Towards Formal Verification of Neuro-symbolic Multi-agent Systems

Panagiotis Kouvaros
Department of Computing, Imperial College London
p.kouvaros@imperial.ac.uk

Abstract

This paper outlines some of the key methods we developed towards the formal verification of multi-agent systems, covering both symbolic and connectionist systems. It discusses logic-based methods for the verification of unbounded multi-agent systems (i.e., systems composed of an arbitrary number of homogeneous agents, e.g., robot swarms), optimisation approaches for establishing the robustness of neural network models, and methods for analysing properties of neuro-symbolic multi-agent systems.

1 Introduction

Significant advances in Artificial Intelligence (AI) have enabled the automation of challenging tasks, such as computer vision, that have been traditionally difficult to tackle using classical approaches. This accelerated the trend of incorporating AI components in diverse applications with high societal impact, such as healthcare and transportation. Still, even though there is an increasing consensus in AI being beneficial for society, its inherent fragility hinders its adoption in safety-critical applications. In response to these concerns the area of formal verification of AI has grown rapidly over the past few years to provide methods to automatically verify that AI systems robustly behave as intended.

The formal verification problem is concerned with establishing whether a MAS S satisfies a safety property P. *Model checking*, a key method for the formal verification of reactive systems, has also been used in the past fifteen years to provide automated solutions to this problem [Clarke *et al.*, 1999]. In model checking, the system is represented as a model M_S, the specification is encoded as formula φ and it is then checked whether M_S satisfies φ. In the case of MAS, the formula φ, does not simply express temporal properties of systems, as in reactive systems, but it also denotes high-level attitutes of agency, such as knowledge and strategies, as these are expressed in temporal-epistemic logic [Fagin *et al.*, 1995] and alternating-time logic [Alur *et al.*, 1998].

A number of methods were put forward in the area that enabled the computation of the model checking query for progressively bigger systems. These include binary decision diagrams [Gammie and Meyden, 2004; Lomuscio et al., 2009], abstraction [Co- hen et al., 2009], partial order reduction [Lomuscio et al., 2010] and bounded model checking [Lomuscio et al., 2007].

Even though the methods have enabled to model checking of complex systems of very large state spaces, model checking limits the formal verification of MAS to (i) systems with a known number of participants at design time; (ii) systems with purely symbolic components. This is in contrast to the current trend of developing and deploying MAS with an unbounded number of participants, as in robot swarms, multiparty negotiation protocols and auctions, voting protocols and e-sevices.

Unbounded Multi-agent Systems (UMAS) are MAS composed of homogeneous agents, each instantiated by a unique *agent template*, whose number in not known at design time.

In contrast to traditional models of agency, where the agent's behaviour is given in an agent-based programming language, these methods do not accounts for the recent shift to synthesise the agents' behaviour from data.

the methods cannot in principle completely overcome the *state-space explosion problem*, a key limitation of model checking whereby the state-space is exponential in the number of variables encoding the system to be checked.

State space explosion problem

Purely symbolic agents.

In these cases, one could encode a system with a given number of agents and verify that a specification holds. However, additional agents may possibly interfere with the system in unpredictable ways resulting in the specifications being violated. Therefore, to fully verify the system, the process would have to be repeated for any possible number of components

2 Unbounded Multi-agent Systems

Interpreted systems is a main semantical structure for the formal description of multi-agent systems and the interpretation of agent-based specifications, including those expressed in temporal-epistemic logic and alternate time logic [Fagin *et al.*, 1995; Lomuscio and Raimondi, 2006]. Parameterised Interpreted Systems (PIS) is an extension to interpreted systems that we put forward to reason about the temporal-epistemic properties of UMAS in both synchronous [Kouvaros and Lomuscio, 2015b] and asynchronous [Kouvaros and Lomuscio, 2016b] settings. The parameter in PIS denotes the number of agents in the system, each homogeneously constructed from an agent template.

The verification problem for PIIS (generally known as the *parameterised verification problem* in the reactive systems' literature [Bloem *et al.*, 2015] is to check whether any system, for any value of the parameter, satisfies a given specification. This is in general undecidable [Kouvaros and Lomuscio, 2016b]. General solutions can thus be given only in the form of incomplete techniques. Alternatively, decidable fragments of the problem can be curved by imposing restrictions on the systems and/or the specifications.

