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

Recent 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 and opacity hinders its adoption in safety-critical applications. The associated risks are compounded in an increasingly interconnected, socio-techno-economic world, where systems of multiple interacting intelligent agents, or multi-agent systems (MAS), constitute a paradigm shift from object-oriented to interaction-oriented design standards.

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.

One of the key techniques that has emerged in the area is that of *model checking* [Clarke *et al.*, 1999]. Model checking provides automated solutions to the problem of estabilishing whether a model M_S representing a system S satisfies a logical formula ϕ_P encoding a specification P. In the case of MAS, the formula φ does not simply express temporal properties of systems, as in reactive systems, but it may also be accounting for high-level attitudes of agency, such as knowledge and strategies, which can be described in temporal-epistemic logic [Fagin *et al.*, 1995a] and alternating-time logic [Alur *et al.*, 1998].

Whilst a number of methods, such as binary binary decision diagrams [Gammie and van der Meyden, 2004] and bounded model checking [Penczek and Lomuscio, 2003], enabled the model checking of complex systems of large state spaces, a main drawback of the approach remains the *state-space explosion problem*. Succinctly put, whereas model checking whether M_S satisfies ϕ can be solved in polynomial time [Clarke *et al.*, 1999], the size of the model M_S to analyse is exponential in the number of variables encoding the agents in the system S.

Notwithstanding that in practice this limits model checking to the verification of systems with only few constituents, the analysis of systems with arbitrarily many participants, such as robot swarms and applications in the Internet of Things, raises a principal barrier to the application of model checking. Indeed, verifying systems of this kind, henceforth *unbounded multi-agent systems* (UMAS), requires checking whether any system for any number of agents satisfies the specification in question. This renders model checking intractable when enumerating and analysing all individual systems.

Another key limitation of model checking is the requirement that the systems are given in traditional and agent-based programming languages, thereby not accounting for the recent shift to synthesise parts of the agents from data and implement via neural networks. Systems of this kind, henceforth *neuro-symbolic multi-agent systems* (NMAS), constitute important forthcoming applications, such as autonomous vehicles and personal negotiation assistants, where the neural components are employed to perform complex tasks, such as computer vision and natural language processing.

While the efficacy of neural networks with respect to these tasks has significantly improved over the past few years, their fragility to adversarial attacks [Szegedy *et al.*, 2014] and lack of interpretability [Doshi-Velez and Kim, 2017] raise additional concerns regarding the overall system safety, thereby strengthening the need for their principled analysis before deployment.

This paper gives an overview of the methods that we developed within the Verification of Automous Systems research group [1] towards the formal verification of UMAS and NMAS. Our pioneering work in the verification of UMAS, discussed in Section 2, overcomes the model checking barrier with the

[1] https://vas.doc.ic.ac.uk

development of methods that enable the derivation of the number of agents that is sufficient to consider when evaluating a specification. Our studies in the analysis of NMAS, outlined in Section 3, include efficient methods for the verification of neural networks and mixed-integer linear programming (MILP) formulations for checking system-level specifications. The paper concludes in Section 4 with directions for future work.

2 Unbounded Multi-agent 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 temporal logic [Fagin *et al.*, 1995a; Lomuscio and Raimondi, 2006]. Parameterised Interpreted Systems (PIS) is a parametric 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, 2016c] settings. The parameter in PIS denotes the number of agents in the system, each homogeneously constructed from an agent template.

The verification problem for PIS (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, 2016c]. Solutions to the problem can thus be given only in the form of incomplete techniques. Alternatively, decidable fragments can be curved by imposing restrictions on the systems and/or the specifications.

In either case, a key concept that enables the verification of UMAS is that of a *cutoff*. A cutoff is a natural number that expresses the number of agents that is sufficient to consider 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.

Whilst we've shown that 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]. Moreover, for the cases where they do not exist, theoretical analyses that we conducted show that these often concern systems that can exhibit *impractical* cyclic behaviours whose number of repetitions depends on the exact number of agents in the system [Kouvaros and Lomuscio, 2013b].

