis of dynamic norms. In (Huang et al. 2016) the synthesis of dynamic norms to satisfy Computation Tree Logic (CTL) objectives is considered, and in (Perelli 2019), the synthesis of dynamic norms for LTL objectives and Nash equilibria.

In this paper, we present a new approach to the automated synthesis of dynamic norms to satisfy objectives expressed in ATL*. In contrast to objectives expressed in LTL, ATL* allows us to place constraints on the strategies of particular groups of agents. We consider a very general setting of norm synthesis in multi-agent systems, in which only the actions the agents may perform, the norms already in force, and the system objective to be achieved are specified. In particular, we make no assumptions about the goals, preferences or states of agents, as in designing open multi-agent systems where agents are developed by different organisations or developers, such information is often unavailable. The system objectives and norms we consider are also very general. We show that synthesising dynamic norms in this setting is $(k+1)$-EXPTIME where k is the alternation of quantifiers in the ATL* specification. Note that for typical cases of interest, k is either 1 or 2. We also study the problem of removing existing norms to satisfy a new objective, which we show to be 2EXPTIME-complete.

2 Framework

In this section we introduce our framework for reasoning about dynamic norms in multi-agent systems, and briefly recall some necessary preliminaries.

2.1 System Models

We model a normative multi-agent system as a particular kind of game, similar to concurrent game structures, but extended with norms. For example, if the MAS is a city, then norms are things like traffic laws, hygiene rules, social conventions, etc. that apply there. Norms do not have to apply to all agents in the same way. For example, some traffic gets priority; small children are exempt from wearing face masks; children are not allowed to drive a car. These considerations are reflected in the definition below.

A *k-normed Multi-Agent System* (k-MAS), sometimes also called *game*, $\mathcal{G}$ is a tuple:

$$\langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{AP}, \mathsf{Cap}, (\mathrm{Nrm}_i)_{i\leq k}, \vec{\mathsf{q}}_0, \mathsf{tr}, (\mathsf{illegal}_i)_{i\leq k}, (\eta_i)_{i\leq k}\rangle$$

where:

- $\mathrm{Ag} = \{1, \ldots, N\}$ is a finite set of N *agents*, denoted by natural numbers;
- Ac is a finite set of *actions* that agents can perform (in some state of the environment);
- AP is a finite set of *atomic propositions*; an assignment of truth values to AP determines environment states of the system;
- $\mathsf{Cap} : \mathrm{Ag} \times 2^{\mathrm{AP}} \to 2^{\mathrm{Ac}}$ is a *capability function* that assigns to each agent in each environment state the set of actions it is capable of performing in that state;
- Nrm_i is a finite set of *normative states*, one for each $i \leq k$, with $\vec{\mathrm{Nrm}} = \mathrm{Nrm}_1 \times \ldots \times \mathrm{Nrm}_k$, being the *normative vector state space*;
- $\vec{\mathsf{q}}_0 \in \vec{\mathrm{Nrm}}$ is a designated *initial normative state*;
- $\mathsf{tr} : 2^{\mathrm{AP}} \times \mathrm{Ac}^{\mathrm{Ag}} \to 2^{\mathrm{AP}}$ is a *transition function* that determines the next state of the environment given the current state of the environment and the actions performed by the agents;
- $\mathsf{illegal}_i : \mathrm{Nrm}_i \times 2^{\mathrm{AP}} \to 2^{\mathrm{Ac}\times\mathrm{Ag}}$ is the *illegality function* that returns a set pairs of actions and agents that are illegal given the current state of a norm and the environment;
- $\eta_i : \mathrm{Nrm}_i \times 2^{\mathrm{AP}} \to \mathrm{Nrm}_i$ is a *normative function* that determines the next state of a norm given the current state of the norm and the environment.

Intuitively, starting from the empty set of atomic propositions[1] and from the initial vector of normative states $\vec{\mathsf{q}}_0$, a game moves forward according to the transition function, triggered by an action tuple $\vec{a} \in \mathrm{Ac}^{\mathrm{Ag}}$, changing the underlying evaluation of the propositions in AP. Simultaneously, each normative component is updated by the corresponding normative function.

A *configuration* of $\mathcal{G}$ is a tuple $c = (\pi, \vec{\mathsf{q}}) \in 2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}}$. Sometimes, with an abuse of notation, we denote by $\mathsf{illegal}_i(\mathsf{q}_i, \pi, j) \doteq \{a \in Ac : (a, j) \in \mathsf{illegal}_i(\mathsf{q}_i, \pi)\}$ the set of actions that are made illegal for agent j by the i-th normative component. Analogously, for a configuration $c = (\pi, \vec{\mathsf{q}})$, by $\mathsf{Avl}_{\mathcal{G}}(c, j) \doteq \mathsf{Cap}(j, \pi) \setminus (\cup_{i\leq k}\mathsf{illegal}_i(\mathsf{q}_i, \pi, j))$, we denote the set of actions available to agent j in configuration c, where $\vec{\mathsf{q}}^i$ is the i-th component of $\vec{\mathsf{q}}$.

The set $\mathsf{Avl}_{\mathcal{G}}(c) \doteq \mathsf{Avl}_{\mathcal{G}}(c, 1) \times \ldots \times \mathsf{Avl}_{\mathcal{G}}(c, N)$ denotes the action vectors that are available in a configuration c.

Note that agents can select only actions that are in their capability and that are allowed by each normative component. More precisely, at each configuration $c = (\pi, \vec{\mathsf{q}})$, each agent j can select only an action $a^j \in \mathsf{Avl}_{\mathcal{G}}(c, j)$. Once each agent j has chosen an available action a^j and the corresponding action vector $\vec{a} = (a^1, \ldots, a^k)$ is formed, the system moves its components forward to the configuration $(\pi', \vec{\mathsf{q}}')$, with $\pi' = \mathsf{tr}(\pi, \vec{a})$ and $\vec{\mathsf{q}}' = (\eta_1(\mathsf{q}^1, \pi), \ldots, \eta_k(\mathsf{q}^k, \pi))$.

A *legal run*, or simply *run* is an infinite sequence $r \in (2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}})^\omega$ such that, for each $n \in \mathbb{N}$, there exists an action vector $\vec{a}_n \in \mathsf{Avl}_{\mathcal{G}}(r_n)$, such that

$$r_{n+1} = (\mathsf{tr}(\pi_n, \vec{a}_n), \eta_1(\mathsf{q}^1_n, \pi_n), \ldots, \eta_k(\mathsf{q}^k_n, \pi_n))$$

with $r_n = (\pi_n, \vec{\mathsf{q}}_n)$. We use the notation $r_{\leq n}$ to denote the prefix of r up to and including r_n. Similarly, $r_{\geq n}$ is the suffix of r starting from r_n. Moreover, we write $c \xrightarrow{\vec{a}} c'$ to denote that the action vector $\vec{a}$ determines a transition from configuration c to configuration c'.

Intuitively, a run is an infinite sequence that, starting from a given configuration, evolves according to the action vectors as the agents select them. A run is *initial* if it starts from the initial configuration, that is, $r_0 = (\emptyset, \vec{\mathsf{q}}_0)$.

A *strategy* for agent j in the game $\mathcal{G}$ is a Mealy machine of the form

$$\sigma_j = (S_j, s^0_j, \vec{\mathrm{Nrm}} \times 2^{\mathrm{AP}}, \mathrm{Ac}, \delta_j, \tau_j).$$

Intuitively, a strategy is a machinery that, for each internal state $s \in S_j$ and a configuration $c = (\pi, \vec{\mathsf{q}})$ of $\mathcal{G}$, selects an action in Ac determined by $\tau_j(s, (\pi, \vec{\mathsf{q}}))$ and updates its internal state $\delta_j(s, (\pi, \vec{\mathsf{q}}))$ accordingly. Clearly, not every strategy is available in the game, only those that comply with the *normative requirements* specified by the game itself. We say that a strategy σ_j is *legal* with respect to $\mathcal{G}$ if, and only if, $\tau_j(s, (\pi, \vec{\mathsf{q}})) \in \mathsf{Avl}(\vec{\mathsf{q}}, \pi, j)$. From now on, we restrict our attention to legal strategies, and, unless otherwise stated, we refer to them simply as strategies. Moreover, for simplicity, for a given strategy σ_j and a finite sequence $\hat{r} \in (2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}})^*$, by $\sigma_j(\hat{r}) \in \mathrm{Ac}$ we denote the action determined by the action function τ_j in σ_j after the sequence $\hat{r}$ has been fed to the internal transition function δ_j.[2]

[1] The assumption that the initial state is empty is made for convenience; the developments below would go through for an arbitrary initial state.

