Towards a Coalition Refinement Approach in the Strategic Verification of Multi-Agent Systems

Angelo Ferrando

Department of Informatics, Bioengineering, Robotics and Systems Engineering, University of Genoa, Italy

Abstract

In the context of formal verification of Multi-Agent Systems, it is common to check whether a subset of agents (also called a coalition) can achieve specific goals of interest, usually expressed as temporal properties. However, each coalition is hand-picked, and there is no guarantee that the agents within it will actually cooperate during system execution. This creates a gap between what the agents are assumed to achieve statically and what they can achieve in practice dynamically. In this paper, we explore and lay the foundation for an engineering approach to guide a coalition refinement technique. Multi-Agent Systems are first statically verified with respect to certain coalitions and then revised based on the actual dynamic behaviour of the agents during runtime.

Keywords

Multi-Agent Systems, Model Checking, Runtime Monitoring, Strategic Reasoning

1. Introduction

Multi-Agent Systems (MAS) are distributed systems composed of intelligent components, defined as agents. Nowadays, software systems are at the centre of our lives and are becoming more ubiquitous, decentralised, and complex. MAS are a good abstraction and engineering methodology to both tackle the theory and practice of nowadays software systems [2]. Nonetheless, as it is hard for monolithic software systems, it is even harder for distributed ones to guarantee correctness. The process of testing [3], debugging [4], and verifying [5] such systems can be quite complex.

In the research area of MAS, formal verification is especially used to check whether a subset of agents (also called coalition) is capable of achieving a set of goals (usually specified through temporal logics, like Alternating-Time Temporal Logic (ATL) [6]). In such scenarios, we talk about formal verification of strategic properties; since we are interested (in general) in the existence of strategies[1] for the agents to achieve their own goals. This is usually considered for a specific set of agents, that are assumed to cooperate. Nonetheless, it is not always possible to predict which agents will cooperate in reality. Because of that, even though we can verify that

IPS-RCRA-SPIRIT 2023: Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy. November 7-9th, 2023, Rome, Italy [1]

angelo.ferrando@unige.it (A. Ferrando)
https://angeloferrando.github.io (A. Ferrando)
0000-0002-8711-4670 (A. Ferrando)

[1]We formally present the notion of strategies later on.

by collaborating in a coalition, agents can indeed achieve their own goals; it is not guaranteed whether they will decide to do so at execution time (when the system will be effectively executed). This choice affects the knowledge we have of the system (as well as the agents in it) and can guide a refinement of which (if any) coalition of agents can be used.

In this paper, we present an approach to guide the refinement of coalition of agents for the strategic verification of MAS. We outline the foundational steps required and focus on the monitoring process used to detect when agents stop cooperating at runtime. By doing so, we put the basis for further works on the combination of runtime monitoring and strategic verification of MAS.

2. Related Work

Among the logics for strategic reasoning, we may find Strategy Logic (SL). SL is a powerful formalism for strategic reasoning, extensively covered in the work of Mogavero *et al.* [7]. SL treats strategies as first-order objects, employing existential ($\exists x$) and universal ($\forall x$) quantifiers to denote the existence of a strategy (x) and the consideration of all strategies (x) in the reasoning process.

Strategic reasoning encompasses strategy classification into memoryless and memoryful categories, where memoryless strategies depend solely on the current game state, and memoryful strategies take into account the entire game history. To connect strategies with specific agents, SL employs an explicit binding operator (a, x).

Despite its expressiveness, SL's computational complexity presents challenges. It has been shown that the model-checking problem for SL becomes non-elementary complete [7], and its satisfiability becomes undecidable [8]. To mitigate this, researchers have explored various fragments of SL.

One such fragment is Strategy Logic with Simple-Goals (SL-SG) [9], where strategic operators, binding operators, and temporal operators are combined. Importantly, SL-SG is demonstrated to strictly subsume ATL and shares a P-Complete model checking problem with ATL [6].