Following these observations we have analysed various sufficient conditions for the identification of cutoffs. The conditions were drawn 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 derivation [Kouvaros and Lomuscio, 2013a]. When 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 these 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, 2016c]. The heterogeneous semantics that we introduced allow for broadcast actions that may either concern all agents of all agent templates or all agents following a certain template. They additionally enable pairwise interactions between agents of different roles, thereby surpassing the expressive power of the homogeneous model.

Further gains in the expressivity of the protocols that can be verified have been obtained from our studies on UMAS programmed using variables with infinite domains [Kouvaros and Lomuscio, 2017a]. The resulting verification 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.

We have released the open-source parameterised verification toolkit MCMAS-P implementing the cutoff procedures for UMAS. 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, 2016c].

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 comprising *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]. Having used simulation-based abstractions to deal with the infinity of the agents, we have then presented a translation to PIS to solve their verification problem [Belardinelli *et al.*, 2017]. Analogous translations to PIS were given for *open MAS*, where countably many agents can join and leave the system at runtime [Kouvaros *et al.*, 2019].

Finally, adaptations of the verification methods for PIS enabled us to derive techniques for the verification of opinion formation protocols in swarms, which we used to give formal guarantees on the outcome of consensus protocols [Kouvaros and Lomuscio, 2016a]. Still others facilitated the verification of strategic properties of UMAS expressed in a parameterised variant of alternating-time temporal logic that we introduced [Kouvaros and Lomuscio, 2016b].

I conclude this section by noting that complementary to protocol correctness, which the aforementioned 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 a UMAS can tolerate [Kouvaros *et al.*, 2018].

3 Neuro-symbolic Multi-agent Systems

To reason about the properties of NMAS, we have introduced *neural interpreted systems* (NIS), a novel formalisation of MAS based on interpreted systems. In a nutshell, an agent in NIS comprises a perception mechanism implement via neural networks and coupled with a symbolic action mechanism.

The neural network components, which endow the agents with infinite domains, pose significant challenges to the verification problem. In particular, differently from traditional verification for symbolic systems, where atomic formulae are evaluated in constant time at symbolic states of the system, the evaluation of atomic formulae in NIS includes the computation of the output regions of the neural networks for a (potentially infinite) set of inputs, which is an NP-complete problem [Katz *et al.*, 2017]. Increasingly sophisticated solutions to the problem have been put forward in the past few years offering a principled way of analysing the robustness of neural networks to adversarial attacks [Singh *et al.*, 2019; Kouvaros and Lomuscio, 2021; Wang *et al.*, 2021]

We first discuss our work on the verification of standalone neural networks which forms the backbone of our studies on the analysis of NMAS outlined later.

3.1 Neural network verification

The neural network verification problem is to determine whether the output of a given network is as expected for a potentially infinite set of inputs. One of its most common instantiations is the *local adversarial robustness problem* whereby the set of inputs denotes imperceptible perturbations to a given input and the set of outputs encodes classification equivalence for all perturbations.

While significant progress has been made in push-down engines to solve the problem [Brix *et al.*, 2023], scalability to industrial-size models found in complex tasks such as computer vision remains a key difficulty in the area.

Advances are driven by *complete* and *incomplete* methods. Complete methods can in principle return a definite answer as to the whether the verification problem is satisfied, whereas incomplete methods may be unable to decide one way or the other. Complete methods are based on exact MILP/SMT formulations [Bastani *et al.*, 2016; Katz *et al.*, 2017] and dedicated branch-and-bound procedures that tackle optimisation mappings of the problem [Bunel *et al.*, 2018]. Incomplete methods rely on linear/semi-definite relaxations of the ReLU non-linearities [Wong and Kolter, 2018; Raghunathan *et al.*, 2018] thereby displaying better scalability over complete techniques. In the cases where the induced over-approximations do not allow for solutions to be given, network properties such as operational bounds computed by the methods can be used to strengthen the formulations used in complete verification [Botoeva *et al.*, 2020].