[2] Note that any conventional strategy $\sigma : (2^{\mathrm{AP}})^* \to \mathrm{Ac}$ has a

Consider a subset $A \subseteq \mathrm{Ag}$ of agents and a set of strategies σ_A, one for each agent $j \in A$. We say that a run r is *compatible* with σ_A if, for every $n \in \mathbb{N}$, it holds that there exists an action vector $\vec{a}_n$, with $a^j_n = \sigma_j(r_{\leq n})$ for each $j \in A$ and $\sigma_j \in \sigma_A$, such that r_{n+1} is obtained from r_n by applying $\vec{a}_n$. Essentially, a run r is compatible with σ_A if it can be generated when the agents in A play according to their respective strategies. The set of runs starting from a given configuration c and compatible with σ_A is denoted by $\mathsf{out}_{\mathcal{G}}(c, \sigma_A)$. Observe that the set of runs of a given k-MAS $\mathcal{G}$ starting from a configuration c, sometimes denoted $\mathsf{Paths}_{\mathcal{G}}(c)$, can also be written as $\mathsf{out}_{\mathcal{G}}(c, \emptyset)$. Moreover, when it is clear from the context, we omit the subscript and simply write $\mathsf{Paths}(c)$ or $\mathsf{out}(c, \sigma_A)$.

2.2 ATL*– Alternating-Time Temporal Logic

We use Alternating-Time Temporal Logic (ATL*) to express system objectives. For example, we may want a particular group of agents to be able to achieve a temporal goal, such as provide timely assistance to sick people or respond to other emergencies. Or, we may want to preclude a group of agents from achieving a goal, such as having a road race on a residential street. Norms are synthesised and added to the system in order to satisfy new objectives. Note that this means that new norms may make previous objectives unachievable (e.g. if speeds over 20 mph are made uniformly illegal, then ambulances also cannot drive at more than 20 mph). However, we can always add old objectives conjunctively to the new one when synthesising norms.

We now recall the syntax of ATL* and provide a definition of its semantics over a k-MAS. We start with the definition of the syntax.

ATL* formulas are built inductively from the set of atomic propositions AP and agents Ag, by using the following grammar, where $p \in \mathrm{AP}$ and $A \subseteq \mathrm{Ag}$:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathtt{X}\varphi \mid \varphi \mathtt{U} \varphi \mid \langle\langle A \rangle\rangle \varphi.$$

As syntactic sugar we also use $\varphi_1 \vee \varphi_2 \doteq \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \rightarrow \varphi_2 \doteq \neg\varphi_1 \vee \varphi_2$, $\varphi_1 \leftrightarrow \varphi_2 \doteq (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$, $[\![A]\!]\varphi \doteq \neg\langle\langle A \rangle\rangle\neg\varphi$, $\mathtt{F}\varphi \doteq \mathtt{true}\mathtt{U}\varphi$, and $\mathtt{G}\varphi \doteq \neg\mathtt{F}\neg\varphi$.

Intuitively, $\langle\langle A \rangle\rangle\psi$ means that each agent in A has a strategy such that, whatever the agents $\mathrm{Ag} \setminus A$ do, the resulting outcome satisfies ψ. This translates into the semantics as follows. For a given k-MAS $\mathcal{G}$ and a run r over it, the semantics of an ATL* formula φ, denoted $\mathcal{G}, r \models \varphi$, is given recursively as follows:

- $\mathcal{G}, r \models p$ iff $p \in \pi_0$, with $r_0 = (\pi_0, \vec{\mathsf{q}}_0)$ for some $\vec{\mathsf{q}}_0 \in \vec{\mathrm{Nrm}}$;
- $\mathcal{G}, r \models \neg\varphi$ iff $\mathcal{G}, r \not\models \varphi$;
- $\mathcal{G}, r \models \varphi_1 \wedge \varphi_2$ iff both $\mathcal{G}, r \models \varphi_1$ and $\mathcal{G}, r \models \varphi_2$.
- $\mathcal{G}, r \models \langle\langle A \rangle\rangle\varphi$ iff there is a strategy σ_A such that $\mathcal{G}, r' \models \varphi$, for all $r' \in \mathsf{out}(r_0, \sigma_A)$;
- $\mathcal{G}, r \models \mathtt{X}\varphi$ iff $\mathcal{G}, r_{\geq 1} \models \varphi$;
- $\mathcal{G}, r \models \varphi_1 \mathtt{U} \varphi_2$ iff there exists $j \in \mathbb{N}$ such that $\mathcal{G}, r_{\geq i} \models \varphi_1$, for all $i < j$, and $\mathcal{G}, r_{\geq j} \models \varphi_2$.

The *model-checking problem* for ATL* is: given a structure $\mathcal{G}$, a run r and an ATL* formula φ, does it hold that $\mathcal{G}, r \models \varphi$? Although the result below was established for concurrent game structures (essentially, 0-MAS without normative components), it also holds for k-MAS.

Theorem 1. *(Alur, Henzinger, and Kupferman 2002, Theorem 5.6) The model-checking problem for ATL* is 2EXPTIME-complete.*

2.3 ATL* for Strategic Permission and Prohibition

In this section, we introduce two important classes of normative specifications, *strategic permissions* and *strategic prohibitions*, which can be expressed in ATL* in a simple way.

Definition 1 (Strategic Permission)**.** *A strategic permission is a positive Boolean combination of formulas of the form $\langle\langle A \rangle\rangle\varphi$, where φ is a purely temporal formula (not containing any strategy quantification $\langle\langle A' \rangle\rangle$).*

Intuitively, a strategic permission objective $\langle\langle A \rangle\rangle\varphi$ ensures that the agent(s) A have the strategic ability to bring about φ. Strategic permissions are a form of reachability property that specify that agents should have the freedom to do something if they wish. Note that a norm satisfying a strategic permission may restrict the actions of agents not in A, that is, that the agents $\mathrm{Ag} \setminus A$ may be constrained so that they not have a strategy to prevent A achieving ϕ. For example,

Example 1 (Strategic Permission)**.** *The property that ambulances should be able to park in their designated places at a hospital can be expressed in ATL* as $\bigwedge_{i \in A} \langle\langle i \rangle\rangle \mathtt{G}(\mathtt{ready}_i \rightarrow \mathtt{X}\ \mathtt{park}_i)$, where A is the set of ambulances, $\mathtt{ready}_i$ stands for 'ambulance i is ready to park' and $\mathtt{park}_i$ stands for 'ambulance i is parked in its designated place'.*

This property can be enforced by making it illegal for all agents different from i to park in i's place.

Definition 2 (Strategic Prohibition)**.** *A strategic prohibition is a positive Boolean combination of formulas of the form $\neg\langle\langle A \rangle\rangle\varphi$, where φ is a purely temporal formula.*

A strategic prohibition objective $\neg\langle\langle A \rangle\rangle\varphi$ ensures that the agent(s) A do *not* have the strategic ability to bring about φ. Strategic prohibitions are a form of safety property that specify that agents should not have the freedom to bring something about even if they wish to. In general, a norm satisfying a strategic prohibition restricts the actions of agents in A. For example,

Example 2 (Strategic Prohibition)**.** *The property that a set of juvenile delinquents D should not be able to organise a road race can be expressed in ATL* as $\neg\langle\langle D \rangle\rangle \mathtt{F}\ \mathtt{road\_race}$.*

corresponding strategy of the form $\sigma : (2^{\mathrm{AP}} \times \vec{\mathrm{Nrm}})^* \rightarrow \mathrm{Ac}$, because the system is deterministic and the next normative state can be computed from the previous normative and environment state. Here we consider regular strategies, that is, those that can be represented as a Mealy machines. This is not a limitation, as, when the specification language is ω-regular, if a strategy satisfying the specification exists, then a regular strategy also exists. Moreover, this representation means strategies have a compact form, and is similar to the form of dynamic norms introduced later.

This property can be enforced by making it illegal for all agents to drive fast, but such a blanket prohibition may conflict with a strategic permission allowing ambulances to drive fast: $\bigwedge_{i\in A} \langle\langle i\rangle\rangle \mathtt{G}\,\mathtt{X}\,\mathtt{speed}_i^{\geq 20}$. A norm that makes driving fast illegal only for agents in D will not violate the latter property.