A key notion that enables the construction of verification methods in both settings is that of a *cutoff*. A cutoff is a natural number that expresses the number of components that is sufficient to analyse when evaluating a given specification. In other words, if a cutoff can be computed, then the verification problem can be solved by checking all systems whose number of agents is below the cutoff value. In addition to providing solutions to the verification problem, the identification of cutoffs can also be used to check whether the underlying system exhibits a certain *emergent behaviour* (i.e., a behaviour that is realised only when certain lower bounds on the number of agents are met) of interest.

Although in theory cutoffs do not always exist [Kouvaros and Lomuscio, 2013b], strong empirical evidence supports their existence for real-world systems [Emerson and Kahlon, 2000; Emerson and Namjoshi, 1995; Aminof *et al.*, 2014]. For the cases where they do not exist, theoretical analyses show that these often concern impractical cyclic behaviours whose number of repetitions depends on the exact number of agents in the system [Kouvaros and Lomuscio, 2013b].

We have analysed various sufficient conditions for the identification of cutoffs with respect to different synchronisation primitives endowing the agents. In the fully synchronous setting, we have shown that cutoffs can always be identified and gave a procedure for their computation [Kouvaros and Lomuscio, 2015b]. In the asynchronous case, where agents communicate via broadcast actions, we have similarly given a sound and complete technique for their identification [Kouvaros and Lomuscio, 2013a]. For the instances where the agents can additionally participate in pairwise communication with their environment, we have shown that if

(i) the environment can never block pairwise synchronisations for the system of one agent only, and

(ii) each synchronisation can happen in unique configurations for the environment,

then cutoffs can be computed in an efficient procedure that runs in linear time in the size of the agent template [Kouvaros and Lomuscio, 2013b]. The second restriction can be lifted in a cutoff procedure that runs in exponential time [Kouvaros and Lomuscio, 2015a].

While the results were drawn with respect to *homogeneous* UMAS, where every agent is instantiated from a unique agent template, we have also provided extensions that account for *heterogeneous* UMAS, where agents can assume different roles and responsibilities, e.g., heterogeneous robot swarms [Kouvaros and Lomuscio, 2016b]. The heterogeneous semantics allow for broadcast actions that may either concern all agents of all agent templates or all agents following a certain template. Additionally, they enable pairwise interactions between agents of different roles, thereby surpassing the expressive power of the homogeneous model.

Further gains in the expressivity of protocols that can be verified have been obtained by the verification method we introduced for UMAS programmed using variables with infinite domains [Kouvaros and Lomuscio, 2017a]. The method combines predicate abstraction [Lomuscio and Michaliszyn, 2015] with parameterised verification (the former addressing the unboundedness of the state-space of the agents and the latter tackling the unboundedness of their number). Other extensions of PIS have been used to describe *open MAS*, where countably many agents can join and leave the system at runtime. We have given verification methods for open MAS for both synchronous and asynchronous semantics [Kouvaros *et al.*, 2019].

We have released the open-source parameterised verification toolkit MCMAS-P implementing the aforementioned cutoff procedures. MCMAS-P enabled for the first time the verification of aggregation and foraging algorithms for robot swarms irrespective of the number of robots composing the swarm [Kouvaros and Lomuscio, 2015b; Kouvaros and Lomuscio, 2016b]. Further applications included the analysis of the security of an unbounded number of concurrent sessions of cryptographic protocols, for which we provided a mapping from a Dolev-Yao threat model to PIS [Boureanu *et al.*, 2016]. Others concerned the verification of UMAS with *data-aware agents*, i.e., agents that are endowed with possibly infinite domains and that interact with an environment composed of (semi)-structured data [Montali *et al.*, 2014]. For this class of UMAS we similarly gave a mapping to PIS [Belardinelli *et al.*, 2017]. Finally, adaptations of the counter-abstraction methods for PIS enabled us to derive methods for the verification of opinion formation protocols in swarms, which we used to give formal guarantees on the outcome of consensus protocols. Other adaptations facilitated the verification of strategic properties of UMAS expressed in a parameterised variant of Alternating-time temporal logic [Alur *et al.*, 1998] that we introduced [Kouvaros and Lomuscio, 2016a].

We conclude this section by noting that complementary to protocol correctness, which the aforementioned cutoffs methods can formally ascertain, the evaluation of protocols also requires analyses of the extent to which they are are resilient to adverse functioning behaviours for some of the agents in the system. For instance, when evaluating a robot swarm search-and-rescue scenario, it is not sufficient to establish that the swarm will collectively cover the search area, but it is also crucial to determine that local faults, e..g, hardware malfunctions, will be tolerated by the swarm, instead of being propagated through agent interactions thereby dis-coordinating the search. To address this concern we have put forward an automated procedure to establish the robustness of UMAS against a given ratio of faulty to non-faulty agents in the system [Kouvaros and Lomuscio, 2017b], which we followed by a symbolic method to automatically synthesise the maximum ratio of faulty to non-faulty agents before the robustness of UMAS is violated [Kouvaros *et al.*, 2018].