Our efforts in the area included methods towards improving scalability in complete verification and techniques in the direction of strengthening precision in incomplete verification. Concerning complete verification, we have introduced the novel concept of *ReLU dependency* whereby ReLU nodes are in a dependency relation if their operational states for a set of inputs is connected by logical implication. We have devised methods for the computation of these dependencies, which we have translated into MILP cuts, thereby improving the efficacy of MILP formulations [Botoeva *et al.*, 2020]. Further improvements have been possible via the derivation of dependency-based branching heuristics in branch-and-bound procedures [Kouvaros and Lomuscio, 2021].

Regarding incomplete verification, we have devised an abstraction method that strengthens the linear relaxations of the ReLU non-linearities by additionally accounting for intra-layer dependencies, instead of simply relying on local over-approximation areas, when choosing a relaxation [Hashemi *et al.*, 2021]. This enabled proofs of adversarial robustness to be given to computer vision models whose verification problem could not be previously resolved. To a similar effect we have strengthen the semidefinite approximations of the ReLU non-linearities by considering layer-wise relaxations and incorporating linear cuts into their formulation [Batten *et al.*, 2021].

We have released the open-source neural network verification toolkit VENUS implementing the aforementioned methods. In the span of four years, VENUS has progressed from analysing networks of few thousands of nodes to examining networks of millions of nodes. Among the latter are neural network-based systems in the aircraft domain developed by Boeing, including object detection and landing assistance systems [Kouvaros *et al.*, 2021], and semantic segmentation models for pose estimation [Kouvaros *et al.*, 2023].

3.2 NMAS verification

The scalability hurdle previously discussed is even more prevalent when analysisng closed-loop systems with neural components such as NMAS. In particular, we've shown their verification problem to be undecidable for plain reachability properties [Akintunde *et al.*, 2022]. In the light of this we've isolated bounded fragments of computation tree logic and alternating-time logic, where formulae can be evaluated in a bounded number of execution steps. This enabled us to analyse properties concerning, for instance, whether the agents can bring about a state of affairs or reach a safe configuration within a bounded number of steps. We have shown that verification with respect to these fragments is in coNEXPTIME and solved the resulting verification problems via MILP formulations [Akintunde *et al.*, 2020a; Akintunde *et al.*, 2020b]. At the heart of these formulations lie the the neural network verification procedures described earlier which are used for tightening the encodings of the networks. Inspired by bounded model checking [Clarke *et al.*, 2001], we have also developed compositional encodings, which can be used to check for the occurrence of bugs in parallel over the execution paths. As we have experimentally shown, the encodings often help to alleviate the difficulty of verification by enabling the identification of property violations in shallow depths of the corresponding branching paths [Akintunde *et al.*, 2020a].

4 Conclusions and Future Work

As argued in the Introduction, with the development of autonomous agents and multi-agent systems in diverse applications, there is an urgent need to study principled methods for their verification before deployment. This paper gave an overview of some of the key methods that we put forward towards the verification of key classes of systems, including standalone neural networks and unbounded and neuro-symbolic multi-agent systems.

While significant progress has been made in the formal analysis of these systems, scalability remains the main challenge to overcome to address industrial-scale models embedded in forthcoming applications such as autonomous vehicles. Formal reasoning at this level needs also to account for richer specifications beyond the ones discussed in this paper, including robustness to semantic perturbations such as geometric transformations.

In future work we will focus on conquering the scalability of formal verification, extending the specification languages, and expanding the formal models to account for richer classes of systems, including unbounded systems of neuro-symbolic agents.

Acknowledgments

I am grateful to the Verification of Autonomous Systems research group, Imperial College London, for the mentoring and support in conducting the research presented in this paper. I acknowledge the financial support of the DARPA Assured Autonomy Programme (FA8750-18-C-0095) and the EPSRC Research Project Trusted Autonomous Systems (EP/I00520X/1). Finally, I thank the IJCAI 2023 program committee for the invitation to deliver a spotlight talk in the Early Career Spotlight Track.

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.

[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.

[Bastani *et al.*, 2016] O. Bastani, Y. Ioannou, L. Lampropoulos, D. Vytiniotis, A. V. Nori, and A. Criminisi. Measuring neural net robustness with constraints. In *NeurIPS16*, pages 2613–2621, 2016.