2.4 ATL* with Strategy Context

For technical purposes, we also make use of an extension of ATL*, namely ATL* with strategy context (Laroussinie and Markey 2015), denoted ATL$^*_{\mathsf{sc}}$, in which another form of quantification $\langle\langle\cdot A\cdot\rangle\rangle\varphi$ is used, together with its dual $[\![\cdot A\cdot]\!]\varphi$, defined as $\neg\langle\langle\cdot A\cdot\rangle\rangle\neg\varphi$. Formulas of ATL$^*_{\mathsf{sc}}$ are interpreted over the same structures as ATL* together with a strategy context. More formally, for a given k-MAS $\mathcal{G}$ and a set of strategies σ_B for the subset $B \subseteq \mathrm{Ag}$ of agents, we write:

- $\mathcal{G}, r \models_{\sigma_B} \langle\langle\cdot A\cdot\rangle\rangle\varphi$ if there is a strategy σ_A such that $\mathcal{G}, r' \models_{\sigma_B\circ\sigma_A} \varphi$, for all $r' \in \mathsf{out}(r_0, \sigma_B \circ \sigma_A)$;

where $\sigma_B \circ \sigma_A \doteq \sigma_B \cup \sigma_{A\setminus B}$ denotes the set of strategies obtained from σ_B by adding strategies of σ_A that are for agents in A but not in B.

The *quantifier alternation*, or simply *alternation* of a ATL$^*_{\mathsf{sc}}$ formula φ is the number of times an existential quantification $\langle\langle\cdots\rangle\rangle$ is followed by a universal one $[\![\cdots]\!]$, and vice-versa. The model checking problem for ATL$^*_{\mathsf{sc}}$ was shown to have a TOWER-complete complexity in (Laroussinie and Markey 2015), whose height depends on the number of alternations in the formula. More precisely, they show that model checking an ATL$^*_{\mathsf{sc}}$ formula with h alternations is $(h+1)$-EXPTIME-complete.

Theorem 2. *(Laroussinie and Markey 2015, Corollary 14) The model-checking problem for an ATL$^*_{\mathsf{sc}}$ formula with h nested strategy quantifiers is $(h+1)$-EXPTIME-complete.*

2.5 Norms

A *Norm* over a k-MAS $\mathcal{G}$ is a Mealy machine of the form

$$\mathcal{N} = \langle \mathrm{Nrm}, \mathsf{q}_0, 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta, \mathsf{illegal}\rangle$$

A norm takes a state of the world as input and returns a set pairs of actions and agents that are illegal given the current state of a norm and the environment. $\mathcal{N}$ is well-defined on every k-MAS $\mathcal{G}$ having the same set Ag of agents, Ac of actions and AP of propositions.

Example 3. *A trivial norm (that does not impose any restrictions) can be defined as follows:* $\mathrm{Nrm} = \{\mathsf{q}_0\}$, $\eta(\mathsf{q}_0, \pi) = \mathsf{q}_0$ *for all* $\pi \in 2^{\mathrm{AP}}$, *and* $\mathsf{illegal}(\mathsf{q}_0, \pi) = \emptyset$ *for all* π.

Example 4. *A norm that forbids agent j executing action a after encountering proposition p twice can be defined as* $\mathrm{Nrm} = \{\mathsf{q}_0, \mathsf{q}_1, \mathsf{q}_2\}$, *where for* $i \in \{0, 1\}$, $\eta(\mathsf{q}_i, \pi) = \mathsf{q}_i$ *if* $p \notin \pi$, $\eta(\mathsf{q}_i, \pi) = \mathsf{q}_{i+1}$ *if* $p \in \pi$, $\mathsf{illegal}(\mathsf{q}_i, \pi) = \emptyset$ *for all* π; *and* $\eta(\mathsf{q}_2, \pi) = \mathsf{q}_2$ *for all* π, $\mathsf{illegal}(\mathsf{q}_2, \pi) = \{(a, j)\}$.

The following example is adapted from (Huang et al. 2016) (slightly simplified for brevity).

Example 5. *In a system consisting of n producers and m consumers, the norm prevents producers failing to supply consumers whose 'turn' it is to be served. Actions of each producer j are of the form $B \subseteq \{1, \ldots, m\}$, corresponding to serving the set of consumers B. If it is the turn of consumer i, then illegal actions are B such that $i \notin B$. This norm can be defined as follows:* $\mathrm{Nrm} = \{\mathsf{q}_1, \ldots, \mathsf{q}_m\}$, *(the states of the norm correspond to which consumer's turn it is),* $\eta(\mathsf{q}_i, \pi) = \mathsf{q}_{i+1 (mod\ m)}$ *for all* π, $\mathsf{illegal}(\mathsf{q}_i, \pi) = \{(B, j) \mid i \notin B\}$.

Conditional prohibition norms introduced in (Tinnemeier et al. 2009) are tuples of the form (condition, prohibited property, deadline, sanction). Condition is a propositional formula that describes states after which the prohibition comes into effect (until a state satisfying the deadline is reached). Sanction is either a negative utility or an indication that violating the norm is impossible (the norm is regimented). A regimented conditional prohibition norm can also be defined in our formalism.

Example 6. *A norm which prohibits agent j moving between the start and the end of some procedure can be defined as follows.* $\mathrm{Nrm} = \{\mathsf{q}_1, \mathsf{q}_2\}$, $\eta(\mathsf{q}_1, \pi) = \mathsf{q}_1$ *if start* $\notin \pi$ *and* $\eta(\mathsf{q}_1, \pi) = \mathsf{q}_2$ *otherwise.* $\mathsf{illegal}(\mathsf{q}_1, \pi) = \emptyset$ *and* $\mathsf{illegal}(\mathsf{q}_2, \pi) = \{move\}$.

Consider a norm $\mathcal{N}_{k+1}$ whose components are all indexed with $k+1$. We can implement $\mathcal{N}_{k+1}$ on a k-MAS $\mathcal{G}$ to obtain a $(k+1)$-MAS defined as the tuple $\mathcal{G} \oplus \mathcal{N}_{k+1} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{AP}, \mathsf{Cap}, (\mathrm{Nrm}_i)_{i\leq k+1}, \vec{\mathsf{q}}_0, \mathsf{tr}, (\mathsf{illegal}_i)_{i\leq k+1}, (\eta_i)_{i\leq k+1}\rangle$, containing an extra normative state component, which are the states of $\mathcal{N}_{k+1}$, and whose evolution is determined by its normative function η_{k+1}.

Intuitively, $\mathcal{N}_{k+1}$ introduces more restrictions on the actions for the agents when implemented in a given k-MAS $\mathcal{G}$. Indeed, for every configuration c in the original game and its extension with the state of $\mathcal{N}_{k+1}$, c', it holds that $\mathsf{Avl}_{\mathcal{G}\oplus\mathcal{N}_{k+1}}(c', j) \subseteq \mathsf{Avl}_{\mathcal{G}}(c, j)$ for every agent $j \in \mathrm{Ag}$.

Observe also that every normative state component i of a k-MAS $\mathcal{G}$ can be regarded as a norm $\mathcal{N}_i = \langle \mathrm{Nrm}_i, e_0^i, 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta_i, \mathsf{illegal}_i\rangle$ and so $\mathcal{G}$ can be obtained from a 0-MAS where the norms $\mathcal{N}_1 \ldots, \mathcal{N}_k$ have been applied one by one. A norm $\mathcal{N}_i$ can also be removed from $\mathcal{G}$, denoted $\mathcal{G} \ominus \mathcal{N}_i$, resulting in a $(k-1)$ MAS.

2.6 Norm Synthesis and Revision

We can now define the two problems addressed in this paper. The first is *Norm Synthesis*, which is the problem of finding a norm $\mathcal{N}$ for a k-MAS $\mathcal{G}$ that, if implemented, makes a given ATL* formula φ true over $\mathcal{G} \oplus \mathcal{N}$.

Definition 3 (Norm Synthesis)**.** *For a given k-MAS $\mathcal{G}$ and an ATL* formula φ, determine whether there exists a norm $\mathcal{N}_{k+1}$ such that $\mathcal{G} \oplus \mathcal{N}_{k+1} \models \varphi$.*

On the other hand, *Norm Removal* is the opposite problem. That is, given a k-MAS $\mathcal{G}$, to identify a subset $\{\mathcal{N}_{\iota_1}, \ldots \mathcal{N}_{\iota_h}\}$ of the already implemented norms that, if removed, make a given ATL$^*_{\mathsf{sc}}$ formula true.

