Formal Verification of Parameterised Neural-symbolic Multi-agent Systems

Panagiotis Kouvaros[1] , **Elena Botoeva**[2] and **Cosmo De Bonis-Campbell**[2]
[1]Technology and Innovation School, University of Limassol
[2]School of Computing, University of Kent
pkouvaros@uol.ac.cy, ebotoeva@kent.ac.uk, cd586@kent.ac.uk

Abstract

We study the problem of verifying multi-agent systems composed of arbitrarily many neural-symbolic agents. We introduce a novel parameterised model, where the parameter denotes the number of agents in the system, each homogeneously constructed from an agent template equipped with a neural network-based perception unit and a traditionally programmed action selection mechanism. We define the verification and emergence identification problems for these models against a bounded fragment of CTL. We put forward an abstraction methodology that enables us to recast both problems to the problem of checking Neural Interpreted Systems with a bounded number of agents. We present an implementation and discuss experimental results obtained on a social dilemma game based on guarding.

1 Introduction

Safety concerns stemming from the increasing development of Multi-agent Systems (MAS) have been put under mathematical scrutiny by automated methods that ascertain their correct behaviour. Verification methods based on SAT and BDDs [Kacprzak *et al.*, 2004; Raimondi and Lomuscio, 2005] have resulted in push-button engines such as Verics, MCK and MCMAS [Gammie and van der Meyden, 2004; Kacprzak *et al.*, 2008; Lomuscio *et al.*, 2017]. In conjunction with increasingly sophisticated state-space reduction techniques, such as predicate abstraction [Lomuscio and Michaliszyn, 2015] and partial order reductions [Jamroga *et al.*, 2020], the verifiers have been able to scale to the analysis of systems with very large state spaces.

While the different methods target the provision of effective solutions to different types of analyses, e.g., fast search for counterexamples [Penczek and Lomuscio, 2003] as opposed to fast correctness proofs [Ball and Kupferman, 2006], and for different classes of MAS, e.g., MAS defined over infinite-state as opposed to finite-state variables [Lomuscio and Michaliszyn, 2015], all methods make two fundamental assumptions. The first is that the MAS under analysis is composed of a known number of agents specified at design time. The second is that the agents composing the MAS are specified using traditional programming languages. The approaches cannot therefore be used to verify important classes of MAS, where either the systems have arbitrarily many participants or the agents are endowed with machine learning components. The former class of systems includes open systems, where agents can join and leave the system at runtime, and applications designed irrespective of the number of participants, such as robot swarms and scenarios in the Internet of Things. The latter class comprises forthcoming neural-symbolic applications such as autonomous vehicles.

More recent methods have addressed the verification of systems with an unbounded number of constituents [Kouvaros and Lomuscio, 2016b; Kouvaros and Lomuscio, 2016a]. While the various verification methods focus on different communication primitives for the agents, most of them rely on abstractions whereby the unbounded verification problem is reduced to analysing a finite state-space. These abstractions formed the backbone of various formal reasoners such as those targeting fault-tolerance [Kouvaros *et al.*, 2018] and data-aware systems [Belardinelli *et al.*, 2017].

In a different line of work, verification methods for MAS comprising agents with neural components were put forward [Akintunde *et al.*, 2020b; Akintunde *et al.*, 2022]. To deal with the real-valued operational domain of the neural network models, the methods recast verification queries for bounded properties into Mixed-Integer-Linear-Programming.

While these two lines of work independently tackle unbounded and neural-symbolic MAS, none of the underlying methods can be used for analysis of systems that are both unbounded and neural-symbolic. In this paper we overcome this limitation. Specifically, we introduce *Parameterised Neural-symbOlic interpreted Systems (PANoS)*, a formal model for modelling unbounded neural-symbolic MAS. We develop an abstraction methodology for PANoS whereby we derive novel sound and complete procedures for the verification and emergence identification problems with respect to bounded universal and existential CTL formulae. We use these procedures to analyse a social dilemma scenario.

The rest of the paper is organised as follows. After discussing related work below, we present PANoS in Section 2, followed by the development of the verification and emergence identification procedures in Section 3, which we evaluate in Section 4 on a social dilemma scenario. We conclude in Section 5.

Related Work. The contribution is related to the two lines of work discussed above, namely parameterised verification and verification for neural-symbolic MAS. Previous models in parameterised verification targeted either arbitrarily many agents operating in fixed environments [Kouvaros and Lomuscio, 2016b; Kouvaros and Lomuscio, 2016a; Felli *et al.*, 2020] or a fixed number of agents living in environments of arbitrary size [Aminof *et al.*, 2016]. None of these models include neural components. Systems comprising homogeneous agents were also analysed within the framework of Alternating-time Temporal Logic but in a fixed, non-parameterised and purely symbolic setting [Pedersen and Dyrkolbotn, 2013].

Existing verification methods for MAS with neural components [Akintunde *et al.*, 2020b; Akintunde *et al.*, 2022] take as input systems with a known number of agents. The main theoretical finding of this work is that verification for an unbounded number of agents can be reduced to the verification of (abstract) systems with a bounded number of agents.

2 Parameterised Neural-symbolic Interpreted Systems

Interpreted systems are a standard semantics for describing multi-agent systems [Fagin et al., 1995b]. They provide a natural setup to interpret specifications in a variety of languages, including temporal-epistemic logic and alternating-time temporal logic [Fagin et al., 1995a; Lomuscio and Raimondi, 2006]. Parameterised interpreted systems is a parametric extension of interpreted systems put forward to reason about unbounded multi-agent systems [Kouvaros and Lomuscio, 2015]. The parameter in a system of this kind denotes the number of agents composing the system, each homogeneously constructed from an agent template.

In this section we extend parameterised interpreted systems to Parameterised Neural-symbOlic interpreted Systems (PANoS), where the template for the agents is not purely symbolic but *(i)* comprises a perception mechanism that is implemented via neural networks, *(ii)* it is coupled with a symbolic action mechanism. This neural-symbolic treatment of the agents follows the Neural Interpreted Systems (NIS) model from [Akintunde *et al.*, 2020b]. Differently from PANoS however, NIS are limited to standard non-parametric systems with a pre-defined number of agents.

A PANoS consists of the descriptions of an agent template, from which an unbounded number of concrete agents may be constructed, and of an environment in which the agents operate. The agent template is defined as follows.