Shifting focus to agents' information, we differentiate between perfect and imperfect information games [10]. In perfect information games, agents possess complete knowledge of the game. However, real-world scenarios often involve agents making decisions without access to all relevant information, akin to situations where some system variables are private [11, 12]. In game modeling, imperfect information is typically addressed by defining an indistinguishability relation over game states [11, 10, 13].

The presence of imperfect information significantly impacts model checking complexity. For instance, with imperfect information and memoryful strategies, ATL becomes undecidable [14]. To address these challenges, researchers have developed various approaches, including approximations to perfect information [15, 16, 17], notions of bounded memory [18, 19], and hybrid techniques [20, 21, 22].

To the best of our knowledge, the closest work to this paper is [23], where the authors discuss how to abstract the notion of coalitions from ATL specifications. In [23], the coalitions are not hard-coded by the user, but instead are automatically synthesised. This is related to the work presented herein because, in some sense, we can see [23] as a static coalition refinement method, while in this paper, we focus on a more dynamic refinement approach. Moreover, in [23], no consideration is done on the actual implementation of the MAS under analysis, while we consider its execution and guide the coalition refinement consequently.

3. Preliminaries

In this section we recall some preliminary notions. Given a set U, $\overline{U}$ denotes its complement. We denote the length of a tuple v as $|v|$, its j-th element as v_j, and its last element $v_{|v|}$ as $last(v)$. For $j \leq |v|$, let $v_{\geq j}$ be the suffix $v_j, \ldots, v_{|v|}$ of v starting from v_j and $v_{\leq j}$ the prefix $v_1, \ldots, v_j$ of v.

3.1. Model

We start by showing a formal model for Multi-Agent Systems via concurrent game structures with imperfect information [6, 24].

Definition 1. *A Concurrent Game Structure with imperfect information (iCGS) is a tuple* $M = \langle Ag, AP, S, s_I, \{Act_i\}_{i \in Ag}, \{\sim_i\}_{i \in Ag}, d, \delta, V \rangle$ *such that:*

- $Ag = \{1, \ldots, m\}$ *is a nonempty finite set of agents.*
- AP *is a nonempty finite set of atomic propositions (atoms).*
- $S \neq \emptyset$ *is a finite set of* states, *with* initial state $s_I \in S$.
- *For every* $i \in Ag$, Act_i *is a nonempty finite set of* actions. *Let* $Act = \bigcup_{i \in Ag} Act_i$ *be the set of all actions, and* $ACT = \prod_{i \in Ag} Act_i$ *the set of all joint actions.*
- *For every* $i \in Ag$, $\sim_i$ *is a relation of* indistinguishability *between states. That is, given states* $s, s' \in S$, $s \sim_i s'$ *iff* s *and* s' *are indistinguishable for agent* i.
- *The* protocol function $d : Ag \times S \to (2^{Act} \setminus \{\emptyset\})$ *defines the availability of actions so that for every* $i \in Ag$, $s \in S$, *(i)* $d(i, s) \subseteq Act_i$ *and (ii)* $s \sim_i s'$ *implies* $d(i, s) = d(i, s')$.
- *The* transition function $\delta : S \times ACT \to S$ *assigns a successor state* $s' = \delta(s, \vec{a})$ *to each* $s \in S$, *for every joint action* $\vec{a} \in ACT$ *such that* $a_i \in d(i, s)$ *for every* $i \in Ag$.
- $V : S \to 2^{AP}$ *is the* labelling function.

According to Definition 1, a Concurrent Game Structure with imperfect information (iCGS) characterises how a collection of agents denoted as Ag interact. This interaction originates from an initial state $s_I \in S$ and follows the guidance of a transition function δ. The behaviour of this function is confined by the feasible actions available to agents, which are determined by the protocol function d. Additionally, we make the assumption that agents might possess incomplete information about the game. Consequently, in any given state s, agent i regards all states s' that are indistinguishable from s with respect to agent i, as being epistemically possible [25]. When each relation $\sim_i$ reduces to the identity, meaning that $s \sim_i s'$ only when $s = s'$, the outcome is a conventional Concurrent Game System (CGS) exhibiting perfect information [6].