Definition 4 (Norm Removal)**.** *For a given k-MAS $\mathcal{G}$ and an ATL* formula φ, determine whether there exists a subset $\{\mathcal{N}_{\iota_1}, \ldots \mathcal{N}_{\iota_h}\}$ of norms such that $\mathcal{G} \ominus \mathcal{N}_{\iota_1} \ominus \ldots \ominus \mathcal{N}_{\iota_h} \models \varphi$.*

3 Automated Norm Synthesis

We start by solving Norm Synthesis. We do this by reinterpreting norms of a given k-MAS $\mathcal{G}$ as strategies of an accessory k-MAS $\mathcal{G}'$, reducing the norm synthesis to strategy synthesis in this accessory game.

In order to simplify our reasoning, first observe that a k-MAS $\mathcal{G}$ can always be regarded as a 1-MAS where the only applied norm is the product of the k norms given in $\mathcal{G}$. For this reason, from now on, we assume without loss of generality that $\mathcal{G}$ is a 1-MAS.

To illustrate Norm Synthesis, we use the following running example.

Example 7. *$\mathcal{G}_{ex}$ has two agents, 1 and 2, two actions $wait$ and ask, two propositions $rest$ and $work$. Agent 1 can only do $wait$. Agent 2 can always do $wait$ and ask. The transition function is: if both agents perform $wait$, then $rest$ becomes true. If agent 2 performs ask, then $work$ becomes true. The initial norm is trivial (it has one state and no illegal actions). The task is to synthesise a norm that will enable agent 1 to have two consecutive moments of rest after $work$ becomes true: $\varphi_{rest} = \langle\langle 1\rangle\rangle \mathtt{G}(work \to \mathtt{X}(rest \wedge \mathtt{X}rest))$.*

Consider, for instance, a strategy for Agent 2 defined as $\sigma_2 = \langle S_2, s_2^0, 2^{\mathrm{AP}}, \mathrm{Ac}, \delta_2, \tau_2\rangle$,[3] with $S_2 = \{s_2^0\}$, $\delta_2(s_2^0, \pi) = s_2^0$ and $\tau_2(s_2^0, \pi) = ask$, for each $\pi \in 2^{\mathrm{AP}}$. Note that this strategy prevents φ_{rest} from becoming true, as there is no strategy for Agent 1 that, combined with σ_2, makes the temporal part satisfied. To prevent Agent 2 from repeatedly executing ask, we can implement a norm that works as a counter: once the proposition $work$ becomes true, Agent 2 is not allowed to execute action ask twice in a row.

In the following, we show how to solve Norm Synthesis automatically, by employing the construction of an accessory game. The definitions are inspired by the *encoding game* defined in (Perelli 2019).

Construction 1 (Accessory game)**.** *Consider a 1-MAS $\mathcal{G} = \langle \mathrm{Ag}, \mathrm{Ac}, \mathrm{AP}, \mathsf{Cap}, \mathrm{Nrm}_1, \mathsf{q}_0^1, \mathsf{tr}, \mathsf{illegal}_1, \eta_1\rangle$ and define the accessory 1-MAS as:*

$$\mathcal{G}' = \langle \mathrm{Ag}', \mathrm{Ac}', \mathrm{AP}', \mathsf{Cap}', \mathrm{Nrm}_1, \mathsf{q}_0^1, \mathsf{tr}', \mathsf{illegal}_1', \eta_1'\rangle,$$

where

- $\mathrm{Ag}' = \{0\} \cup \mathrm{Ag}$ *includes a 0-agent, sometimes called the* normative agent*;*
- $\mathrm{Ac}' = \mathrm{Ac} \cup (2^{\mathrm{Ac}\times\mathrm{Ag}})$ *includes all possible sets of pairs of actions and agents as possible actions;*
- $\mathrm{AP}' = \mathrm{AP} \cup (\mathrm{Ac}\times\mathrm{Ag})$ *includes the set of pairs of actions and agents in the atomic propositions;*
- $\mathsf{Cap}'(j, \pi') = \begin{cases} 2^{\mathrm{Ac}\times\mathrm{Ag}}, & \text{if } j = 0 \\ \mathsf{Cap}(j, \pi'_{\restriction \mathrm{AP}}) \setminus (\{j\} \cap \pi'), & \text{o/w} \end{cases}$
- $\mathsf{tr}'(\pi', \vec{a}) = \mathsf{tr}(\pi'_{\restriction \mathrm{AP}}, \vec{a}_{-0}) \cup \vec{a}_0$*;*
- $\mathsf{illegal}'(\mathsf{q}_1, \pi') = \mathsf{illegal}(\mathsf{q}_1, \pi'_{\restriction \mathrm{AP}})$*;*
- $\eta'(\mathsf{q}_1, \pi') = \eta(\mathsf{q}_1, \pi'_{\restriction \mathrm{AP}})$.

[3]Note that we consider only 2^{AP} as input alphabet, since only the trivial norm is currently implemented in the game.

The idea of this construction is to embed the reasoning about the existence of norms in terms of a strategy in the accessory game. To do so, we add to $\mathcal{G}$ an extra agent, the normative agent, whose capability is precisely that of *preventing* actions of other agents. To suitably encode this capability, we also expand the state and action spaces of the game with all possible subsets of pairs of agents and actions. The transition function τ' mimics τ with regards to the evaluation of AP and copies the action taken by the normative agent into the next state. The capability function Cap' is also extended accordingly. It prescribes agent 0 to take actions that correspond to the output of the norm for $\mathcal{G}$ (the set of pairs of actions and agents that are illegal according to the norm). Regarding the other agents, it assigns the subset of actions originally available in $\mathcal{G}$ which are not prevented by the action taken by the normative agent in the previous step.

Example 8. *The accessory game corresponding to $\mathcal{G}_{ex}$ is $\mathcal{G}'$ where:*

- $\mathrm{Ag}' = \{0, 1, 2\}$*;*
- $\mathrm{Ac}' = \{wait, ask\} \cup 2^{\{wait, ask\}\times\{1,2\}}$*;*
- $\mathrm{AP}' = \{work, rest\} \cup \{(wait, 1), (wait, 2), (ask, 1), (ask, 2)\}$*;*
- $\mathsf{Cap}'(0, \pi') = 2^{\{wait, ask\}\times\{1,2\}}$*, and* $\mathsf{Cap}'(j, \pi') = \mathsf{Cap}(j, \pi'_{\restriction \mathrm{AP}}) \setminus (\{j\} \cap \pi')$*, for $j = 1, 2$. For instance, it holds that* $\mathsf{Cap}'(2, \{work, (2, ask)\}) = \{wait\}$.
- $\mathsf{tr}'(\pi', \vec{a}) = \mathsf{tr}(\pi'_{\restriction \mathrm{AP}}, \vec{a}_{-0}) \cup \vec{a}_0$. *For instance, it holds that* $\mathsf{tr}'(\{work\}, (wait, wait, \{(2, ask)\})) = \{rest, (2, ask)\}$*;*
- $\mathsf{illegal}'(\mathsf{q}_1, \pi')$ *and* $\eta'(\mathsf{q}_1, \pi')$ *defined as in Construction 1.*

We can now make a connection between norms for $\mathcal{G}$ and strategies of agent 0 in $\mathcal{G}'$. Indeed, a strategy for the normative agent is of the form $\sigma_0 = \langle S_0, s_0^0, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}'}, \mathrm{Ac}', \delta_0, \tau_0\rangle$. Observe that $\mathsf{Cap}'(0, \pi') = 2^{\mathrm{Ac}\times\mathrm{Ag}}$, and the set $2^{\mathrm{AP}'} = 2^{\mathrm{AP}\cup(\mathrm{Ac}\times\mathrm{Ag})}$ is isomorphic to $2^{\mathrm{AP}} \times 2^{\mathrm{Ac}\times\mathrm{Ag}}$. This allows us to rewrite the strategy as $\sigma_0 = \langle S_0, s_0^0, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}} \times 2^{\mathrm{Ac}\times\mathrm{Ag}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \delta_0, \tau_0\rangle$, from which a corresponding norm can be constructed.

Construction 2 (Norm construction)**.** *For a given strategy $\sigma_0 = \langle S_0, s_0^0, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}} \times 2^{\mathrm{Ac}\times\mathrm{Ag}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \delta_0, \tau_0\rangle$ of agent 0 in $\mathcal{G}'$, the norm $\mathcal{N}_{\sigma_0}$ for $\mathcal{G}$ is defined as:*

$$\langle S_0 \times \mathrm{Nrm}_1 \times 2^{\mathrm{Ac}\times\mathrm{Ag}}, (s_0, \mathsf{q}_0, \emptyset), 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta, \mathsf{illegal}\rangle,$$

where $\eta((s, \mathsf{q}_1, A), \pi) = (s', \mathsf{q}_1', A')$, with

- $s' = \delta_0(s, (\mathsf{q}_1, \pi \cup A))$,
- $\mathsf{q}_1' = \eta_1(\mathsf{q}_1, \pi)$*, and*
- $A' = \tau_0(s, (\mathsf{q}_1, \pi \cup A), \pi)$,

and $\mathsf{illegal}((s, \mathsf{q}_1, A), \pi) = \tau_0(s, (\mathsf{q}_1, \pi \cup A), \pi)$*, for every $s \in S_0$, $\mathsf{q}_1 \in \mathrm{Nrm}_1$, and $A \in 2^{\mathrm{Ac}\times\mathrm{Ag}}$.*

On the other hand, we can generate a strategy for the normative agent.