Definition 1 (Agent template)**.** *An* agent template *is a tuple* $t = (L_t, \iota_t, obs_t, Act_t, prot_t, tr_t)$*, where:*

- $L_t = Prv_t \times Per_t$ *is a nonempty (possibly infinite) set of template* local states*. Each local state is a pair* (prv_t, per_t) *of a private state* $prv_t \in Prv_t \subseteq \mathbb{R}^{m_{prv}}$ *and a percept* $per_t \in Per_t \subseteq \mathbb{R}^{m_{per}}$ *that encodes a perception of the environment by the agent.*
- $\iota_t \in L_t$ *is the unique* initial template state.
- $obs_t : L_t \times L_e \to Per_t$ *is an* observation function *that maps pairs of template local states* $L_t \subseteq \mathbb{R}^{m_{prv}+m_{per}}$ *and environment states* $L_e \subseteq \mathbb{R}^{m_e}$ *(see below) to percepts* $Per_t \subseteq \mathbb{R}^{m_{per}}$*. We assume, without loss of generality to the parameterised verification results present in the Secton 3, that the observation function is implemented via a neural network* $f : \mathbb{R}^{m_{prv}+m_{per}+m_e} \to \mathbb{R}^{m_{per}}$.
- Act_t *is a nonempty and finite set of* actions.
- $prot_t : L_t \to 2^{Act_t} \setminus \{\emptyset\}$ *is a* local protocol function *that returns the set of actions available at a given local state.*
- $tr_t : L_t \times Act_t \times 2^{Act_t} \times Act_e \to Prv_t$ *is a* local transition function *that determines the next private state for an agent instantiated from the template given its current local state, its action, the set of actions performed by all the agents and the action of the environment.*

Example 1. We consider the example of a guarding game, an instance of a collective risk dilemma [Milinski *et al.*, 2008]. In this game there is a colony of agents that needs to be guarded. Guarding duty costs a guard some health G_r ($G_r < 0$) and resting improves an agent's health by R_r ($R_r > 0$). If no one is guarding, then all agents in the colony lose some health U_r ($U_r < 0$). The maximum value of an agent's health is M_h. If an agent does not have any health left, then it *expires*. We formalise the agent template $t = (L_t, \iota_t, obs_t, Act_t, prot_t, tr_t)$ in this game as follows.

- $L_t = \{(h, per) \mid h \in \{0, \ldots, M_h\},\ per \in \{G_p, R_p, E_p\}$, where h is the health of the agent and percepts G_p, R_p and E_p indicate what the agent is willing to do (see below).
- $\iota_t = (|G_r| + 1, G_p)$, i.e., at the start each agent has enough health to be able to guard at least once without expiring.
- $Act_t = \{G_a, R_a, E_a\}$, where G_a, R_a and E_a stand for *G*uarding, *R*esting and *E*xpired *a*ctions, respectively.
- The observation function is assumed to be given as a neural network. It returns whether the agent volunteers to guard, G_p, can only rest, R_p, or is expired, E_p.
- $prot_t(h, per) = \begin{cases} \{G_a, R_a\}, & \text{if } per = G_p \\ \{R_a\}, & \text{if } per = R_p \\ \{E_a\}, & \text{if } per = E_p \end{cases}$

  If the percept is G_p, then the agent can both guard and rest. If the percept is R_p or E_p, the only available actions are resting and the dummy action E_a, respectively.
- The local transition function updates the health according to the rules. Formally, for $P \in \{G_p, R_p\}$:

  $tr_t((h, P), G_a, A) = \max(h + G_r, 0)$
  $tr_t((h, P), R_a, A) = \min(h + R_r, M_h) \quad \text{if } G_a \in A$
  $tr_t((h, P), R_a, A) = \max(h + U_r, 0) \quad \text{if } G_a \notin A$
  $tr_t((h, E_s), E_a, A) = h.$

The environment has a similar description to an agent template but it *(i)* does not include a perception mechanism, *(ii)* may include an infinite set of initial local states.

Definition 2 (Environment)**.** *An* environment *is a tuple* $e = (L_e, I_e, Act_e, prot_e, tr_e)$*, where* $L_e \subseteq \mathbb{R}^{m_e}$ *is a nonempty set of* local states*,* $I_e \subseteq L_e$ *is a nonempty set of* initial local states*,* Act_e *is a nonempty and finite set of* actions*,* $prot_e\colon L_e \to 2^{Act_e} \setminus \{\emptyset\}$ *is a* local protocol function*, and* $tr_e\colon L_e \times Act_e \times 2^{Act_t} \to L_e$ *is a* local transition function.

The agent template and the environment define the main formal structure we will be using in this paper.

Definition 3. *A* Parameterised Neural Interpreted System (PANoS) *is a tuple $\mathcal{S} = (t, e, \ell)$, where t is an agent template, e is an environment, and $\ell : AP \to 2^{L_t}$ is a labelling function on the agent template states for a set AP of atomic propositions.*

Example 2. *Consider the agent template t in Example 1. Assume a dummy environment e where $L_e = I_e = \{0\}$ and $Act_e = \{0\}$ are singleton states and actions, $prot_e(0) = \{0\}$ and $tr_e(0, 0, A) = 0$ for all $A \in 2^{Act_t}$. Suppose $AP = \{\mathsf{a}, \mathsf{d}\}$, where a stands for alive and d for dead, and ℓ is a labelling function defined as $\ell(\mathsf{a}) = \{(h, S) \mid h > 0$ and $S \neq E_p\}$ and $\ell(\mathsf{d}) = L_t \setminus \ell(\mathsf{a})$. Then $\mathcal{S} = (t, e, \ell)$ is a PANoS for the guarding example.*

A PANoS $\mathcal{S}$ gives a parametric description of an unbounded collection of *concrete* NIS. In particular, for any value $n \geq 1$ of the parameter, the concrete system $\mathcal{S}^{(n)}$ composes n copies $1, \dots, n$ of the agent template with the environment e. We write $Ag^{(n)}$ for the set $\{1, \dots, n\}$ of concrete agents instantiated from t. A *global state* in $\mathcal{S}^{(n)}$ describes the system at a particular instant of time. It is a tuple $q = (l_1, \dots, l_n, l_e)$ of local states for all the agents and the environment in $\mathcal{S}^{(n)}$. For a global state q, we write $lprv_i(q)$, $lper_i(q)$ and $ls_i(q)$ to denote the *private part prv_i* and the *perception part per_i* of the local state $ls_i(q) = (prv_i, per_i)$ of agent i in q. The set $G^{(n)}$ of all possible global states is $G^{(n)} = L_1 \times \cdots \times L_n \times L_e$. A *joint action* $\alpha = (\alpha_1, \dots, \alpha_n, \alpha_e)$ in $\mathcal{S}^{(n)}$ is a tuple of local actions for all the agents and the environment. For a joint action α, we write $la_i(\alpha)$ to denote the *local action* of agent i in α. The set $ACT^{(n)}$ of all joint actions is $ACT^{(n)} = Act_1 \times \cdots \times Act_n \times Act_e$. We now formally define the concrete NIS generated from PANoS.

