Towards Assume-Guarantee Verification of Strategic Ability

Extended Abstract

Łukasz Mikulski[1,2], Wojciech Jamroga[2,3], and Damian Kurpiewski[2,1]
[1] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University, Toruń, Poland
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
[3] Interdisciplinary Centre for Security, Reliability and Trust, SnT, University of Luxembourg, Luxembourg
lukasz.mikulski@mat.umk.pl,wojciech.jamroga@uni.lu,d.kurpiewski@ipipan.waw.pl

ABSTRACT

Formal verification of strategic abilities is a hard problem. We propose to use the methodology of assume-guarantee reasoning in order to facilitate model checking of alternating-time temporal logic with imperfect information and imperfect recall.

KEYWORDS

model checking, assume-guarantee reasoning, strategic ability

ACM Reference Format:
Łukasz Mikulski, Wojciech Jamroga, and Damian Kurpiewski. 2022. Towards Assume-Guarantee Verification of Strategic Ability: Extended Abstract. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022*, IFAAMAS, 3 pages.

1 INTRODUCTION

Alternating-time temporal logic **ATL*** [1, 2, 32] and *Strategy Logic* **SL** [10, 29] provide powerful tools to reason about strategic aspects of MAS. Specifications in agent logics can be used as input to algorithms and tools for *model checking* [3–5, 7–9, 11, 19, 21, 24, 26]. However, verification of strategic abilities suffers both from state-space and strategy-space explosion. Even for the more restricted logic **ATL**, model checking of its imperfect information variants ranges from $\Delta_2^{\mathbf{P}}$–complete to undecidable [6, 14, 16, 20, 32].

In this paper, we make the first step towards compositional model checking of strategic properties in asynchronous multi-agent systems with imperfect information and imperfect recall. To this end, we creatively expand the assume-guarantee framework of [27, 28]. Instead of searching through the states and strategies of the entire model, we "factorize" it and perform most of the search locally, using *assume-guarantee reasoning* [12, 31]. We illustrate the approach by means of a simple voting scenario. Finally, we evaluate the practical gains through verification experiments.

Related work. Compositional verification dates back to [18, 23, 30], and has been intensively studied for temporal specifications [12, 13, 15, 17, 25, 27, 28, 31]. Our approach is based on [27, 28], where assume-guarantee rules were defined for liveness properties of MAS. A related scheme for **ATL** with perfect information strategies and aspect-oriented programs was considered in [13].

2 MODELS OF CONCURRENT MAS

We use the MAS representations of [27, 28], which allow for asynchronous and synchronous composition of local transitions.

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online.

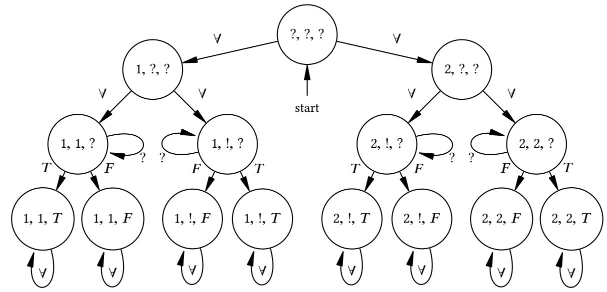

Figure 1: Module of a voter deciding between two candidates

Modules. Let D be a domain (for all variables used in the system). For any set of variables X, let D^X be a set of all valuation functions on X. We follow [28], and divide the variables in a module into *state variables* and *input variables*. An agent can read and modify only its state variables. The input variables are not a part of its state, but their values limit the set of executable transitions.

Definition 2.1 (Module [28]). A *module* is $M = (X, I, Q, T, \lambda, q_0)$, where: X, I are finite sets of state and input variables, respectively, with $X \cap I = \emptyset$; Q is a finite set of states; $\lambda : Q \to D^X$ labels each state with a valuation of the state variables; $q_0 \in Q$ is the initial state; $T \subseteq Q \times D^I \times Q$ is a transition relation. We require that if $(q, \alpha, q') \in T, q \neq q'$, then $(q, \alpha, q) \notin T$, and for each pair $(q, \alpha) \in Q \times D^I$ there exists $q' \in Q$ such that $(q, \alpha, q') \in T$.

Example 2.2. We use a voting scenario, inspired by [21, 22], consisting of modules $M^{(1)}, \ldots, M^{(n)}$ of voters and $M^{(c)}$ of a coercer.