Construction 3 (Strategy construction)**.** *For a given norm $\mathcal{N}_2 = \langle \mathrm{Nrm}_2, \mathsf{q}_0^2, 2^{\mathrm{AP}}, 2^{\mathrm{Ac}\times\mathrm{Ag}}, \eta_2, \mathsf{illegal}_2 \rangle$ in $\mathcal{G}$, the strategy $\sigma_{\mathcal{N}_2}$ for agent 0 in $\mathcal{G}'$ is defined as:*

$$\langle S, s_0, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}'}, \mathrm{Ac}', \delta, \tau \rangle,$$

where $S = \mathrm{Nrm}_2$, $s_0 = \mathsf{q}_0^2$, and the internal and output functions are defined as:

- $\delta(\mathsf{q}_2, (\mathsf{q}_1, \pi')) = \eta_2(\mathsf{q}_2, \pi'_{\restriction \mathrm{AP}})$,
- $\tau(\mathsf{q}_2, (\mathsf{q}_1, \pi')) = \mathsf{illegal}_2(\mathsf{q}_2, \pi'_{\restriction \mathrm{AP}})$,

for every $\mathsf{q}_2 \in \mathrm{Nrm}_2$ and $\pi' \in 2^{\mathrm{AP}'}$.

Note that a run r in $\mathcal{G}'$ belongs to the set $(2^{\mathrm{AP}'} \times \mathrm{Nrm}_1)^\omega$. By $r_{\restriction \mathrm{AP}}$ we denote the sequence in $(2^{\mathrm{AP}})^\omega$ obtained from r by projecting out everything but the sequence of evaluations in AP. Analogously, such projection can be extended to sets of runs. In particular, we consider $\mathsf{Paths}_{\mathcal{G}'}(c)_{\restriction \mathrm{AP}}$ to be the set of projections over all possible paths in $\mathcal{G}'$ starting from configuration c, and $\mathsf{out}_{\mathcal{G}'}(\sigma_A, c)_{\restriction \mathrm{AP}}$ as the projections of outcomes in $\mathcal{G}'$ with initial configuration c, with σ_A be the strategy profile for the set A of agents.

There is a connection between σ_0 and the corresponding norm. Precisely, they produce runs in games that relate to the same projections over AP, as stated by the lemma below.

Lemma 1. *For a given 1-MAS $\mathcal{G}$, its accessory game $\mathcal{G}'$, and a state $\pi \in 2^{\mathrm{AP}}$, the following two statements hold:*

1. *For every strategy σ_0 of agent 0 in $\mathcal{G}'$, it holds that* $\mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}} = \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$*;*
2. *For every norm $\mathcal{N}$ on $\mathcal{G}$, it holds that* $\mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}} = \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$.

Proof. We prove the two statements separately.

1. The proof proceeds by double inclusion. For the left to right direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ and let r' be a run in $\mathcal{G}'$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \dots, \vec{a}^h$ in $\mathcal{G}'$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ and $\vec{a}_0^{h'}$ is always the action selected by σ_0 as the execution evolves. Note that such a sequence exists, as the run r' belongs to $\mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$. By the construction of $\mathcal{N}_\sigma$, we obtain that the sequence $(\vec{a}^0)_{-0}, \dots, (\vec{a}^h)_{-0}$ generates a partial run in $\mathcal{G} \oplus \mathcal{N}_{\sigma_0}$, denoted $r_0, \dots, r_{h+1}$ that can be extended to a run $r \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ in a way that $(r_{h\leq h+1})_{\restriction \mathrm{AP}} = (r'_{h\leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which such property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$.

For the right to left direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ and let r' be a run in $\mathcal{G} \oplus \mathcal{N}_{\sigma_0}$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \dots, \vec{a}^h$ in $\mathcal{G} \oplus \mathcal{N}$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ Note that such sequence exists as the run r' belongs to $\mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$. By the construction of $\mathcal{N}_\sigma$, we obtain that the sequence $(\vec{a}^0, a_0^0), \dots, (\vec{a}^h, a_0^h)$, with $a_0^0, \dots a_0^h$ being the sequence of actions generated by σ_0 generates a partial run in $\mathcal{G}'$, denoted $r_0, \dots, r_{h+1}$ that can be extended to a run $r \in \mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ in a way that $(r_{h\leq h+1})_{\restriction \mathrm{AP}} = (r'_{h\leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which such property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{out}_{\mathcal{G}'}(\sigma_0, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$.

2. The proof proceeds by double inclusion. For the left to right direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ and let r' be a run in $\mathcal{G} \oplus \mathcal{N}$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \dots, \vec{a}^h$ in $\mathcal{G} \oplus \mathcal{N}$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ Note that such a sequence exists, as the run r' belongs to $\mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$. By the construction of $\sigma_{\mathcal{N}}$, we obtain that the sequence $(\vec{a}^0, a_0^0), \dots, (\vec{a}^h, a_0^h)$, with $a_0^0, \dots a_0^h$ being the sequence of actions generated by $\sigma_{\mathcal{N}}$ generates a partial run in $\mathcal{G}'$, denoted $r_0, \dots, r_{h+1}$ that can be extended to a run $r \in \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ in a way that $(r_{h\leq h+1})_{\restriction \mathrm{AP}} = (r'_{h\leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which the property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$.

For the right to left direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$ and let r' be a run in $\mathcal{G}'$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \dots, \vec{a}^h$ in $\mathcal{G}'$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ and $\vec{a}_0^{h'}$ is always the action selected by $\sigma_{\mathcal{N}}$ as the execution evolves. Note that such a sequence exists, as the run r' belongs to $\mathsf{out}_{\mathcal{G}'}(\sigma_{\mathcal{N}}, (\pi, \mathsf{q}_0^1))_{\restriction \mathrm{AP}}$. By the construction of $\sigma_{\mathcal{N}}$, we obtain that the sequence $(\vec{a}^0)_{-0}, \dots, (\vec{a}^h)_{-0}$ generates a partial run in $\mathcal{G} \oplus \mathcal{N}$, denoted $r_0, \dots, r_{h+1}$ that can be extended to a run $r \in \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$ in such a way that $(r_{h\leq h+1})_{\restriction \mathrm{AP}} = (r'_{h\leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which such property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{Paths}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0)_{\restriction \mathrm{AP}}$. $\square$

Lemma 1 establishes the equivalence between norms in $\mathcal{G}$ and their corresponding strategies in $\mathcal{G}'$, in the sense that they allow/disallow the same set of possible outcomes.

We now show that, once a norm $\mathcal{N}$ is implemented, it is also possible to establish a correspondence between agents' strategies in $\mathcal{G} \oplus \mathcal{N}$ and $\mathcal{G}'$, when agent 0 is employing the corresponding strategy $\sigma_{\mathcal{N}}$. Indeed, consider again a 1-MAS $\mathcal{G}$ and the accessory $\mathcal{G}'$ as defined above. We have the following:

Construction 4 (Outgoing strategy mapping)**.** *Consider a norm $\mathcal{N}_2$ over $\mathcal{G}$ with $\sigma_{\mathcal{N}}$ being the corresponding strategy in $\mathcal{G}'$ for the normative agent. Moreover, consider an agent $j \in \mathrm{Ag}$ and a strategy σ_j for j in $\mathcal{G} \oplus \mathcal{N}_2$ of the form $\sigma_j = \langle S_j, s^0_k, \mathrm{Nrm}_1 \times \mathrm{Nrm}_2 \times 2^{\mathrm{AP}}, \mathrm{Ac}, \delta_j, \tau_j\rangle$. The* outgoing strategy *σ'_j for agent $j \in \mathrm{Ag}$ in $\mathcal{G}'$ is defined as*

$$\langle S'_j, s^{0\prime}_k, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}'}, \mathrm{Ac}', \delta'_j, \tau'_j\rangle,$$

where $S'_j = S_j \times \mathrm{Nrm}_2$, $s^{0\prime}_k = (s^0_k, \mathsf{q}^0_2)$, and the internal and output functions are defined as:

- $\delta'_j((s, \mathsf{q}_2), (\mathsf{q}_1, \pi')) = (\delta_j(s, (\mathsf{q}_1, \mathsf{q}_2, \pi'_{\restriction \mathrm{AP}})), \eta_2(\mathsf{q}_2, \pi'_{\restriction \mathrm{AP}}))$*;*
- $\tau'_j((s, \mathsf{q}_2), (\mathsf{q}_1, \pi')) = \tau_j(s, (\mathsf{q}_1, \mathsf{q}_2, \pi'_{\restriction \mathrm{AP}}))$.