A history $h \in S^+$ is a finite (non-empty) sequence of states. The indistinguishability relations are extended to histories in a synchronous, point-wise way, *i.e.*, histories $h, h' \in S^+$ are *indistinguishable* for agent $i \in Ag$, or $h \sim_i h'$, iff (i) $|h| = |h'|$ and (ii) for all $j \leq |h|$, $h_j \sim_i h'_j$.

3.2. Syntax

We use ATL* [6] to reason about the strategic abilities of agents.

Definition 2. *State (φ) and path (ψ) formulas in ATL* are defined as follows:*

$$\begin{array}{rcl} \varphi & ::= & q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\!\langle \Gamma \rangle\!\rangle \psi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid (\psi U \psi) \end{array}$$

where $q \in AP$ and $\Gamma \subseteq Ag$.

Formulas in ATL are all and only the state formulas.*

As usual, a formula $\langle\!\langle \Gamma \rangle\!\rangle \Phi$ is read as "the agents in coalition Γ have a strategy to achieve Φ". The meaning of temporal operators *next* X and *until* U is standard [26]. Operators $[\![\Gamma]\!]$, *release* R, *eventually* F, and *globally* G can be introduced as usual.

3.3. Semantics

We assume that agents employ *uniform strategies* [24], *i.e.*, they perform the same action whenever they have the same information.

Definition 3. *A* uniform perfect recall strategy *for agent $i \in Ag$ is a function $\sigma_i : S^+ \to Act_i$ such that for all histories $h, h' \in S^+$, (i) $\sigma_i(h) \in d(i, last(h))$ and (ii) $h \sim_i h'$ implies $\sigma_i(h) = \sigma_i(h')$.*

As per Definition 3, any strategy adopted by agent i necessitates the selection of actions that are valid for that specific agent. Additionally, whenever two histories appear indistinguishable to agent i, the same action is expected to be chosen. It is worth noting that in cases involving perfect information, condition (ii) is met by any strategy σ. Moreover, memoryless (or imperfect recall) strategies can be achieved by considering the domain of σ_i within S; in other words, $\sigma_i : S \to Act_i$. In the context of an iCGS M, a "path" π signifies an unending sequence of states. The collection of such paths over S is denoted as S^ω. Given a collective strategy Σ_Γ, which includes an individual strategy for each agent within the coalition Γ, a path π is considered Σ_Γ-*compatible* if, for each $j \geq 1$, $\pi_{j+1} = \delta(\pi_j, \vec{a})$ for some joint action $\vec{a}$, where for every $i \in \Gamma$, $a_i = \sigma_i(\pi_{\leq j})$ and for each $i \in \overline{\Gamma}$, $a_i \in d(i, \pi_j)$. The set of all Σ_Γ-*compatible* paths starting from state s is denoted as $out(s, \Sigma_\Gamma)$.

Now, we have all the ingredients to give the semantics of ATL*.

Definition 4. *The satisfaction relation $\models$ for an iCGS M, state $s \in S$, path $\pi \in S^\omega$, atom $q \in AP$, and ATL* formula ϕ is defined as (clauses for Boolean connectives are immediate and thus omitted):*

$$\begin{array}{lll} (M, s) \models q & \textit{iff} & q \in V(s) \\ (M, s) \models \langle\!\langle \Gamma \rangle\!\rangle \psi & \textit{iff} & \textit{for some joint strategy } \Sigma_\Gamma, \\ & & \textit{for all } \pi \in out(s, \Sigma_\Gamma),\ (M, \pi) \models \psi \\ (M, \pi) \models \varphi & \textit{iff} & (M, \pi_1) \models \varphi \\ (M, \pi) \models X\psi & \textit{iff} & (M, \pi_{\geq 2}) \models \psi \\ (M, \pi) \models \psi U \psi' & \textit{iff} & \textit{for some } k \geq 1,\ (M, \pi_{\geq k}) \models \psi', \textit{ and} \\ & & \textit{for all } 1 \leq j < k,\ (M, \pi_{\geq j}) \models \psi \end{array}$$

We say that formula φ is *true* in an iCGS M, or $M \models \varphi$, iff $(M, s_I) \models \varphi$. Now, we state the model checking problem.