Every voter has three local variables in $X^{(i)}$: $vote^{(i)}$: the vote being cast (?, 1, or 2); $reported^{(i)}$: the vote value presented to the coercer (?, 1, 2 or !), where ! means that the voter decided not to show her vote; $pstatus^{(i)}$: the punishment status (?, T or F). Moreover, $I^{(i)}$ consists of variable $pun^{(i)}$ controlled by the coercer. The voter first casts her vote, then decides whether to share its value with the coercer. Finally, she waits for the coercer's decision to punish her or to refrain from punishment. The module is shown in Figure 1.

The coercer has two available actions per voter: to punish the voter or to refrain from punishment. He can execute them in any order, but only after the respective voters decide to share or not.

Modules M, M' are *asynchronous* if $X \cap X' = \emptyset$. Note that all the modules presented in Example 2.2 are asynchronous.

Composition of Modules. The model of a MAS is given by the asynchronous composition $M = M^{(1)}| \ldots |M^{(n)}$ that combines modules $M^{(1)}, \ldots, M^{(n)}$ into a single module M [28]. The composition is standard; it only requires the compliance of the valuations.

Traces and Words. A trace of a module M is an infinite sequence of alternating states and transitions $\sigma = q_0\alpha_0q_1\alpha_1\ldots$, where q_0 is the initial state and $(q_i, \alpha_i, q_{i+1}) \in T$ for every $i \in \mathbb{N}$. An infinite word $w = v_0v_1,\ldots \in (D^X)^\omega$ is *derived* by module M with trace $\sigma = q_0\alpha_0q_1\alpha_1\ldots$ if $v_i = \lambda(q_i)$ for all $i \in \mathbb{N}$. An infinite word $u = \alpha_0\alpha_1,\ldots \in (D^I)^\omega$ is *admitted* by M with σ if $\sigma = q_0\alpha_0q_1\alpha_1\ldots$. Finally, w (resp. u) is derived (resp. admitted) by M if there exists a trace of M that derives (resp. admits) it.

3 WHAT AGENTS CAN ACHIEVE

Alternating-time temporal logic **ATL*** [1, 2, 32] introduces *strategic modalities* $\langle\langle C\rangle\rangle\gamma$, expressing that coalition C can enforce the temporal property γ. In this paper, we use the *imperfect information/imperfect recall* variant without next step operator X and nested strategic modalities, denoted **sATL*** ("simple **ATL***").

Syntax. Formally, the syntax of **sATL*** is defined by:

$$\phi ::= p(Y) \mid \neg\phi \mid \phi \wedge \phi \mid \langle\langle C\rangle\rangle\gamma; \quad \gamma ::= p(Y) \mid \neg\gamma \mid \gamma \wedge \gamma \mid \gamma\, \mathrm{U}\, \gamma.$$

where $p : Y \to D$ for some subset of domain variables $Y \subseteq X$. That is, each atomic statement refers to the valuation of a subset of variables used in the system. U is the "strong until" operator of **LTL**. The "sometime" and "always" operators F and G can be defined as usual by $\mathrm{F}\gamma \equiv \top\, \mathrm{U}\, \gamma$ and $\mathrm{G}\gamma \equiv \neg(\top\, \mathrm{U}\, \neg\gamma)$.

Semantics. A *memoryless imperfect information strategy* for agent i is a function $s_i : Q_i \to T_i$. We say that a trace σ (word derived with σ) *implements* a strategy s_i if for any j where $q_j^{(i)} \neq q_{j+1}^{(i)}$ we have $s_i(q_j^{(i)}) = (q_j^{(i)}, \alpha_j, q_{j+1}^{(i)})$, where $\alpha_j : I_i \to D$ and $\alpha_j(x) = \lambda(q_j)(x)$.

Let $C \subseteq \{1,\ldots,n\}$ be a set of agent indices. We define *joint strategies* for C as tuples of individual strategies, one per $i \in C$. The semantics of strategic operators is given by the following clause:

$M, q \models \langle\langle C\rangle\rangle\gamma$ if there exists a joint strategy s_C for C such that, for any word w that implements s_C, we have $M, w \models \gamma$.

4 ASSUMPTIONS AND GUARANTEES

We propose an assume-guarantee scheme, where one can reduce the complexity of model checking **sATL*** by verifying individual strategic abilities of single agents against overapproximating abstractions of its environment, i.e., the rest of the system. The general idea is that if an agent has a successful strategy in a more nondeterministic environment, then it can can use the same strategy to succeed in the original model. Moreover, it often suffices to prepare the abstraction based only of the modules that are connected with the agent by at most k synchronization steps.