[Batten *et al.*, 2021] B. Batten, P. Kouvaros, A. Lomuscio, and Y. Zheng. Efficient neural network verification via layer-based semidefinite relaxations and linear cuts. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, pages 2184–2190. ijcai.org, 2021.

[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.

[Botoeva *et al.*, 2020] E. Botoeva, P. Kouvaros, J. Kronqvist, A. Lomuscio, and R. Misener. Efficient verification of neural networks via dependency analysis. In *AAAI20*, 2020.

[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.

[Brix *et al.*, 2023] Christopher Brix, Mark Niklas Müller, Stanley Bak, Taylor T Johnson, and Changliu Liu. First three years of the international verification of neural networks competition (vnn-comp). *arXiv preprint arXiv:2301.05815*, 2023.

[Bunel *et al.*, 2018] R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. Mudigonda. A unified view of piecewise linear neural network verification. In *Proceedings of the 31st An-*

nual Conference on Neural Information Processing Systems (NeurIPS18), pages 4790–4799, 2018.

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

[Clarke *et al.*, 2001] E. Clarke, A. Biere, R. Raimi, and Y. Zhu. Bounded model checking using satisfiability solving. *Formal Methods in System Design*, 19(1):7–34, 2001.

[Doshi-Velez and Kim, 2017] F. Doshi-Velez and B. Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv: 1702.08608*, 2017.

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

[Fagin *et al.*, 1995b] R. Fagin, J. Y. Halpern, and M. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79, 1995.

[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.

[Hashemi *et al.*, 2021] V. Hashemi, P. Kouvaros, and A. Lomuscio. Osip: Tightened bound propagation for the verification of relu neural networks. In *Proceedings of the 19th International Conference on Software Engineering and Formal Methods (SEFM21)*, pages 463–480. IEEE Computer Society, 2021.

[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. Formal verification of opinion formation in swarms. In *Proceedings of the 15th International Conference on Autonomous Agents and Multi-Agent systems (AAMAS16)*, pages 1200–1209. IFAAMAS Press, 2016.

[Kouvaros and Lomuscio, 2016b] 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, 2016c] 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 multi-agent 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 multi-agent systems. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI17)*, pages 288–294. AAAI Press, 2017.

[Kouvaros and Lomuscio, 2021] P. Kouvaros and A. Lomuscio. Towards scalable complete verification of relu neural networks via dependency-based branching. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, pages 2643–2650. ijcai.org, 2021.

[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.

[Kouvaros *et al.*, 2021] P. Kouvaros, T. Kyono, F. Leofante, A. Lomuscio, D. Margineantu, D. Osipychev, and

Y. Zheng. Formal analysis of neural network-based systems in the aircraft domain. In *Proceedings of the 24th International Symposium on Formal Methods (FM21)*, volume 13047 of *Lecture Notes in Computer Science*, pages 730–740. Springer, 2021.

[Kouvaros *et al.*, 2023] P. Kouvaros, F. Leofante, C. Chung, B. Edwards, D. Margineantu, and A. Lomuscio. Verification of semantic key point detection for aircraft pose estimation. In *To Appear.*, 2023.

[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.

[Penczek and Lomuscio, 2003] W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.

[Raghunathan *et al.*, 2018] A. Raghunathan, J. Steinhardt, and P. S. Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS18*, pages 10900–10910, 2018.

[Singh *et al.*, 2019] G. Singh, T. Gehr, M. Püschel, and P. Vechev. An abstract domain for certifying neural networks. *POPL19*, 3(POPL):1–30, 2019.

[Szegedy *et al.*, 2014] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR14*, 2014.

[Wang *et al.*, 2021] S. Wang, H. Zhang, K. Xu, X. Lin, S. Jana, C. Hsieh, and Z.Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. In *Advances in Neural Information Processing Systems*, pages 29909–29921, 2021.

[Wong and Kolter, 2018] E. Wong and J. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML18*, pages 5286–5295, 2018.