Definition 5. *Given an iCGS M and a formula φ, the model checking problem concerns determining whether $M \models \varphi$.*

Given that the interpretation presented in Definition 4 corresponds to the conventional understanding of ATL* [6], it is a recognised fact that verifying ATL* through model checking against iCGS characterised by imperfect information and perfect recall is an undecidable problem [27]. Since in this paper we are interested in proposing an engineering approach to revise the coalitions used in the verification of ATL* formulas, we assume to be always in the decidable fragments. That is, CGSs with perfect recall strategies, or, iCGSs with imperfect recall strategies.

4. Towards a coalition refinement approach

In this section, we overview our envisaged engineering methodology. It consists of the following steps (as depicted in Figure 1). First, an iCGS (representing a MAS) is verified against one (or multiple) ATL* properties. Naturally, this requires choosing the coalition we assume the agents will be in. Afterwards, by verifying these properties, we extract the strategies employed by the agents in the coalitions (*i.e.*, the strategies that, if properly enacted, enable the agents to achieve their temporal goals). Once these strategies are extracted, we can synthesise the corresponding monitors to check at execution time whether the agents adhere to the winning strategies or not. If that is the case, then the approach concludes. However, if at least one agent is not following any of the winning strategies as intended (*i.e.*, such an agent is not collaborating with the agents in its coalition), then two outcomes need to be reported. Firstly, the temporal objective related to the compromised coalition is no longer guaranteed to be achieved (this information is available at runtime while the system is still operational and can be used to trigger fail-safe behaviours). Secondly, the coalition that has been compromised at runtime is consistently updated for further verification rounds.

Remark 1. *Note that, when solving the model checking problem, we only receive a Boolean result, indicating whether the formal specification of interest is satisfied in the model under analysis or not. However, in our proposed approach, we envision utilising model checking to extract strategies, similar to what can be accomplished with tools like the MCMAS model checker. This approach extends beyond conventional formal verification and moves towards formal synthesis. Consequently, if formal verification is employed, an additional engineering step becomes necessary to reconstruct the actual joint winning strategy, rather than solely relying on the Boolean verification outcome.*

Please note that in Figure 1 and in the rest of the section, we consider only strategic properties with a single strategic operator. This choice is made to enhance readability and serves as a foundation for a more comprehensive approach (which will require further study, as we will discuss in Section 5).

We outline the steps envisioned for our approach, deferring detailed analysis for future exploration and research.