3 Neuro-symbolic Multi-agent Systems

To account for the recent shift to synthesise agents from data, instead of agent-based programming languages, we have introduced a novel formalisation of MAS, which we call *Neuro-symbolic MAS* (NMAS). In a nutshell, an agent in NMAS, comprises a perception mechanism implement via ReLU-based neural networks, coupled with a symbolic action mechanism.

The neural network components, which endow the agents with infinite domains, pose significant challenges to the verification of NMAS. In particular, in contrast to traditional verification for symbolic multiagent systems, where atomic formulae are evaluated in constant time at symbolic states of the system, the evaluation of atomic formulae in NMAS concerns the computation of the output regions of the neural networks for a (potentially infinite) set of inputs. This atomic check is an NP-complete problem [Katz *et al.*, 2017]. It has received considerable attention in the past five years in the context of standalone neural-network models as a principled way of analysing the inherent fragility of neural networks to adversarial attacks.

We first discuss our work on verifying standalone neural networks and then our studies on analysing NMAS.

Neural network verification. Definition. Perhaps the most significant instantiation of the problem is adversarial rob. It is an NP-complete problem [Katz *et al.*, 2017]. While progress has been made, a key difficulty in the area is scalability. Simply put, the present methods, while effective on small models, can- not presently analyse the networks used in vision and other complex tasks. The approaches that have driven this effort are complete and incomplete. ormal verification of neural networks comprises complete and incomplete methods. Complete methods can in principle return a definite answer as to whether the verification prop- erty is satisfied, whereas incomplete methods may be un- able to decide whether the property is satisfied. Complete methods are based on MILP formulations [Botoeva et al., 2020; Bastani et al., 2016; Lomuscio and Maganti, 2017; Cheng et al., 2017; Fischetti and Jo, 2018; Tjeng et al., 2019], SMT encodings [Ehlers, 2017; Katz et al., 2017; Katz et al., 2019], and input refinement [Wang et al., 2018; P. Henriksen, 2020]. Incomplete methods are based on du- ality [Dvijotham et al., 2018; Wong and Kolter, 2018], lin- ear approximations [Tran et al., 2020; Singh et al., 2019; Weng et al., 2018; Tjandraatmadja et al., 2020] and semi- definite relaxations [Fazlyab et al., 2020; Dathathri et al., 2020]. While incomplete approaches differ, they all rely on approximations of the ReLU function. This often improves their scalability over complete methods but can also hinder their efficacy to solve the verification problem.

Improving scalability in complete verification and precision in incomplete verification. Introduced the novel notion of a dependency between nodes which we used to refuced the configurations of the operations of the nodes in the network that need to be analyses when solving the everification problem. We did this for as cuts and as branching heuristic in branch-and-bound procedure. rval propagation in which the choice of the ReLU relaxation at each node is determined via optimisation

We have released the state-of-the-art, open-source toolkit Venus implementing these methods. Venus has been used to analyse Neural Network-Based Systems in the Aircraft Domain developed by Boeing. an object detection system trained for open category detection and (ii.) a neural controller trained to assist landing in non-towered airports. Venus is used to identify the conditions under which these systems satisfy local robustness or generate counterexamples that comprehensively show the circumstances under which safety cannot be guaranteed

NMAS verification. Scalability - more evident when full systems are considered. The overall verification problem against CTL properties is undecidable [Akintunde *et al.*, 2022]. Decidable fragments can be obtained by the consideration of bounded properties, i.e.,

We develop and solve the resulting verification problem via a mixed-integer linear pro- gramming (MILP) formulation, present a tool developed for this task and evaluate the approach on an avionics advisory system for collision avoidance.

formulae of bounded CTL build upon temporal modalities indexed with natural numbers denoting the temporal depth up to which the formula is evaluated

MILP translations. Parallel executions

o further alleviate the difficulty of the verification problem, we also introduce a novel algorithm that checks for the occurrence of bugs in parallel over the execution paths. As we show, in the case of bounded safety specifications, this enables us to return a bug to the user as soon as a violation is identified on any of the branching paths that are explored in parallel.

monolithic and compositional verification algorithms

to coNE xp T imE upper bound for bounded CTL.