On the other hand, we can map a strategy in $\mathcal{G}'$ back to a strategy in $\mathcal{G} \oplus \mathcal{N}$.

Construction 5 (Incoming strategy mapping)**.** *Consider a strategy σ'_j in $\mathcal{G}'$ for agent j, given as $\sigma'_j = \langle S'_j, s^{0\prime}_k, \mathrm{Nrm}_1 \times 2^{\mathrm{AP}'}, \mathrm{Ac}', \delta'_j, \tau'_j\rangle$.*

The incoming strategy *for agent j in $\mathcal{G} \oplus \mathcal{N}_2$ is defined as*

$$\sigma_j = \langle S_j, s^0_k, \mathrm{Nrm}_1 \times \mathrm{Nrm}_2 \times 2^{\mathrm{AP}}, \mathrm{Ac}, \delta_j, \tau_j\rangle,$$

where $S_j = S'_j$, $s^0_k = s^{0\prime}_k$, and the internal and output functions are defined as:

- $\delta_j(s, (\mathsf{q}_1, \mathsf{q}_2, \pi)) = \delta'_j(s, (\mathsf{q}_1, \pi \cup \mathsf{illegal}_2(\mathsf{q}_2, \pi)))$*;*
- $\tau_j(s, (\mathsf{q}_1, \mathsf{q}_2, \pi)) = \tau'_j(s, (\mathsf{q}_1, \pi \cup \mathsf{illegal}_2(\mathsf{q}_2, \pi)))$.

As for norms and strategies for agent 0, the same kind of correspondence holds between strategies for the agents in the two structures, as the following lemma states.

Lemma 2. *For a given 1-MAS $\mathcal{G}$, its accessory game $\mathcal{G}'$, a pair $(\mathcal{N}, \sigma_0)$ of the corresponding norm and normative agent strategy, a set $A \subseteq \mathrm{Ag}$ of agents, and a state $\pi \in 2^{\mathrm{AP}}$, the following two statements hold:*

1. *For every strategy σ_A in $\mathcal{G} \oplus \mathcal{N}$ it holds that $\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}} = \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$;*
2. *For every strategy σ'_A in $\mathcal{G}'$ it holds that $\mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}} = \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$.*

Proof. We prove the two statements separately.

1. The proof proceed by double inclusion. For the left to right direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$ with r' being a run in $\mathcal{G} \oplus \mathcal{N}$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \ldots, \vec{a}^h$ in $\mathcal{G} \oplus \mathcal{N}$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ and $\vec{a}^{h'}_j$ is always the action selected by σ_j, for each $j \in A$, as the execution evolves. Note that such a sequence exists, as the run r' belongs to $\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$. By the construction of $\sigma_{\mathcal{N}}$ and σ'_A, we obtain that the sequence $(\vec{a}^0, a^0_0), \ldots, (\vec{a}^h, a^h_0)$ with $a^0_0, \ldots, a^h_0$ being he sequence of actions generated by $\sigma_{\mathcal{N}}$, generates a partial run in $\mathcal{G}'$, denoted $r_0, \ldots, r_{h+1}$ that can be extended to a run $r \in \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$ in such a way that $(r_{h \leq h+1})_{\restriction \mathrm{AP}} = (r'_{h \leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which such property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$.

For the right to left direction, consider a sequence $r'_{\restriction \mathrm{AP}} \in \mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$ with r' being a run in $\mathcal{G}'$ from which $r'_{\restriction \mathrm{AP}}$ is obtained by projection. By contradiction, let us assume that r'_{AP} does not belong to $\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$ and let $h \in \mathbb{N}$ be the greatest natural number for which $(r'_{\leq h})_{\restriction \mathrm{AP}} \cdot r'' \in \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$, for some sequence r''. More specifically, there does not exist any sequence $\hat{r}$ such that $(r'_{\leq h+1})_{\restriction \mathrm{AP}} \cdot \hat{r} \in \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$. Now, consider a sequence of action vectors $\vec{a}^0, \ldots, \vec{a}^h$ in $\mathcal{G}'$ such that, for every $h' \leq h$, it holds that $r'_{h'} \xrightarrow{\vec{a}^{h'}} r'_{h'+1}$ and $\vec{a}^{h'}_j$ is always the action selected by σ_j, for each $j \in A \cup \{0\}$, as the execution evolves. Note that such a sequence exists, as the run r' belongs to $\mathsf{out}_{\mathcal{G}'}((\pi, \mathsf{q}^1_0), \sigma_0 \cup \sigma'_A)_{\restriction \mathrm{AP}}$.

By the construction of $\sigma_{\mathcal{N}}$ and σ'_A of Construction 3 and Construction 4, respectively, we obtain that the sequence $(\vec{a}^0)_{-0}, \ldots, (\vec{a}^h)_{-0}$ generates a partial run in $\mathcal{G} \oplus \mathcal{N}$, denoted $r_0, \ldots, r_{h+1}$ that can be extended to a run $r \in \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$ in such a way that $(r_{h \leq h+1})_{\restriction \mathrm{AP}} = (r'_{h \leq h+1})_{\restriction \mathrm{AP}}$, resulting in a contradiction with h being the greatest number for which such property holds, and subsequently that $r'_{\mathrm{AP}} \notin \mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\pi, \vec{\mathsf{q}}_0, \sigma_A)_{\restriction \mathrm{AP}}$.

2. Note that the proof of this statement is identical to the previous one, except for the fact that this time we use Construction 2 and Construction 5 for $\mathcal{N}_{\sigma_0}$ and σ_A, respectively. $\square$

An immediate consequence of Lemma 1 and Lemma 2 is that the existence of a norm in $\mathcal{G}$ can be restated as the existence of the corresponding strategy for agent 0 in $\mathcal{G}'$.

Lemma 3. *For a given 1-MAS $\mathcal{G}$ and an ATL* formula φ, the following two statements are equivalent:*

1. *There exists a norm $\mathcal{N}$ over $\mathcal{G}$ such that $\mathcal{G} \oplus \mathcal{N} \models \varphi$*
2. *$\mathcal{G}' \models \langle\!\langle \cdot 0 \cdot \rangle\!\rangle \varphi'$, where φ' is obtained from φ by replacing all strategy quantifiers $\langle\!\langle A \rangle\!\rangle$ with $\langle\!\langle \cdot A \cdot \rangle\!\rangle$.*

Proof. We prove the equivalence by double implication. Both directions proceed by structural induction on the formula φ. Here, for simplicity, we show only the base case of $\varphi = \langle\!\langle A \rangle\!\rangle \psi$, with $\psi \in LTL$, as all the others are a simple variant of this.

To prove that Statement 1 implies Statement 2, assume that $\mathcal{G} \oplus \mathcal{N} \models \varphi$. Therefore, there exists σ_A such that $r \models \psi$ for every $r \in (\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\sigma_A, \emptyset))_{\restriction \mathrm{AP}}$.[4] Then, consider the strategy σ'_A in $\mathcal{G}'$ obtained from σ_A by applying Construction 4. By Lemma 2, we obtain that $(\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}}(\sigma_A, \emptyset))_{\restriction \mathrm{AP}} = (\mathsf{out}_{\mathcal{G}'}(\sigma'_A \cup \sigma_{\mathcal{N}}, \emptyset))_{\restriction \mathrm{AP}}$ with $\sigma_{\mathcal{N}}$ obtained from $\mathcal{N}$ by applying Construction 3, which in turns implies that $\mathcal{G}' \models \langle\!\langle \cdot 0 \cdot \rangle\!\rangle \varphi'$.