Definition 4 (Concrete Neural Interpreted System)**.** *Given a PANoS $\mathcal{S} = (t, e, \ell)$ and $n \geq 1$, a* concrete neural interpreted system *is a tuple $\mathcal{S}^{(n)} = \left(\{1, \dots, n, e\}, I^{(n)}, \ell^{(n)}\right)$, where $I^{(n)} = \{(\iota_1, \dots, \iota_n, \iota_e) \mid \iota_e \in I_e\}$ is the set of initial global states and $\ell^{(n)} : AP \times \{1, \dots, n\} \to 2^{G^{(n)}}$ is the concrete labelling function satisfying $q \in \ell^{(n)}(p, i)$ iff $ls_i(q) \in \ell_i(p)$.*

So the atomic propositions in a concrete system are indexed by each of the concrete agents: (p, i) holds in a global state if the agent i is at a local state labelled with p by the template labelling function. This will enable us to construct specifications independently of the size of the concrete system on which they are evaluated.

Given the current global state $(l_1, \dots, l_n, l_e)$, where $l_i = (prv_i, per_i)$, for $i \in Ag^{(n)}$, of the agents and the environment, the operational cycle of the agents is described as follows. First, every agent $i \in Ag^{(n)} \cup \{e\}$ selects an action α_i that is permitted by its protocol, i.e., $\alpha_i \in prot_i(l_i)$. Then, the agents synchronously perform the joint action $\alpha = (\alpha_1, \dots, \alpha_n, \alpha_e)$. Following the execution of the joint action, each agent $i \in Ag^{(n)}$ updates the private component of its local state as per its local transition function to $prv'_i = tr_i(l_i, \alpha_i, \alpha_{\to\{\}}, \alpha_e)$, where $\alpha_{\to\{\}} = \{\alpha_j \mid j \in \{1, \dots, n\}\}$ is the projection of the joint action for the agents onto a set. This generates an intermediate local state $l'_i = (prv'_i, per_i)$ for each of the agents. Similarly, the environment updates its local state to l'_e. Finally, every agent $i \in Ag^{(n)}$ observes the update on its local state and the update on the local state of the environment via its neural perception module obs_i, thus generating a percept $per'_i = obs_i(l'_i, l'_e)$, with which it updates its perception part thereby obtaining a new local state $l''_i = (prv'_i, per'_i)$.

So, differently from the standard treatment of interpreted systems, the transition function of a concrete agent in PANoS does not depend on the joint action performed in the system, but it depends on the local action performed by the agent and on the set of actions performed by all of the agents and the environment. Thus, the identities of the agents are abstracted away in a joint action, thereby reflecting the unbounded nature of PANoS. In other words, whereas a concrete agent can observe which actions were performed in the system at a given time, it cannot observe which agent or how many agents performed each action.

We now formally define the temporal evolution of a concrete system $\mathcal{S}^{(n)}$.

Definition 5 (Global transition function)**.** *The* global transition function $tr^{(n)} : G^{(n)} \times ACT^{(n)} \to G^{(n)}$ *of a concrete system $\mathcal{S}^{(n)}$ satisfies $tr^{(n)}(q, \alpha) = q'$ iff the following hold:*

- *$la_e(\alpha) \in prot_e(ls_e(q))$, $tr_e(ls_e(q), la_e(q), \alpha_{\to\{\}}) = ls_e(q')$; i.e., the environment's action is protocol compliant and its local state is updated as per its local transition function and the actions that were performed in the round.*
- *For all $i \in \{1, \dots, n\}$, we have that $la_i(\alpha) \in prot_i(ls_i(q))$, $tr_i(ls_i(q), la_i(\alpha), \alpha_{\to\{\}}, la_e(\alpha)) = lprv_i(q')$, and $obs_i((lprv_i(q'), lper_i(q)), ls_e(q')) = lper_i(q')$; i.e., the agent's action is protocol compliant, the private part of its local state is updated as per its local transition function and the actions performed at the round, and the perception part of its local state is updated as per its observation function.*

Each concrete system is associated with a temporal model that we will use to interpret our specification language.

Definition 6 (Model)**.** *Given a concrete NIS $\mathcal{S}^{(n)}$, its* (induced) model *$\mathcal{M}_{\mathcal{S}^{(n)}}$ is a tuple $(G^{(n)}, ACT^{(n)}, T^{(n)}, \ell^{(n)})$, where $G(n)$ is the set of global states, $ACT^{(n)}$ is the set of joint actions, $T^{(n)}$ is the global transition relation defined as $(q, \alpha, q') \in T^{(n)}$ iff $tr^{(n)}(q, \alpha) = q'$, and $\ell^{(n)}$ is the labelling function as in Definition 4.*

A *path* in a model $\mathcal{M}_{\mathcal{S}^{(n)}}$ is an infinite sequence of global states and joint actions $q^0\alpha^0q^1\alpha^1\dots$ s.t. $(q^i, \alpha^i, q^{i+1}) \in T^{(n)}$ for all $i \geq 0$. For a path ρ, we write $\rho(i)$ for the i-th global state in ρ. Given a global state q in $\mathcal{M}_{\mathcal{S}^{(n)}}$ we write $\Pi(q)$ for the set of all paths originating from q.

Example 3. *Consider a concrete neural interpreted system $\mathcal{S}^{(2)} = \left(\{1, 2, e\}, I^{(2)}, \ell^{(2)}\right)$ for the agent template in Example 1 and PANoS in Example 2, where $\ell^{(2)}$ is defined from ℓ, $I^{(2)} = \{(\iota_1, \iota_2, 0)\}$ and $\iota_i = (|G_p| + 1, G_p)$. A model $\mathcal{M}_{\mathcal{S}^{(2)}}$ of $\mathcal{S}^{(2)}$ is depicted in Figure 1, where for brevity we omit the environment local state l_e.*

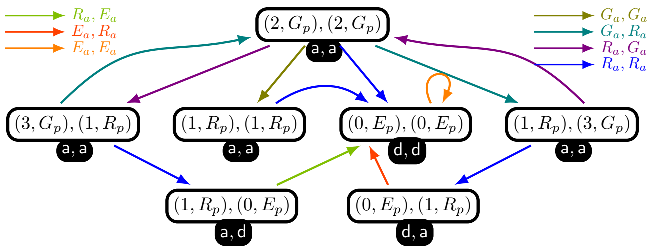

Figure 1: The model $\mathcal{M}_{\mathcal{S}^{(2)}}$ for the guarding example for $G_r = -1$, $R_r = 1$, and $U_r = -2$.

We express specifications for PANoS in an indexed and bounded variant of Computation Tree Logic (CTL), henceforth bICTL. The logic *(i)* introduces indexed atomic propositions that are quantified over the agents of the concrete system on which the formula in question is evaluated, and *(ii)* permits only the construction of formulae whose evaluation can be realised on paths of bounded lengths. The former extends CTL by allowing the formulation of properties irrespective of the concrete system on which they are evaluated. The latter restricts CTL to bounded formulae [Akintunde *et al.*, 2020a].