Assumptions and Guarantees. The environmental abstractions are formalized by *assumptions* $A = (M_A, F)$, where M_A is a module and F is a set of accepting states that provide Büchi-style accepting rules for infinite traces derived by M. The assumption should be constructed so that it *guarantees* that the set of computations accepted by A covers the sequences of changes in the input variables I_M of module M. We capture those changes by the notion of *curtailment*. Formally, a sequence $v = v_1v_2\ldots$ over D^Y is a curtailment of sequence $u = u_1u_2\ldots$ over D^X (where $Y \subseteq X$) if there exists an infinite sequence of indices $j_1 < j_2 < \ldots$ with $j_1 = 1$ such that $\forall_i \forall_{j_i \le k < j_{i+1}} v_i = u_k|_Y$.

V	Monolithic model checking				Assume-guarantee verification			
	#st	**#tr**	**DFS**	**Apprx**	**#st**	**#tr**	**DFS**	**Apprx**
2	529	2216	<0.1	<0.1/Yes	161	528	<0.1	<0.1/Yes
3	1.22e4	1.28e5	<0.1	0.8/Yes	1127	7830	<0.1	<0.1/Yes
4	2.79e5	6.73e6	<0.1	30/Yes	7889	1.08e5	<0.1	0.5/Yes
5	6.43e6	3.42e8	timeout		5.52e4	1.45e6	<0.1	8/Yes
6	timeout				3.86e5	1.92e7	<0.1	135/Yes
7	timeout				timeout			

Table 1: Results of assume-guarantee verification for simple voting (times given in seconds; timeout=2h)

The Scheme. Let $M = M_1|M_2|\ldots|M_n$ be a system composed from modules $M_1, M_2, \ldots, M_n$, where $X_{M_i} \cap X_{M_j} = \emptyset$ for $i \neq j$. By $Comp_i^1$ we denote the composition of all modules directly related to M_i. Moreover, $Comp_i^k$ denotes the composition of the modules in $Comp_i^{k-1}$ and the modules directly related to them (except for M_i). Further, let $\psi_i, i \in C$ be path formulas of **sATL***, one for each agent in C. Simple assume-guarantee reasoning for strategic ability is provided by the following inference rule:

$$\mathbf{R_k} \quad \frac{\forall_{i\in C}\ M_i|A_i \models_{ir} \langle\langle i\rangle\rangle\psi_i \qquad \forall_{i\in C}\ Comp_i^k \models A_i}{M_1|\ldots|M_n \models_{ir} \langle\langle C\rangle\rangle \bigwedge_{i\in C}\psi_i}$$

5 EXPERIMENTS

Here, we present preliminary experimental results for the assume-guarantee rule proposed in Section 4, using the voting scenario of Example 2.2 as the benchmark. The assumptions are provided by a simplified module of the coercer, where he only waits for the value reported by $Voter_1$, no matter how he reacts to other voters' choices. The algorithms have been implemented in Python, and run on a server with 2.40 GHz Intel Xeon Platinum 8260 CPU, 991 GB RAM, and 64-bit Linux.

The verified formula was $\varphi \equiv \langle\langle Voter_1\rangle\rangle \mathrm{G}(\neg \mathrm{pstatus}_1 \vee \mathrm{voted}_1 = 1)$. The results are presented in Table 1. The first column describes the configuration of the benchmark, i.e., the number of the voters. Then, we report the performance of model checking algorithms that operate on the explicit model of the whole system vs. assume-guarantee verification. *DFS* is a straightforward implementation of depth-first strategy synthesis. *Apprx* refers to the method of fixpoint-approximation [21]; besides the time, we also report if the approximation was conclusive.

6 CONCLUSION

In this paper, we sketch how assume-guarantee reasoning can be extended for verification of strategic abilities. The main idea is to factorize coalitional abilities by the abilities of the coalition members, and to verify the individual abilities against Büchi-style abstractions of the agents' environment of action. Preliminary experimental evaluation has produced very promising results, showing noticeable improvement in the verification of large models consisting of asynchronous agents with independent goals.

ACKNOWLEDGMENTS

We acknowledge the support of the National Centre for Research and Development, Poland (NCBR), and the Luxembourg National Research Fund (FNR), under the PolLux/FNR-CORE project STV (POLLUX-VII/1/2019 – C18/IS/12685695/IS/STV/Ryan).

REFERENCES

[1] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. In *Proc. of FOCS'97*, pages 100–109. IEEE Comput. Soc. Press, 1997.

[2] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *J. of the ACM*, 49:672–713, 2002.

[3] R. Alur, T.A. Henzinger, F.Y.C. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. MOCHA: Modularity in model checking. In *Proc. of CAV'98*, volume 1427 of *LNCS*, pages 521–525. Springer, 1998.

[4] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of broadcasting multi-agent systems against an epistemic strategy logic. In *Proc. of IJCAI'17*, pages 91–97, 2017.