The other direction proceeds as follows. Assume that $\mathcal{G}' \models \langle\!\langle \cdot 0 \cdot \rangle\!\rangle \langle\!\langle A \rangle\!\rangle \psi$. Therefore, there exists a strategy σ_0 for agent 0 and a strategy profile σ'_A, such that $r \models \psi$ for every $r \in (\mathsf{out}_{\mathcal{G}'}(\sigma'_A \cup \sigma_{\mathcal{N}}, \emptyset))_{\restriction \mathrm{AP}}$. Now consider the norm $\mathcal{N}_{\sigma_0}$ obtained from Construction 2 and the strategy profile σ_A obtained from Construction 5. By Lemma 2, we obtain that $(\mathsf{out}_{\mathcal{G}'}(\sigma'_A \cup \sigma_{\mathcal{N}}, \emptyset))_{\restriction \mathrm{AP}} = (\mathsf{out}_{\mathcal{G}\oplus\mathcal{N}_{\sigma_0}}(\sigma_A, \emptyset))_{\restriction \mathrm{AP}}$ and so that $\mathcal{G} \oplus \mathcal{N}_{\sigma_0} \models \langle\!\langle A \rangle\!\rangle \psi$. □

Example 9. *The formula from Example 7 is translated into* ATL^*_{sc} *as* $\langle\!\langle \cdot 0 \cdot \rangle\!\rangle \langle\!\langle \cdot 1 \cdot \rangle\!\rangle \mathtt{G}(\mathit{work} \rightarrow \mathtt{X}(\mathit{rest} \wedge \mathtt{X}\mathit{rest}))$*. A possible norm (corresponding to a strategy for 0) is:* $\mathrm{Nrm} = \{\mathsf{q}_0, \mathsf{q}_1, \mathsf{q}_2\}$*, where* $\eta(\mathsf{q}_0, \pi) = \mathsf{q}_0$ *if* $\mathit{work} \notin \pi$*, else* $\eta(\mathsf{q}_0, \pi) = \mathsf{q}_1$*;* $\eta(\mathsf{q}_1, \pi) = \mathsf{q}_2$ *and* $\eta(\mathsf{q}_2, \pi) = \mathsf{q}_0$ *if* $\mathit{rest} \in \pi$*, else* $\eta(\mathsf{q}_1, \pi) = \mathsf{q}_1$ *and* $\eta(\mathsf{q}_2, \pi) = \mathsf{q}_2$*.* $\mathsf{illegal}(\mathsf{q}_0, \pi) = \emptyset$ *and* $\mathsf{illegal}(\mathsf{q}_1, \pi) = \mathsf{illegal}(\mathsf{q}_2, \pi) = \{(2, \mathit{ask})\}$*.*

As a consequence of the lemma, the norm synthesis problem for a MAS $\mathcal{G}$ and an ATL* objective φ can be reduced to the model-checking of an $\mathrm{ATL}^*_{\mathsf{sc}}$ formula over the accessory game $\mathcal{G}'$. The complexity of such a procedure depends on the number h of alternation quantifiers in the formula and the first quantifier modality. More precisely, it is $(h+1)$-EXPTIME for existential ATL*, that is, the fragment of ATL* with formulas starting with an existential quantification, and $(h+2)$-EXPTIME for universal ATL*. The reason for this is that the resulting $\mathrm{ATL}^*_{\mathsf{sc}}$ formula comes with an extra alternation, if the original ATL* formula φ begins with a universal quantification.

Theorem 3. *The following two statements hold:*

1. *Norm Synthesis for a k-MAS for an existential ATL* formula with alternation quantifier h can be solved in $(h+1)$-EXPTIME.*
2. *Norm Synthesis for a k-MAS for a universal ATL* formula with alternation quantifier h can be solved in $(h+2)$-EXPTIME.*

Notice that the two fragments introduced in Section 2.3 are both of alternation 1. In particular, formulas for strategic permission (Cf. Definition 1) are in the existential ATL* fragment, whereas formulas for strategic prohibition (Cf. Definition 2) are in the universal ATL* fragment. Therefore, combining this with the complexity results of Theorem 3, we obtain the following corollary.

[4] Note that we can consider the projection over AP as the formula ψ ranges over the same set of atomic propositions.

Corollary 1. *The following two hold:*

1. *Norm Synthesis for a k-MAS for a strategic permission specification can be solved in 2-EXPTIME.*
2. *Norm Synthesis for a k-MAS for a strategic prohibition specification can be solved in 3-EXPTIME.*

It is worth mentioning a further special case that arises with strategic prohibition specifications, when every agent is mentioned in the universal quantification. In this case, the alternation of the formula is 0, as there is no implicit existential quantification, as in Example 2. For this case, the complexity is 2EXPTIME.

Observe that we can obtain a Norm Synthesis algorithm by the following steps. First, we use Construction 1 to reduce from Norm Synthesis to model checking $\mathrm{ATL}^*_{\mathsf{sc}}$. Note that such translation is both effective and polynomial in the size of the original k-MAS $\mathcal{G}$. Then, we solve the corresponding model checking instance by employing any applicable procedure, returning a normative strategy σ_0, if available. Finally, by employing Construction 2 on σ_0, we obtain a norm for $\mathcal{G}$. Again, the construction is both effective and polynomial in the size of σ_0. Moreover, Lemma 3 and Theorem 3 combined guarantee the procedure to solve the original Norm Synthesis problem correctly.

4 Norm Removal

Regarding Norm Removal, note that the solution space is finite, as it is given by the 2^k possible subsets of norms that are implemented in a k-MAS. Therefore, it suffices to model-check the ATL* formula φ against all possible 2^k removals of norms.

Theorem 4. *The norm removal problem is 2EXPTIME-complete w.r.t. the size of the ATL* formula and EXPTIME w.r.t. the number of norms implemented.*

5 Related Work

There is an extensive literature on using norms of differing types for the formal analysis and design of single- and multi-agent systems. In some approaches, norms correspond to labelling some states in a state transition system as 'violating', e.g., (Meyer and Wieringa 1993). For example, in (Astefanoaei et al. 2009; Dennis, Tinnemeier, and Meyer 2010; Dastani, Grossi, and Meyer 2013), norms are represented by "counts-as" rules characterising violation states (e.g., a state where an agent exceeds the speed limit "counts as" a violation). Other kinds of norms label some transitions as violating, e.g., (Ågotnes, van der Hoek, and Wooldridge 2010), or label particular paths through the system as violating, e.g., (Bulling, Dastani, and Knobbout 2013). For example, conditional prohibitions with deadlines and sanctions (Dignum et al. 2004; Boella and van der Torre 2004; Boella, Broersen, and van der Torre 2008; Tinnemeier et al. 2009) are of the form $P(c, \varphi, d, s)$ where c is the detachment condition (when a state satisfying c is encountered the prohibition becomes active), φ is the state property which is prohibited to bring about, and d is the deadline (after a state satisfying d is encountered, the prohibition is no longer active), and s is a sanction (explained below). Modulo a translation between state-based and action-based prohibitions (Alechina, Dastani, and Logan 2018), regimented conditional prohibitions are clearly a special case of the norms considered in this paper.

What happens as a result of implementing a norm also varies. For example, in some approaches the violating states or transitions are removed (norm regimentation). We can then check whether some desirable properties (objectives) are satisfied in the resulting smaller system, see, e.g., (Ågotnes et al. 2007). In other approaches, violating states or paths are labelled with sanctions (norm enforcement). Sanctions are 'fines' applied to violating traces (in the special case of regimented norms, like the ones we consider in this paper, the trace is terminated). Again, we can check whether it is possible to execute a 'good' behaviour without incurring a sanction, or it is impossible to execute a 'bad' behaviour without incurring a sanction, see, e.g., (Alechina, Dastani, and Logan 2013). However, the aim in such approaches is to *verify* whether some property is true after implementing a norm, rather than to *synthesise* a norm, that, if implemented, will satisfy the property as in our approach.