As neural networks are functions over the reals, the states of the agents have infinite domains, as opposed to finite ones in standard MAS. Starting from an initial state the sequence of joint actions of the agents in a NIS induces a computation tree where the evaluation of what holds true at each node of the tree, is not a simple check in constant time set inclusion, but it involves the non-trivial check of whether the output of neural networks satisfies some linear constraints for an infinite set of inputs. — Accounting for real-valued inputs. This is an NP-complete problem.

4 Conclusions

Acknowledgments

VAS group.

References

[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.

[Alur *et al.*, 1998] R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the International Symposium Compositionality: The Significant*

Difference (COMPOS97), volume 1536 of *Lecture Notes in Computer Science*, pages 23–60. Springer, 1998.

[Aminof *et al.*, 2014] B. Aminof, T. Kotek, S. Rubin, F. Spegni, and H. Veith. Parameterized model checking of rendezvous systems. In *CONCUR 2014–Concurrency Theory*, pages 109–124. Springer, 2014.

[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.

[Bloem *et al.*, 2015] R. Bloem, S. Jacobs, A. Khalimov, I. Konnov, S. Rubin, H. Veith, and J. Widder. *Decidability of Parameterized Verification*. Morgan and Claypool Publishers, 2015.

[Boureanu *et al.*, 2016] I. Boureanu, P. Kouvaros, and A. Lomuscio. Verifying security properties in unbounded multiagent systems. In *Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS16)*, pages 1209–1218. IFAAMAS Press, 2016.

[Clarke *et al.*, 1999] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, 1999.

[Emerson and Kahlon, 2000] E. Emerson and V. Kahlon. Reducing model checking of the many to the few. In *Proceedings of the 17th International Conference on Automated Deduction (CADE00)*, volume 1831 of *Lecture Notes in Computer Science*, pages 236–254. Springer, 2000.

[Emerson and Namjoshi, 1995] E. Emerson and K. Namjoshi. Reasoning about rings. In *Proceedings of the 22nd Annual Sigact-Aigplan on Principles of Programming Languages (POPL95)*, pages 85–94. Pearson Education, 1995.

[Fagin *et al.*, 1995] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, Cambridge, 1995.

[Katz *et al.*, 2017] G. Katz, C. W. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *CAV17*, pages 97–117, 2017.

[Kouvaros and Lomuscio, 2013a] P. Kouvaros and A. Lomuscio. Automatic verification of parametrised interleaved multi-agent systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent systems (AAMAS13)*, pages 861–868. IFAAMAS Press, 2013.

[Kouvaros and Lomuscio, 2013b] P. Kouvaros and A. Lomuscio. A cutoff technique for the verification of parameterised interpreted systems with parameterised environments. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI13)*, pages 2013–2019. AAAI Press, 2013.

[Kouvaros and Lomuscio, 2015a] P. Kouvaros and A. Lomuscio. A counter abstraction technique for the verification of robot swarms. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI15)*, pages 2081–2088. AAAI Press, 2015.

[Kouvaros and Lomuscio, 2015b] 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 and Lomuscio, 2017a] P. Kouvaros and A. Lomuscio. Parameterised verification of infinite state multiagent systems via predicate abstraction. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI17)*, pages 3013–3020. AAAI Press, 2017.

[Kouvaros and Lomuscio, 2017b] P. Kouvaros and A. Lomuscio. Verifying fault-tolerance in parameterised multiagent systems. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI17)*, pages 288–294. AAAI Press, 2017.

[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 and 23rd European Conference on Artificial Intelligence (IJCAI-ECAI18)*, pages 324–330. AAAI Press, 2018.

[Kouvaros *et al.*, 2019] P. Kouvaros, A. Lomuscio, E. Pirovano, and H. Punchihewa. Formal verification of open multi-agent systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS19)*, pages 179–187. IFAAMAS Press, 2019.

[Lomuscio and Michaliszyn, 2015] A. Lomuscio and J. Michaliszyn. Verifying multi-agent systems by model checking three-valued abstractions. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS15)*, pages 189–198, 2015.

[Lomuscio and Raimondi, 2006] A. Lomuscio and F. Raimondi. Model checking knowledge, strategies, and games in multi-agent systems. In *Proceedings of the 5th International Joint Conference on Autonomous agents and Multi-Agent Systems (AAMAS06)*, pages 161–168. ACM Press, 2006.

[Montali *et al.*, 2014] M. Montali, D. Calvanese, and G. De Giacomo. Verification of data-aware commitment-based multiagent system. In *Proceedings of AAMAS14*, pages 157–164. IFAAMAS, 2014.