Definition 7. *Given a set AP of atomic propositions and a set VAR of variables, the* bICTL *formulae are defined by the following BNF:*

$$\varphi ::= (p, v) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid EX^k\varphi \mid AX^k\varphi \mid \forall v\colon \phi,$$

where $p \in AP$, $v \in VAR$ and $k \geq 1$.

The formula $AX^k\varphi$ is read as "for all paths, φ holds at the k-th step", and $EX^k\varphi$ as "there exists a path where φ holds at the k-th step". We note that bounded until $A(\varphi U^k \psi)$ can be inductively abbreviated as in [Akintunde *et al.*, 2020a].

The logic bCTL [Akintunde *et al.*, 2022] is the logic obtained from the above BNF, but replacing the first clause with p and removing the last clause. The restriction of the logic to bounded properties follows the undecidability of verification for the unbounded case [Akintunde *et al.*, 2022]. While the parameterised verification results presented in the next section still hold should negation be allowed in our syntax, verification methods for concrete models, including VENMAS used in the present contribution, do not support negation, since its inclusion may hinder the completeness of verification [Akintunde *et al.*, 2022].

A bICTL formula is said to be a *sentence* if every variable appearing in the formula is in the scope of either a universal or an existential agent quantifier. Hereafter we consider only indexed bICTL sentences of the form $\forall v_1 \cdots \forall v_m\colon \varphi$, or, shortly, $\forall_{v_1,\ldots,v_m}\varphi$, where: *(i)* $v_i \in VAR$ for $i \in \{1, \ldots, m\}$, *(ii)* φ is either *universal* in that it contains only universal path quantifiers or *existential* in that it contains only existential path quantifiers.

We now define the satisfaction relation for bICTL on the temporal models associated with the concrete systems.

Definition 8 (Satisfaction)**.** *Given a model $\mathcal{M}_{\mathcal{S}^{(n)}}$, a global state q in $\mathcal{M}_{\mathcal{S}^{(n)}}$, and a* bICTL *sentence $\forall_{v_1,\ldots,v_m}\varphi$ with $m \leq n$, the* satisfaction *of φ at q, denoted $(\mathcal{M}_{\mathcal{S}^{(n)}}, q) \models \varphi$, or simply $q \models \varphi$ when $\mathcal{M}_{\mathcal{S}^{(n)}}$ is clear from the context, is defined as:*

$q \models (p, i)$ *iff* $q \in \ell^{(n)}((p, i))$, *for* $p \in AP$ *and* $i \in Ag^{(n)}$*;*

$q \models \varphi \vee \psi$ *iff* $q \models \varphi$ *or* $q \models \psi$*;*

$q \models \varphi \wedge \psi$ *iff* $q \models \varphi$ *and* $q \models \psi$*;*

$q \models EX^k\varphi$ *iff there is* $\rho \in \Pi(q)$ *such that* $\rho(k) \models \varphi$*;*

$q \models AX^k\varphi$ *iff for all* $\rho \in \Pi(q)$ *we have that* $\rho(k) \models \varphi$*;*

$q \models \forall_{v_1,\ldots,v_m}\varphi$ *iff for all* $h\colon \{v_1, \ldots, v_m\} \to \{1, \ldots, n\}$, *we have that* $q \models \varphi[v_1 \mapsto h(v_1), \ldots v_m \mapsto h(v_m)]$.

By the definition above, a bICTL sentence is evaluated only up to a bounded number of time steps. Given a bICTL sentence, we call this number of steps the *temporal depth* of the sentence, which we formally define below.

Definition 9. *The* temporal depth $td(\varphi)$ *of a* bICTL *sentence $\forall_{v_1,\ldots,v_m}\varphi$ is inductively defined as follows:*

$$\begin{array}{ll} td(\varphi) = 0 & \text{if} \quad \varphi = (p, i); \\ td(\varphi) = \max(td(\psi_1), td(\psi_2)) & \text{if} \quad \varphi = \psi_1 \vee \psi_2; \\ td(\varphi) = \max(td(\psi_1), td(\psi_2)) & \text{if} \quad \varphi = \psi_1 \wedge \psi_2; \\ td(\varphi) = k + td(\psi_1) & \text{if} \quad \varphi = EX^k\psi_1; \\ td(\varphi) = k + td(\psi_1) & \text{if} \quad \varphi = AX^k\psi_1; \\ td(\varphi) = td(\psi_1) & \text{if} \quad \varphi = \forall_{v1,\ldots,v_m}\psi_1. \end{array}$$

A sentence $\forall_{v_1,\ldots,v_m}\varphi$ is said to *be true* in $\mathcal{M}_{\mathcal{S}^{(n)}}$, denoted $\mathcal{M}_{\mathcal{S}^{(n)}} \models \forall_{v_1,\ldots,v_m}\varphi$, if $(\mathcal{M}_{\mathcal{S}^{(n)}}, q) \models \forall_{v_1,\ldots,v_m}\varphi$ for all $q \in I^{(n)}$. The sentence is said to *be true* in $\mathcal{S}$, denoted $\mathcal{S} \models \forall_{v_1,\ldots,v_m}\varphi$, if it is true in every model induced by every concrete system instantiated from $\mathcal{S}$ with at least m agents, i.e., $\mathcal{M}_{\mathcal{S}^{(n)}} \models \forall_{v_1,\ldots,v_m}\varphi$ for all $n \geq m$. The *parameterised verification problem* is to check whether this holds.

Definition 10 (Parameterised verification problem)**.** *Given a PANoS $\mathcal{S}$ and a* bICTL *sentence $\forall_{v_1,\ldots,v_m}\varphi$, determine whether $\mathcal{S} \models \forall_{v_1,\ldots,v_m}\varphi$.*

While solutions to the parameterised verification problem can be used to establish whether a property holds for any number of agents, some properties, often called *emergent properties* [Bonabeau *et al.*, 1999], are expected to be realised only after certain conditions on the number of agents are met. For instance, foraging protocols for robot swarms, require a certain number of agents to be present for the underlying formations to be established. A natural number $th \in \mathbb{N}$ is an *emergence thershold* for a PANoS $\mathcal{S}$ and a bICTL sentence $\forall_{v_1,\ldots,v_m}\varphi$ if we have that $\mathcal{M}_{\mathcal{S}^{(n)}} \models \forall_{v_1,\ldots,v_m}\varphi$ for every $n \geq th$. The emergence identification problem is to compute an emergence threshold.