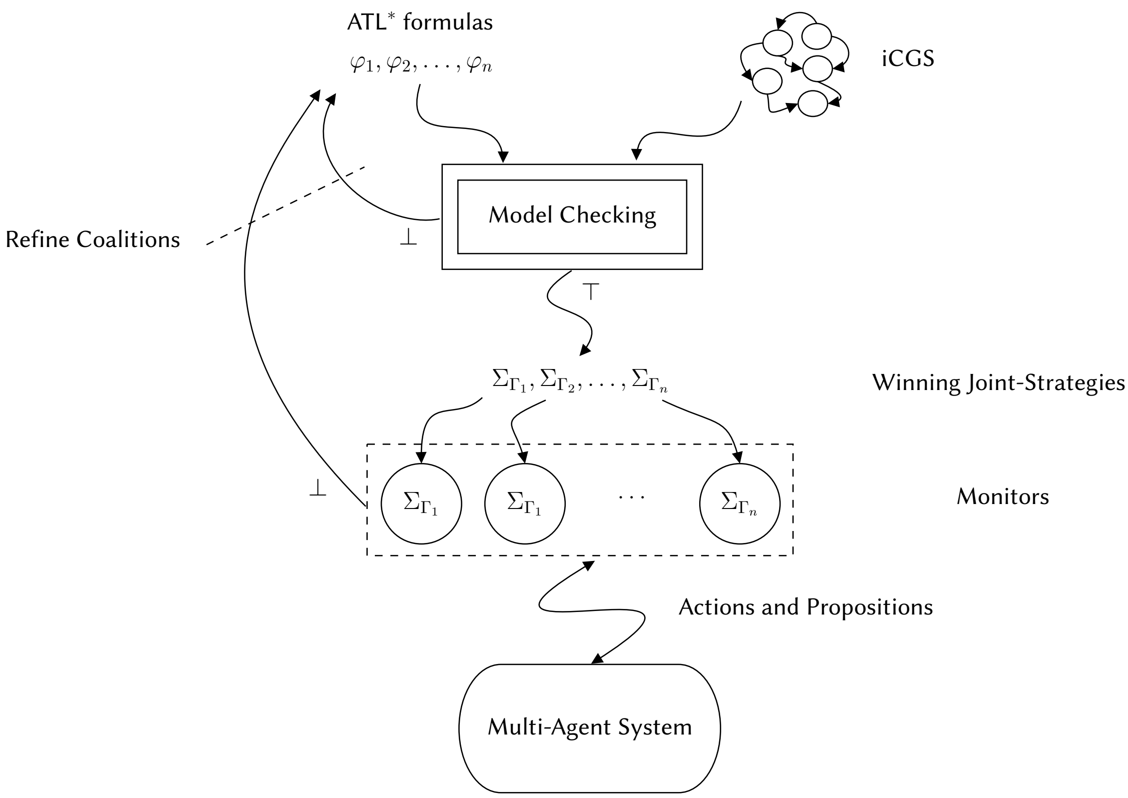

Figure 1: Overview of the approach, where each ATL formula is of the form $\varphi_i = \langle\langle\Gamma_i\rangle\rangle\psi_i$ (with ψ_i comprising only temporal operators).

1. *Step i: Formal Verification.* The initial stage of our methodology involves formal verification. Specifically, we verify one or multiple ATL* properties against an iCGS, representing the model of the Multi-Agent System under analysis. This verification is accomplished by solving the corresponding model checking problem, as defined in Definition 5.
2. *Step ii: Formal Synthesis.* Once the verification step is completed, the joint winning strategies Σ_Γ^{win} can be extracted and analysed, a task that can be performed using the MCMAS model checker.
3. *Step iii: Strategy Violation Detection.* With the set Σ_Γ^{win} of winning strategies in hand, we synthesise corresponding monitors to assess the agents' runtime conformance. These monitors check whether the agents adhere to their winning strategies, as extracted in the previous step.
4. *Step iv: Coalition Revision.* In case a violation is detected, this information can be leveraged to adjust the agent coalitions employed during the static verification phase. This adjustment is based on the observation that certain agents may not have adhered to their winning strategies, potentially jeopardising the attainment of the temporal goals.

Up to this point, our primary focus has been on verifying the MAS under analysis and assessing whether the agents (integral to the verification) adhere to expected behaviour (*i.e.*, whether they enact winning strategies or not). However, there are two crucial aspects that require consideration. Firstly, we need to address how the monitors will gather information about the MAS. Secondly, once a monitor reports a violation, we must determine the appropriate course of action.