The first formal treatment of norm synthesis (social laws to coordinate agents' behaviour) was proposed in (Shoham and Tennenholtz 1995), see also (Fitoussi and Tennenholtz 2000). There, norms are constraints on agent behaviours of the form (a, φ), which are interpreted as: in a state satisfying φ action a is prohibited. The objectives are essentially strategy existence properties (or strategic permissions in our terminology). The decision form of the synthesis problem of 'useful social laws' (enabling the objectives) is shown to be NP-complete. In (Christelis and Rovatsos 2009), an EXPTIME algorithm for synthesising state-based prohibitions was proposed. In a somewhat different approach to norm synthesis, (Corapi et al. 2011) use ILP to synthesise norms from use cases. In (Morales et al. 2015; Morales et al. 2018) on-line norm synthesis is proposed as a more feasible way of synthesising norms when the state space is not known in advance. In (van der Hoek, Roberts, and Wooldridge 2007), the problem of synthesising norms is reduced to model-checking in ATL (a fragment of ATL*). In (Bulling and Dastani 2016), norm synthesis is studied in a setting similar to ours, but where agent preferences are known and it is possible to consider Nash equilibria. The system is represented as a CGS. Agents' preferences are represented by a list of pairs (φ_j, u_j), where φ_j is an LTL formula and u_j is a natural number (utility). Nash equilibrium is defined in terms of the utilities obtained by the agents when adopting a given strategy. The system objective is represented by a normative choice function, which is also an LTL formula. Their 'regimenting norms' are closest to norms considered in this paper, and are of the form $(\varphi, \mathcal{A}, \bot)$ where φ is a propositional formula and $\mathcal{A} \subseteq Act^n$ is a set of joint actions. The norm makes $\mathcal{A}$ illegal in states satisfying φ. Rather than removing illegal actions, they are redirected to loop in the same state (have no effect).[5] Clearly, such norms are less expressive than the dynamic norms in our approach. The problems considered in (Bulling and Dastani 2016) 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 iff all Nash equilibria satisfy the formula. Weak implementation is Σ_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 Σ_2^P and is both Σ_2^P-hard and Π_2^P-hard. Weak implementation existence is Σ_2^P-complete. Strong implementation existence is Σ_3^P-complete.

Dynamic norms were introduced in (Huang et al. 2016; Perelli 2019). In (Huang et al. 2016), dynamic absolute or regimented prohibitions similar to the ones in this paper are considered. The illegality function returns a joint action rather than a set of pairs of an action and agent, as in (Bulling and Dastani 2016). The language for specifying objectives is Computation Tree Temporal Logic (CTL). The main result is that the norm synthesis problem is EXPTIME-complete. Other problems considered are two versions of norm recognition problem. In (Perelli 2019), 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. Since our language for objectives is more expressive, it is not surprising that the complexity of synthesis for our setting is higher.

6 Conclusion

We proposed a framework for modelling dynamic norms that enforce constraints on agent strategies in multi-agent systems in order to satisfy system objectives stated in ATL*. These norms are more general than regimented state-based prohibitions and conditional prohibitions, and are *dynamic*, so can constrain strategies in a more flexible way. We prove that the norm synthesis problem is decidable in $(k + 1)$-EXPTIME; however for two important classes of objectives, strategic permissions and strategic prohibitions, it is in 2EXPTIME and 3EXPTIME, respectively. We conjecture that the $(k + 1)$-EXPTIME bound is tight, but leave the proof of hardness for future work. Another possible direction of future research is synthesis of minimally constraining norms.

Acknowledgments

This work is partially supported by the ERC Advanced Grant WhiteMech (No. 834228), by the EU ICT-48 2020 project TAILOR (No. 952215), by the PRIN project RIPER (No. 20203FFYLK), and by the JPMorgan AI Faculty Research Award "Resilience-based Generalized Planning and Strategic Reasoning".

[5]The authors mention that in previous work they removed illegal joint actions entirely. However this caused problems with actions still being available to individual agents, while a joint action was impossible. In our approach, we avoid this problem by making individual rather than joint actions illegal.

References

Ågotnes, T.; van der Hoek, W.; Rodríguez-Aguilar, J. A.; Sierra, C.; and Wooldridge, M. J. 2007. On the logic of normative systems. In Veloso, M. M., ed., *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 1175–1180.

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

Alechina, N.; Dastani, M.; and Logan, B. 2013. Reasoning about normative update. In *Proceedings of the Twenty Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, 20–26. AAAI Press.

Alechina, N.; Dastani, M.; and Logan, B. 2018. Norm specification and verification in multiagent systems. In Chopra, A.; van der Torre, L.; Verhagen, H.; and Villata, S., eds., *Handbook of Normative Multiagent Systems*, 29–55. College Publications.

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time temporal logic. *J. ACM* 49(5):672–713.

Astefanoaei, L.; Dastani, M.; Meyer, J.; and de Boer, F. 2009. On the semantics and verification of normative multiagent systems. *International Journal of Universal Computer Science* 15(13):2629–2652.

Boella, G., and van der Torre, L. 2004. Regulative and constitutive norms in normative multiagent systems. In *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, 255–266.

Boella, G.; Broersen, J.; and van der Torre, L. 2008. Reasoning about constitutive norms, counts-as conditionals, institutions, deadlines and violations. In *Proceedings of the International Conference on Principles and Practice of Multi-Agent Systems (PRIMA)*, 86–97.

Bulling, N., and Dastani, M. 2016. Norm-based mechanism design. *Artif. Intell.* 239:97–142.

Bulling, N.; Dastani, M.; and Knobbout, M. 2013. Monitoring norm violations in multi-agent systems. In *Twelfth International conference on Autonomous Agents and Multi-Agent Systems (AAMAS'13)*, 491–498.

Chopra, A.; van der Torre, L.; Verhagen, H.; and Villata, S., eds. 2018. *Handbook of Normative Multiagent Systems*. College Publications.

Christelis, G., and Rovatsos, M. 2009. Automated norm synthesis in an agent-based planning environment. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems*, 161–168.

Corapi, D.; Russo, A.; Vos, M. D.; Padget, J. A.; and Satoh, K. 2011. Normative design using inductive learning. *Theory Pract. Log. Program.* 11(4-5):783–799.

Dastani, M.; Grossi, D.; and Meyer, J.-J. 2013. A logic for normative multi-agent programs. *Journal of Logic and Computation, special issue on Normative Multiagent Systems* 23(2):335–354.

Dennis, L. A.; Tinnemeier, N. A. M.; and Meyer, J.-J. C. 2010. Model checking normative agent organisations. In Dix, J.; Fisher, M.; and Novák, P., eds., *Computational Logic in Multi-Agent Systems - 10th International Workshop, CLIMA X, Revised Selected and Invited Papers*, volume 6214 of *Lecture Notes in Computer Science*, 64–82. Springer.

Dignum, F.; Broersen, J.; Dignum, V.; and Meyer, J.-J. C. 2004. Meeting the deadline: Why, when and how. In *Proceedings of the International Workshop on Formal Approaches to Agent-Based Systems (FAABS)*, 30–40.

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

Grossi, D.; Aldewereld, H.; and Dignum, F. 2006. *Ubi Lex, Ibi Poena* : Designing norm enforcement in e-institutions. In Noriega, P.; Vázquez-Salceda, J.; Boella, G.; Boissier, O.; Dignum, V.; Fornara, N.; and Matson, E., eds., *Coordination, Organizations, Institutions, and Norms in Agent Systems II - AAMAS 2006 and ECAI 2006 International Workshops, COIN 2006, Revised Selected Papers*, volume 4386 of *Lecture Notes in Computer Science*, 101–114. Springer.

Haynes, C.; Luck, M.; McBurney, P.; Mahmoud, S.; Vitek, T.; and Miles, S. 2017. Engineering the emergence of norms: a review. *The Knowledge Engineering Review* 32.

Huang, X.; Ruan, J.; Chen, Q.; and Su, K. 2016. Normative Multiagent Systems: The Dynamic Generalization. 1123–1129.

Laroussinie, F., and Markey, N. 2015. Augmenting ATL with strategy contexts. *Inf. Comput.* 245:98–123.

Meyer, J.-J. C., and Wieringa, R. J. 1993. Deontic logic: A concise overview. In Meyer, J.-J. C., and Wieringa, R., eds., *Deontic Logic in Computer Science: Normative System Specification*. John Wiley & Sons. 3–16.

Morales, J.; López-Sánchez, M.; Rodríguez-Aguilar, J. A.; Vasconcelos, W. W.; and Wooldridge, M. J. 2015. Online automated synthesis of compact normative systems. *ACM Trans. Auton. Adapt. Syst.* 10(1):2:1–2:33.

Morales, J.; Wooldridge, M. J.; Rodríguez-Aguilar, J. A.; and López-Sánchez, M. 2018. Off-line synthesis of evolutionarily stable normative systems. *Auton. Agents Multi Agent Syst.* 32(5):635–671.

Perelli, G. 2019. Enforcing Equilibria in Multi-Agent Systems. 188–196.

Shoham, Y., and Tennenholtz, M. 1995. On social laws for artificial agent societies: off-line design. *Artificial Intelligence* 73(1-2):231–252.

Tinnemeier, N.; Dastani, M.; Meyer, J.-J. C.; and van der Torre, L. 2009. Programming normative artifacts with declarative obligations and prohibitions. In *Proceedings of the IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, WI-IAT'09*, volume 2, 145 –152.

van der Hoek, W.; Roberts, M.; and Wooldridge, M. J. 2007. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese* 156(1):1–19.