Definition 11 (Emergence identification problem)**.** *Given a PANoS $\mathcal{S}$ and a* bICTL *sentence $\forall_{v_1,\ldots,v_m}\varphi$, compute an emergence threshold $th \in \{0, m, m+1, \ldots\}$ for $\mathcal{S}$ and* bICTL. *(If $th = 0$, then $\mathcal{S}$ admits no threshold for $\forall_{v_1,\ldots,v_m}\varphi$.)*

3 Parameterised Verification Procedure

In this section we put forward procedures for solving the parameterised verification and emergence identification problems introduced in the previous section. For ease of presentation, we fix a PANoS $\mathcal{S} = (t, e, \ell)$, where $t = (L_t, \iota_t, obs_t, Act_t, prot_t, tr_t)$, $e =$

$(L_e, I_e, Act_e, prot_e, tr_e)$, and a bICTL sentence $\forall_{v_1,\ldots,v_m}\varphi$ throughout the section.

The verification procedure that we introduce recasts the parameterised verification and emergence identification problems for $\mathcal{S}$ and $\forall_{v_1,\ldots,v_m}\varphi$ to a number of (standard) verification problems for abstract and concrete NIS against the bCTL formula $\varphi[v_1 \mapsto 1, \ldots, v_m \mapsto m]$, which for brevity we denote as $\varphi[m]$. We show that the satisfaction status of $\varphi[m]$ on these systems determines the satisfaction and existence of emergence thresholds for $\forall_{v_1,\ldots,v_m}\varphi$ on $\mathcal{S}$. This enables us to use previously established methodologies for the verification of NIS against bCTL [Akintunde *et al.*, 2020b] to analyse PANoS. We start by reducing the problem of checking the bICTL sentence in question to that of checking $\varphi[m]$.

Lemma 1 (Symmetry reduction). *$\mathcal{S} \models \forall_{v_1,\ldots,v_m}\varphi$ iff $\mathcal{S} \models \varphi[m]$.*

We next construct the *zero-one abstraction* of the systems generated from $\mathcal{S}$. The zero-one abstraction is a NIS comprising a *zero-one agent*, which is an abstraction for arbitrarily many concrete agents, m conrete agents, whose local states determine the satisfaction status of the atomic propositions in $\varphi[m]$ (see Definition 4), and the environment. In other words, the zero-one agent in this abstract NIS encodes how an arbitrary number of agents may interfere with the temporal evolution of the concrete agents $1, \ldots, m$. It is defined in the following.

Definition 12 (Zero-one agent). *Given an agent template $t = (L_t, \iota_t, obs_t, Act_t, prot_t, tr_t)$ over a set Per_t of percepts and a set Prv_t of private states, its associated* zero-one agent *is a tuple $zo = (L_{zo}, \iota_{zo}, obs_{zo}, Act_{zo}, prot_{zo}, tr_{zo})$ over sets $Per_{zo} = 2^{Per_t} \setminus \{\emptyset\}$ and $Prv_{zo} = 2^{Prv_t} \setminus \{\emptyset\}$ of percepts and private states, where:*

- *$L_{zo} = 2^{L_t} \setminus \{\emptyset\}$ is the set of abstract states. An abstract state represents the projection of global states in systems of any size onto a set.*
- *$\iota_{zo} = \{\iota_t\}$ is the unique initial abstract state.*
- *$obs_{zo} : L_{zo} \times L_e \to Per_{zo}$ is the abstract observation function. It maps pairs of abstract and environment states to sets of percepts, where each set includes the percepts that would be collectively generated in a global state represented by the abstract state. Formally, the observation function satisfies $obs_{zo}(l_{zo}, l_e) = per_{zo}$ if:*
  - *for all $l_t \in l_{zo}$ we have that $obs_t(l_t, l_e) \in per_{zo}$;*
  - *for all $per_t \in per_{zo}$ there is $l_t \in l_{zo}$ s.t. $obs_t(l_t, l_e) = per_t$.*
- *$Act_{zo} = 2^{L_t \times Act_t} \setminus \{\emptyset\}$ is the set of abstract actions. As with abstract states, an abstract action represents the projection of joint actions, paired with the local states at which they are performed, of arbitrarily many agents onto a set.*
- *$prot_{zo} \colon L_{zo} \to 2^{Act_{zo}} \setminus \{\emptyset\}$ is the abstract protocol. The protocol prescribes the sets of template actions that can be collectively performed at a global state represented by a given abstract state. It is defined as $prot_{zo}(l_{zo}) = \times_{l_t} \left\{(l_t, A) \mid A \in 2^{prot_t(l_t)} \setminus \{\emptyset\}\right\}$.*
- *$tr_{zo} \colon L_{zo} \times Act_{zo} \times 2^{Act_t} \times Act_e \to Prv_{zo}$ is the abstract transition function. The function determines the set of private states that the agents would collectively transition to in any global state represented by a given abstract state and after they have performed a joint action represented by a given abstract action. It is such that $tr_{zo}(l_{zo}, \alpha_{zo}, A, \alpha_e) = prv_{zo}$ if the following hold:*
  - *$\alpha_{zo} \in prot_{zo}(l_{zo})$;*
  - *for all $(l_t, \alpha_t) \in \alpha_{zo}$, we have $tr_t(l_t, \alpha_t, A, \alpha_e) \in prv_{zo}$;*
  - *for all $prv_t \in prv_{zo}$, there is $(l_t, \alpha_t) \in \alpha_{zo}$ such that $tr_t(l_t, \alpha_t, A, \alpha_e) = prv_t$.*

The abstract NIS comprises the zero-one agent, m concrete agents, and the environment. Formally, it is a tuple $\mathcal{S}_{ab}^{(m)} = \left(\{1, \ldots, m, zo, e\}, I_{ab}^{(m)}, \ell_{ab}^{(m)}\right)$, where $I_{ab}^{(m)} = \{(\iota_1, \ldots, \iota_m, \iota_{ab}, \iota_e) \mid \iota_e \in I_e\}$ is the set of initial global states and $\ell_{ab}^{(m)} \colon AP \times \{1, \ldots, m\} \to 2^{G_{ab}^{(m)}}$ is the abstract labelling function satisfying $q \in \ell_{ab}^{(m)}(p, i)$ iff $ls_i(q) \in \ell_i(p)$. The abstract global transition function is defined as the concrete one but accounting for the zero-one agent:

Definition 13. *The abstract* global transition function $tr_{ab}^{(m)} : G_{ab}^{(m)} \times ACT_{ab}^{(m)} \to G_{ab}^{(m)}$ *of the abstract system $\mathcal{S}_{ab}^{(m)}$ satisfies $tr_{ab}^{(m)}(q, \alpha) = q'$ if the following hold:*

- *$la_e(\alpha) \in prot_e(ls_e(q))$, $tr_e(ls_e(q), la_e(q), \alpha_{\to\{\}}) = q'$, where $\alpha_{\to\{\}} = \{la_i(\alpha) \mid i \in \{1, \ldots, m\}\} \cup \{\alpha_t \mid (l_t, \alpha_t) \in la_{zo}(\alpha) \text{ for some } l_t\}$.*
- *For all $i \in \{1, \ldots, m, zo\}$, we have that $la_i(\alpha) \in prot_i(ls_i(q))$, $tr_i(ls_i(q), la_i(\alpha), \alpha_{\to\{\}}, la_e(\alpha)) = lprv_i(q')$ and $obs_i((lprv_i(q'), lper_i(q)), ls_e(q')) = lper_i(q')$.*

Given the abstract global transition function we can associate an (abstract) model $\mathcal{M}_{\mathcal{S}_{ab}^{(m)}} = \left(G_{ab}^{(m)}, ACT_{ab}^{(m)}, T_{ab}^{(m)}, \ell_{ab}^{(m)}\right)$ to the abstract NIS in a similar manner to the concrete case.