Firstly, the first aspect is quite pragmatic, as it pertains to the practical verification of the MAS runtime execution. To accomplish this, we require a method to map the state of the MAS to the iCGS state, as well as a means to track the actions executed by the agents within the MAS. The latter is a straightforward process and can be achieved by logging every time an agent performs an action during runtime. This logging can be implemented by instrumenting the software system, much like in runtime verification practices [28], where additional instructions (typically for logging) are added to the source code. Consequently, when the instrumented MAS is executed, it produces additional information that the monitors utilise to assess the agents' adherence to any winning strategies. Instrumentation can also be employed to gather information about the agents' states, such as their beliefs. This information can be mapped to the corresponding atomic propositions represented in the iCGS at the verification stage. This alignment allows the runtime execution of the MAS to align with its abstract representation (the model). It is worth noting that this mapping step may depend on domain-specific knowledge and may necessitate at least partial hard-coding.

Secondly, the second aspect we need to address is the coalition revision process. This is, in our opinion, the most significant and challenging aspect of the approach. Indeed, the revision of coalitions can impact both the verification of the MAS and the agents' behaviour during runtime. In this paper, we have laid the foundation and outlined a potential road-map for this approach, with much more to be developed. However, we firmly believe that by combining runtime monitoring and formal verification for strategic reasoning, we can achieve highly flexible and reliable MAS. Runtime monitoring can detect violations of (winning) strategies and, consequently, of agent coalitions. This information can be used to revise the coalitions employed during static verification of strategic properties in the MAS. Furthermore, the presence of monitors that assess agent behaviour during runtime enhances the system's reliability [29, 30]. Indeed, when strategy violations are detected, we can trigger fail-safe behaviours to assist the agents in still achieving their respective temporal goals.

5. Conclusions and Future Work

In this paper, we have highlighted the steps of a potential approach to utilise runtime monitoring to guide coalition revision in the formal verification of strategic properties in MAS. Our primary focus has been on presenting the core idea and emphasising the significance of the resulting approach. We have introduced the high-level concept and outlined its steps.

Given that this paper serves as a foundational step toward the development of a coalition refinement methodology, we hope that the insights presented herein can provide a valuable starting point. Our envisioned future work involves the actual design, implementation and further investigation of the implications of coalition refinement on MAS verification. In this concise paper, we have only scratched the surface, but we firmly believe that deeper exploration will be beneficial for advancing the formal verification of strategic properties in MAS.

Acknowledgments

I would like to express my heartfelt gratitude to Vadim Malvone for his invaluable contributions to this paper. His insightful discussions and thoughtful comments have played a pivotal role in shaping the content and direction of this work. Vadim's expertise and dedication have been instrumental, and I am truly grateful for his ſupport throughout this reſearch endeavour. His input has enriched the quality of this paper, and I am honoured to acknowledge his significant role in its development.

References

[1] R. De Benedictis, M. Castiglioni, D. Ferraioli, V. Malvone, M. Maratea, E. Scala, L. Serafini, I. Serina, E. Tosello, A. Umbrico, M. Vallati, Preface to the italian workshop on planning and scheduling, rcra workshop on experimental evaluation of algorithms for solving problems with combinatorial explosion, and spirit workshop on strategies, prediction, interaction, and reasoning in italy (ips-rcra-spirit 2023), in: Proceedings of the Italian Workshop on Planning and Scheduling, RCRA Workshop on Experimental evaluation of algorithms for solving problems with combinatorial explosion, and SPIRIT Workshop on Strategies, Prediction, Interaction, and Reasoning in Italy (IPS-RCRA-SPIRIT 2023) co-located with 22th International Conference of the Italian Association for Artificial Intelligence (AI* IA 2023), 2023.

[2] F. Bergenti, M.-P. Gleizes, F. Zambonelli, Methodologies and software engineering for agent systems: the agent-oriented software engineering handbook, volume 11, Springer Science & Business Media, 2006.

[3] M. Winikoff, BDI agent testability revisited, Auton. Agents Multi Agent Syst. 31 (2017) 1094–1132. URL: https://doi.org/10.1007/s10458-016-9356-2. doi:10.1007/s10458-016-9356-2.

[4] M. Winikoff, Debugging agent programs with why?: Questions, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017, ACM, 2017, pp. 251–259. URL: http://dl.acm.org/citation.cfm?id=3091166.