We now establish a correspondence between the abstract model and the concrete models. We show in particular that: *(i)* the abstract model simulates every concrete model with at least $m + 1$ agents; *(ii)* there is always a concrete model with a sufficient number of agents that simulates the abstract model; *(iii)* a concrete model always simulates a smaller concrete model. A model simulates another model if every behaviour exhibited by the latter is also admitted by the former. As specifications for PANoS are bounded, we consider simulation up to a bounded number of steps as defined below.

Definition 14 (Bounded simulation.). *A* b-bounded simulation *between two models $\mathcal{M} = (G, ACT, T, \ell)$ and $\mathcal{M}' = (G', ACT', T', \ell')$ with sets of initial global states I and I' is inductively defined on $b \geq 0$ as follows.*

- *A relation $\sim_0 \subseteq G \times G'$ is 0-bounded simulation if for every $\iota \in I$, there is $\iota' \in I'$ with $(\iota, \iota') \in \sim_0$, and whenever $(q, q') \in \sim_0$, we have that $q \in \ell(p, i)$ implies that $q' \in \ell'(p, i)$.*
- *A relation $\sim_b \subseteq G \times G'$ is b-bounded simulation if for every $\iota \in I$, there is $\iota' \in I'$ with $(\iota, \iota') \in \sim_b$, and whenever $(q, q') \in \sim_b$, the following hold:*

1. $(q, q') \in \sim_0$.

2. If $(q, \alpha, q^1) \in T$ *for a joint action* $\alpha \in ACT$ *and global state* $q^1 \in G$*, then there is a joint action* $\alpha' \in ACT'$ *and global state* $q'^1 \in G'$ *such that* $(q', \alpha', q'^1) \in T'$ *and* $(q^1, q'^1) \in \sim_{b-1}$.

We say that a model $\mathcal{M}'$ *simulates* a model $\mathcal{M}$ up to b time steps, denoted $\mathcal{M} \leq_b \mathcal{M}'$, if there is a b-bounded simulation relation between $\mathcal{M}$ and $\mathcal{M}'$. Universal bCTL formulae are preserved from the simulating model to the simulated model and existential bCTL formulae are preserved from the simulated model to the simulating model whenever their temporal depth is at most b.

Theorem 1. *Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be two models such that* $\mathcal{M} \leq_b \mathcal{M}'$*. Then, the following hold.*

1. If $\mathcal{M}' \models \varphi$ *for a universal* bCTL *formula* φ *with* $td(\varphi) \leq b$*, then* $\mathcal{M} \models \varphi$*.*

2. If $\mathcal{M} \models \varphi$ *for an existential* bCTL *formula* φ *with* $td(\varphi) \leq b$*, then* $\mathcal{M}' \models \varphi$*.*

We can now show the simulation results pertaining to the abstract and concrete models. We start by showing that the abstract model simulates every concrete model with at least $m + 1$ agents up to any depth.

Theorem 2. *Let* $n \geq m + 1$ *and* $b \geq 0$*. Then,* $\mathcal{M}_{\mathcal{S}^{(n)}} \leq_b \mathcal{M}_{\mathcal{S}^{(m)}_{ab}}$.

Next, we show that irrespective of the temporal depth of the specification under analysis there is always a concrete model that simulates the abstract model up to that depth.

Theorem 3. *Given* $b \geq 0$*, there is* $n \geq m + 1$ *such that* $\mathcal{M}_{\mathcal{S}^{(m)}_{ab}} \leq_b \mathcal{M}_{\mathcal{S}^{(n)}}$.

Finally, we show that every concrete model simulates every smaller concrete model up to any depth.

Theorem 4. *Let* $n \geq m$*,* $n' > n$ *and* $b \geq 0$*. Then,* $\mathcal{M}_{\mathcal{S}^{(n)}} \leq_b \mathcal{M}_{\mathcal{S}^{(n')}}$.

The above results enable the derivation of procedures for solving the parameterised verification and emergence identification problems. In the case of universal formulae, the emergence identification procedure simply concerns checking the abstract model, and the parameterised verification procedure additionally involves checking a single concrete model against the formula in question.

Corollary 1. *For a universal* bCTL *formula* $\varphi[m]$*:*

- *If* $\mathcal{M}_{\mathcal{S}^{(m)}_{ab}} \models \varphi[m]$*, then* $m + 1$ *is an emergence threshold for* $\forall_{v_1,\ldots,v_m}\varphi$*. Otherwise, there is no emergence threshold for* $\forall_{v_1,\ldots,v_m}\varphi$*.*
- $\mathcal{M}_{\mathcal{S}^{(m)}_{ab}} \models \varphi[m]$ *and* $\mathcal{M}_{\mathcal{S}^{(m)}} \models \varphi[m]$ *iff* $\mathcal{S} \models \forall_{v_1,\ldots,v_m}\varphi$*.*

Theorems 2, 3 and 4, additionally enable the derivation of procedures for the verification of existential properties.

Corollary 2. *If* $\varphi[m]$ *is an existential* bCTL *formula with temporal depth* b *and* $n = \min_i \left(\mathcal{M}_{\mathcal{S}^{(m)}_{ab}} \leq_b \mathcal{M}_{\mathcal{S}^{(i)}}\right)$*, then:*

- *If* $\mathcal{M}_{\mathcal{S}^{(n)}} \models \varphi[m]$*, then* n *is an emergence threshold for* $\forall_{v_1,\ldots,v_m}\varphi$*. Otherwise, there is no emergence threshold for* $\forall_{v_1,\ldots,v_m}\varphi$*.*
- $\mathcal{M}_{\mathcal{S}^{(i)}} \models \forall_{v_1,\ldots,v_m}\varphi$*, for all* $i \in \{m, \ldots, n\}$*, iff* $\mathcal{S} \models \forall_{v_1,\ldots,v_m}\varphi$*.*

In summary, Corollaries 1 and 2 provide constructive, sound and complete methodologies for checking universal and existential bICTL formulae for PANoS. For the case of universal properties, verification can be conducted by constructing and checking the abstract model and the concrete model with m agents, where m is the number of index variables present in the specification in question. The specification is satisfied by the abstract and concrete models if and only if the specification is satisfied in general for any number of agents. The satisfaction of the specification by the abstract model is also connected by biconditional implication with the existence of an emergence threshold for the specification. For the case of existential properties, verification can be performed by enumerating all concrete models, identifying the smallest one that simulates the abstract model, and checking all concrete models up to the simulating one. The specification is satisfied by all these concrete models if and only if the specification is satisfied in general for any number of agents. The satisfaction of the specification by the concrete model that simulates the abstract model is also connected by biconditional implication with the existence of an emergence threshold for the specification.