[5] L. A. Dennis, M. Fisher, M. P. Webster, R. H. Bordini, Model checking agent programming languages, Autom. Softw. Eng. 19 (2012) 5–63. URL: https://doi.org/10.1007/s10515-011-0088-x. doi:10.1007/s10515-011-0088-x.

[6] R. Alur, T. A. Henzinger, O. Kupferman, Alternating-time temporal logic, J. ACM 49 (2002) 672–713. URL: https://doi.org/10.1145/585265.585270. doi:10.1145/585265.585270.

[7] F. Mogavero, A. Murano, G. Perelli, M. Vardi, Reasoning about strategies: On the model-checking problem, ACM Trans. Comp. Log. 15 (2014) 34:1–34:47. URL: http://doi.acm.org/10.1145/2631917. doi:10.1145/2631917.

[8] F. Mogavero, A. Murano, G. Perelli, M. Y. Vardi, Reasoning about strategies: on the satisfiability problem, Log. Methods Comput. Sci. 13 (2017). URL: https://doi.org/10.23638/LMCS-13(1:9)2017. doi:10.23638/LMCS-13(1:9)2017.

[9] F. Belardinelli, W. Jamroga, D. Kurpiewski, V. Malvone, A. Murano, Strategy logic with

simple goals: Tractable reasoning about strategies, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org, 2019, pp. 88–94. URL: https://doi.org/10.24963/ijcai.2019/13. doi:10.24963/ijcai.2019/13.

[10] J. H. Reif, The complexity of two-player games of incomplete information, J. Comput. Syst. Sci. 29 (1984) 274–301.

[11] O. Kupferman, M. Y. Vardi, Module checking revisited, in: CAV'97, Springer, 1997, pp. 36–47.

[12] R. Bloem, K. Chatterjee, S. Jacobs, R. Könighofer, Assume-guarantee synthesis for concurrent reactive programs with partial information, in: TACAS, 2015, pp. 517–532.

[13] A. Pnueli, R. Rosner, Distributed reactive systems are hard to synthesize, in: FOCS, 1990, pp. 746–757.

[14] C. Dima, F. Tiplea, Model-checking ATL under Imperfect Information and PerfectRecall Semantics is Undecidable., Technical Report, 2011.

[15] F. Belardinelli, A. Lomuscio, V. Malvone, An abstraction-based method for verifying strategic properties in multi-agent systems with imperfect information, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press, 2019, pp. 6030–6037. URL: https://doi.org/10.1609/aaai.v33i01.33016030. doi:10.1609/aaai.v33i01.33016030.

[16] F. Belardinelli, V. Malvone, A three-valued approach to strategic abilities under imperfect information, in: D. Calvanese, E. Erdem, M. Thielscher (Eds.), Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020, 2020, pp. 89–98. URL: https://doi.org/10.24963/kr.2020/10. doi:10.24963/kr.2020/10.

[17] F. Belardinelli, A. Ferrando, V. Malvone, An abstraction-refinement framework for verifying strategic properties in multi-agent systems with imperfect information, Artif. Intell. 316 (2023). URL: https://doi.org/10.1016/j.artint.2022.103847. doi:10.1016/j.artint.2022.103847.

[18] F. Belardinelli, A. Lomuscio, V. Malvone, Approximating perfect recall when model checking strategic abilities, in: M. Thielscher, F. Toni, F. Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018, AAAI Press, 2018, pp. 435–444. URL: https://aaai.org/ocs/index.php/KR/KR18/paper/view/18010.

[19] F. Belardinelli, A. Lomuscio, V. Malvone, E. Yu, Approximating perfect recall when model checking strategic abilities: Theory and applications, J. Artif. Intell. Res. 73 (2022) 897–932. URL: https://doi.org/10.1613/jair.1.12539. doi:10.1613/jair.1.12539.