4 Evaluation

In this section we present an evaluation of the parameterised verification procedures described in Section 3 on the guarding game presented in Example 1.

The guarding game is an instance of a social dilemma game, a class of MAS characterised by tension between individual and collective rationality [Van Lange *et al.*, 2013]. The game simulates the fundamental forces of a *collective risk dilemma (CRD)*, a type of social dilemma where a guaranteed "tragedy of the commons" [Hardin, 1968] is avoided by personal sacrifice by a population of agents, or brought on by free riding if all the agents neglect the collective interests [Santos and Pacheco, 2011]. Namely, guarding can be considered equivalent to cooperation (acting in collective interest), and resting to defection (acting in selfish interest).

We train a neural observation function using deep Q-learning, a type of reinforcement learning (RL) algorithm. During the training, the game was played by 4 agents, and the parameters were set as $M_h = 4$, $G_r = -2$, $R_r = 1$ and $U_r = -3$. The rewards were assigned to reflect the tension between individual and collective interests. All agents shared the same neural network, and thus were learning to play against exact copies of themselves. The produced neural network has two hidden layers of four ReLU activated neurons, takes as input a single neuron, representing the normalised health points of the agent, and outputs the estimated Q-values of the two actions 'rest' and 'guard'.

Given the learned neural network, we implemented a template agent and a zero-one agent for the guarding game. We then used Corollaries 1 and 2 to verify whether it is possible for a colony of agents to survive after a number of time steps. Specifically, recall from Example 2 that proposition a labels all states with positive health ("alive") and proposition d la-

	$k=2$ ($n=2$)	$k=3$ ($n=3$)	$k=4$ ($n=3$)	$k=5$ ($n=3$)
$i=2$	0.09s	1.46s	5.49s	61.47s
$i=3$	0.13s	0.30s	0.52s	133.28s
$i=4$	0.53s	1.19s	2.31s	4.28s
$i=5$	1.15s	3.41s	17.74s	95.83s
$i=6$	5.09s	17.58s	–	–

Table 1: Verification times for $\mathcal{M}_{\mathcal{S}^{(i)}} \models \varphi_E^k[2]$ for various k and i. For each k, we indicate the value of n from Corollary 2. Grey cells indicate when the property was not satisfied. Dashes indicate a 1 hour timeout.

bels all other states with no health ("dead"). We considered two specifications (for $v_i \in VAR$):

1. The existential property $\forall_{v_1,v_2}\varphi_E^k$, where $\varphi_E^k = EX^k((\mathsf{a}, v_1) \wedge (\mathsf{a}, v_2))$. The property expresses that there is an evolution where at least 2 agents are alive after k time steps.
2. The universal property $\forall_{v_1,\ldots,v_m}\varphi_A^k$, where $\varphi_A^k = AX^k \bigwedge_{i=1}^m (\mathsf{a}, v_i)$. The property expresses that in every possible evolution at least m agents are alive after k time steps, for $m \in \{2, 3\}$.

We used the VENMAS toolkit [Akintunde *et al.*, 2020b] for checking the conrete and abstract systems prescribed by Corollaries 1 and 2. The experiments were performed on a standard PC running Ubuntu 22.04 with 16GB RAM and processor Intel(R) Core i5-4460. We relied on Gurobi v10.0 [Gu *et al.*, 2016] to solve the mixed integer linear programs generated by VENMAS.

Let $\mathcal{S}$ be the PANoS for the guarding example. For the existential property, we check whether at least two agents can stay alive for k time steps ($\mathcal{S} \models \varphi_E^k$), and whether there is a minimal number (an emergence threshold) of agents that can guarantee that. We vary the temporal depth k from 2 to 5. To verify whether $\mathcal{S} \models \forall_{v_1,v_2}\varphi_E^k$, we use Corollary 2 and check whether $\mathcal{M}_{\mathcal{S}^{(i)}} \models \varphi_E^k[2]$ for every $i \in \{2, \ldots, n\}$, where n is the minimal i such that $\mathcal{M}_{\mathcal{S}^{(i)}}$ k-bounded simulates $\mathcal{M}_{\mathcal{S}_{ab}^{(2)}}$. We have that $n = 2$ for $k = 2$, and $n = 3$ for $k \geq 3$.

Table 1 presents the outcomes of the verification queries $\mathcal{M}_{\mathcal{S}^{(i)}} \models \varphi_E^k[2]$ for $i \in \{2, \ldots, 4\}$. For $k = 2$, since $\mathcal{M}_{\mathcal{S}^{(2)}} \models \varphi_E^2[2]$, we conclude that $\mathcal{S} \models \forall_{v_1,v_2}\varphi_E^2$. This of course additionally implies that $n = 2$ is an emergence threshold for $\forall_{v_1,v_2}\varphi_E^2$. For $k \geq 3$, since $\mathcal{M}_{\mathcal{S}^{(2)}} \not\models \varphi_E^k[2]$, we conclude that $\mathcal{S} \not\models \forall_{v_1,v_2}\varphi_E^k$. Still, since $\mathcal{M}_{\mathcal{S}^{(3)}} \models \varphi_E^k[2]$, we obtain that $n = 3$ is an emergence threshold for $\forall_{v_1,v_2}\varphi_E^k$, so there need to be at least 3 agents present in the colony to ensure a temporal evolution whereby the colony is viable. The verification results for $i \in \{4, 5, 6\}$ reported by the table (which are not used to reason about parameterised verification) demonstrate the increasing computational cost of verifying concrete systems for increasing number of agents, thereby empirically justifying the need for parameterised verification.

Concerning the universal property, we verified $\varphi_A^k[m]$ against the abstract model $\mathcal{M}_{\mathcal{S}_{ab}^{(m)}}$ for the temporal depths $k \in \{1, \ldots, 6\}$ and $m \in \{2, 3\}$. The verification times and results can be found in Table 2. We observe that because of the presence of the zero-one agent, the verification times are longer than the ones observed in Table 1. We additionally notice that the property in question is not satisfied even after 1 time step, which is expected given that there exist paths where no agent is guarding even when there are volunteers.

m	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$
2	0.66s	3.60s	–	16.90s	20.59s	–
3	1.75s	16.03s	–	42.74s	2196.70s	–

Table 2: Verification times for $\mathcal{M}_{\mathcal{S}_{ab}^{(m)}} \models \varphi_A^k[m]$ for various m and k. Grey cells indicate when the property was not satisfied. Dashes indicate a 1 hour timeout.

In summary, our experimental results confirm the intractability of verification for concrete models as the number of agents grows, thereby motivating the need for the parameterised verification methods that we put forward.

5 Conclusions

Advances in interconnectivity of autonomous services and machine learning fuel the development of MAS with arbitrarily many neural-symbolic components, thereby creating a pressing need for their verification.

Towards addressing this need, in this paper we put forward a number of automated procedures for the formal analysis of parameterised, neural-symbolic MAS. The procedures enable conclusions to be drawn on the satisfaction of temporal properties irrespective of the number of agents composing the MAS. They can additionally identify emergence thresholds expressing sufficient conditions on the number agents for a property to be realised.

The theoretical results have driven the implementation of a parameterised, neural-symbolic verifier, which we used to reason about a simple social dilemma game. More generally, the techniques here developed can be used to analyse properties of policies learned to deal with real-life challenges that come in the form of collective risk dilemmas, as well as properties in swarm scenarios and open systems in general.

In future work we target the development of parameterised methods for interleaved semantics for neural-symbolic MAS and strategic properties.

References

[Akintunde *et al.*, 2020a] M. Akintunde, E. Botoeva, P. Kouvaros, and A. Lomuscio. Formal verification of neural agents in non-deterministic environments. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS20)*, page To Appear. ACM, 2020.

[Akintunde *et al.*, 2020b] M. Akintunde, E. Botoeva, P. Kouvaros, and A. Lomuscio. Verifying strategic abilities of neural-symbolic multi-agent systems. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR20)*, pages 22–32. AAAI Press, 2020.

[Akintunde *et al.*, 2022] M. Akintunde, E. Botoeva, P. Kouvaros, and A. Lomuscio. Formal verification of neural agents in non-deterministic environments. *Journal of Autonomous Agents and Multi-Agent Systems*, 36(1), 2022.

[Aminof *et al.*, 2016] B. Aminof, A. Murano, S. Rubin, and F. Zuleger. Automatic verification of multi-agent systems in parameterised grid-environments. In *Proceedings of the 2016 international conference on autonomous agents & multiagent systems*, pages 1190–1199, 2016.

[Ball and Kupferman, 2006] Thomas Ball and Orna Kupferman. An abstraction-refinement framework for multi-agent systems. In *21st Annual IEEE Symposium on Logic in Computer Science (LICS'06)*, pages 379–388. IEEE, 2006.

[Belardinelli *et al.*, 2017] F. Belardinelli, P. Kouvaros, and A. Lomuscio. Parameterised verification of data-aware multi-agent systems. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI17)*, pages 98–104. AAAI Press, 2017.

[Bonabeau *et al.*, 1999] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm intelligence*. Oxford University Press, 1999.

[Felli *et al.*, 2020] P. Felli, A. Gianola, and M. Montali. Smt-based safety verification of parameterised multi-agent systems. *arXiv preprint arXiv:2008.04774*, 2020.

[Gammie and van der Meyden, 2004] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of 16th International Conference on Computer Aided Verification (CAV04)*, volume 3114 of *Lecture Notes in Computer Science*, pages 479–483. Springer, 2004.

[Gu *et al.*, 2016] Z. Gu, E. Rothberg, and R. Bixby. Gurobi optimizer reference manual. http://www.gurobi.com, 2016.

[Hardin, 1968] Garrett Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968.

[Jamroga *et al.*, 2020] Wojciech Jamroga, Wojciech Penczek, Teofil Sidoruk, Piotr Dembiński, and Antoni Mazurkiewicz. Towards partial order reductions for strategic ability. *Journal of Artificial Intelligence Research*, 68:817–850, 2020.

[Kacprzak *et al.*, 2004] M. Kacprzak, A. Lomuscio, and W. Penczek. Verification of multiagent systems via unbounded model checking. In *Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS04)*, pages 638–645. ACM, 2004.

[Kacprzak *et al.*, 2008] M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Woźna, and A. Zbrzezny. VerICS 2007 - a model checker for knowledge and real-time. *Fundamenta Informaticae*, 85(1):313–328, 2008.

[Kouvaros and Lomuscio, 2015] P. Kouvaros and A. Lomuscio. Verifying emergent properties of swarms. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI15)*, pages 1083–1089. AAAI Press, 2015.

[Kouvaros and Lomuscio, 2016a] P. Kouvaros and A. Lomuscio. Parameterised model checking for alternating-time temporal logic. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI16)*, pages 1230–1238. IOS Press, 2016.

[Kouvaros and Lomuscio, 2016b] P. Kouvaros and A. Lomuscio. Parameterised verification for multi-agent systems. *Artificial Intelligence*, 234:152–189, 2016.

[Kouvaros *et al.*, 2018] P. Kouvaros, A. Lomuscio, and E. Pirovano. Symbolic synthesis of fault-tolerance ratios in parameterised multi-agent systems. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI18)*, pages 324–330. AAAI Press, 2018.

[Lomuscio and Michaliszyn, 2015] Alessio Lomuscio and Jakub Michaliszyn. Verifying multi-agent systems by model checking three-valued abstractions. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 189–198, 2015.

[Lomuscio *et al.*, 2017] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A model checker for the verification of multi-agent systems. *Software Tools for Technology Transfer*, 19(1):9–30, 2017.

[Milinski *et al.*, 2008] Manfred Milinski, Ralf D. Sommerfeld, Hans-Jürgen Krambeck, Floyd A. Reed, and Jochem Marotzke. The collective-risk social dilemma and the prevention of simulated dangerous climate change. *Proceedings of the National Academy of Sciences*, 105(7):2291–2294, 2008.

[Pedersen and Dyrkolbotn, 2013] T. Pedersen and SK. Dyrkolbotn. Agents homogeneous: A procedurally anonymous semantics characterizing the homogeneous fragment of atl. In *International Conference on Principles and Practice of Multi-Agent Systems*, pages 245–259. Springer, 2013.

[Penczek and Lomuscio, 2003] Wojciech Penczek and Alessio Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 209–216, 2003.

[Raimondi and Lomuscio, 2005] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via OBDDs. *Journal of Applied Logic*, 5(2):235–251, 2005.

[Santos and Pacheco, 2011] Francisco C. Santos and Jorge M. Pacheco. Risk of collective failure provides an escape from the tragedy of the commons. *Proceedings of the National Academy of Sciences*, 108(26):10421–10425, 2011.

[Van Lange *et al.*, 2013] Paul A.M. Van Lange, Jeff Joireman, Craig D. Parks, and Eric Van Dijk. The psychology of social dilemmas: A review. *Organizational Behavior and Human Decision Processes*, 120(2):125–141, 2013. Social Dilemmas.