[20] A. Ferrando, V. Malvone, Strategy RV: A tool to approximate ATL model checking under imperfect information and perfect recall, in: F. Dignum, A. Lomuscio, U. Endriss, A. Nowé (Eds.), AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021, ACM, 2021, pp. 1764–1766. URL: https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p1764.pdf. doi:10.5555/3463952.3464230.

[21] A. Ferrando, V. Malvone, Towards the combination of model checking and runtime verification on multi-agent systems, in: F. Dignum, P. Mathieu, J. M. Corchado, F. de la Prieta (Eds.), Advances in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection - 20th International Conference, PAAMS 2022, L'Aquila, Italy, July 13-15, 2022, Proceedings, volume 13616 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 140–152. URL: https://doi.org/10.1007/978-3-031-18192-4_12. doi:10.1007/978-3-031-18192-4_12.

[22] A. Ferrando, V. Malvone, Towards the verification of strategic properties in multi-agent systems with imperfect information, in: N. Agmon, B. An, A. Ricci, W. Yeoh (Eds.), Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023, ACM, 2023, pp. 793–801. URL: https://dl.acm.org/doi/10.5555/3545946.3598713. doi:10.5555/3545946.3598713.

[23] A. Ferrando, V. Malvone, How to find good coalitions to achieve strategic objectives, in: A. P. Rocha, L. Steels, H. J. van den Herik (Eds.), Proceedings of the 15th International Conference on Agents and Artificial Intelligence, ICAART 2023, Volume 1, Lisbon, Portugal, February 22-24, 2023, SCITEPRESS, 2023, pp. 105–113. URL: https://doi.org/10.5220/0011778700003393. doi:10.5220/0011778700003393.

[24] W. Jamroga, W. van der Hoek, Agents that know how to play, Fundam. Informaticae 63 (2004) 185–219. URL: http://content.iospress.com/articles/fundamenta-informaticae/fi63-2-3-05.

[25] R. Fagin, J. Y. Halpern, Y. Moses, M. Vardi, Reasoning about Knowledge., MIT, 1995.

[26] C. Baier, J.-P. Katoen, Principles of Model Checking (Representation and Mind Series), 2008.

[27] C. Dima, F. L. Tiplea, Model-checking ATL under imperfect information and perfect recall semantics is undecidable, CoRR abs/1102.4225 (2011). URL: http://arxiv.org/abs/1102.4225. arXiv:1102.4225.

[28] Y. Falcone, K. Havelund, G. Reger, A tutorial on runtime verification, in: M. Broy, D. A. Peled, G. Kalus (Eds.), Engineering Dependable Software Systems, volume 34 of *NATO Science for Peace and Security Series, D: Information and Communication Security*, IOS Press, 2013, pp. 141–175. URL: https://doi.org/10.3233/978-1-61499-207-3-141. doi:10.3233/978-1-61499-207-3-141.

[29] A. Ferrando, R. C. Cardoso, Safety shields, an automated failure handling mechanism for BDI agents, in: P. Faliszewski, V. Mascardi, C. Pelachaud, M. E. Taylor (Eds.), 21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022, International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS), 2022, pp. 1589–1591. URL: https://www.ifaamas.org/Proceedings/aamas2022/pdfs/p1589.pdf. doi:10.5555/3535850.3536044.

[30] D. C. Engelmann, A. Ferrando, A. R. Panisson, D. Ancona, R. H. Bordini, V. Mascardi, Rv4jaca - runtime verification for multi-agent systems, in: R. C. Cardoso, A. Ferrando, F. Papacchini, M. Askarpour, L. A. Dennis (Eds.), Proceedings of the Second Workshop on Agents and Robots for reliable Engineered Autonomy, AREA@IJCAI-ECAI 2022, Vienna, Austria, 24th July 2022, volume 362 of *EPTCS*, 2022, pp. 23–36. URL: https://doi.org/10.4204/EPTCS.362.5. doi:10.4204/EPTCS.362.5.