[5] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. Verification of multi-agent systems with imperfect information and public actions. In *Proc. of AAMAS'17*, pages 1268–1276, 2017.

[6] N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In *Specification and Verification of Multi-Agent Systems*, pages 125–159. Springer, 2010.

[7] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. Improving the model checking of strategies under partial observability and fairness constraints. In *Formal Methods and Software Engineering*, volume 8829 of *LNCS*, pages 27–42. Springer, 2014.

[8] P. Cermák, A. Lomuscio, F. Mogavero, and A. Murano. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *Proc. of CAV'14*, volume 8559 of *LNCS*, pages 525–532. Springer, 2014.

[9] P. Cermák, A. Lomuscio, and A. Murano. Verifying and synthesising multi-agent systems against one-goal strategy logic specifications. In *Proc. of AAAI'15*, pages 2038–2044, 2015.

[10] K. Chatterjee, T.A. Henzinger, and N. Piterman. Strategy Logic. *Inf. and Comp.*, 208(6):677–693, 2010.

[11] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. PRISM-games: A model checker for stochastic multi-player games. In *Proc. of TACAS'13*, volume 7795 of *LNCS*, pages 185–191. Springer, 2013.

[12] E.M. Clarke, D.E. Long, and K.L. McMillan. Compositional model checking. In *Proc. of LICS'89*, pages 353–362. IEEE Comput. Soc. Press, 1989.

[13] B. Devereux. Compositional reasoning about aspects using alternating-time logic. In *Proc. of FOAL'03*, pages 45–50, 2003.

[14] C. Dima and F.L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.

[15] N. Fijalkow, B. Maubert, A. Murano, and M.Y. Vardi. Assume-guarantee synthesis for prompt linear temporal logic. In *Proc. of IJCAI'20*, pages 117–123. ijcai.org, 2020.

[16] D.P. Guelev, C. Dima, and C. Enea. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *J. Appl. Non-Classical Log.*, 21(1):93–131, 2011.

[17] T.A. Henzinger, S. Qadeer, and S.K. Rajamani. You assume, we guarantee: Methodology and case studies. In *Proc. of CAV'98*, volume 1427 of *LNCS*, pages 440–451. Springer, 1998.

[18] C.A.R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[19] X. Huang and R. van der Meyden. Symbolic model checking epistemic strategy logic. In *Proc. of AAAI'14*, pages 1426–1432, 2014.

[20] W. Jamroga and J. Dix. Model checking ATL_{ir} is indeed Δ_2^P-complete. In *Proc. of EUMAS'06*, volume 223 of *CEUR Workshop Proc.*, 2006.

[21] W. Jamroga, M. Knapik, D. Kurpiewski, and Ł. Mikulski. Approximate verification of strategic abilities under imperfect information. *Artif. Int.*, 277, 2019.

[22] W. Jamroga, W. Penczek, T. Sidoruk, P. Dembiński, and A.W. Mazurkiewicz. Towards partial order reductions for strategic ability. *J. Artif. Intell. Res.*, 68:817–850, 2020.

[23] C.B. Jones. Specification and design of (parallel) programs. In *Proc. of IFIP'83*, pages 321–332. North-Holland/IFIP, 1983.

[24] D. Kurpiewski, W. Pazderski, W. Jamroga, and Y. Kim. STV+Reductions: Towards practical verification of strategic ability using model reductions. In *Proc. of AAMAS'21*, pages 1770–1772. ACM, 2021.

[25] M.Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu. Assume-guarantee verification for probabilistic systems. In *Proc. of TACAS'10*, volume 6015 of *LNCS*, pages 23–37. Springer, 2010.

[26] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: An open-source model checker for the verification of multi-agent systems. *Int. J. Soft. Tools Tech. Trans.*, 19(1):9–30, 2017.

[27] A. Lomuscio, B. Strulo, N.G. Walker, and P. Wu. Assume-guarantee reasoning with local specifications. In *Proc. of ICFEM'10*, volume 6447 of *LNCS*, pages 204–219. Springer, 2010.

[28] A. Lomuscio, B. Strulo, N.G. Walker, and P. Wu. Assume-guarantee reasoning with local specifications. *Int. J. Found. Comput. Sci.*, 24(4):419–444, 2013.

[29] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comp. Log.*, 15(4):1–42, 2014.

[30] S.S. Owicki and D. Gries. Verifying properties of parallel programs: An axiomatic approach. *Commun. ACM*, 19(5):279–285, 1976.

[31] A. Pnueli. In transition from global to modular temporal reasoning about programs. In *Logics and Models of Concurrent Systems*, volume 13 of *NATO ASI Series*, pages 123–144. Springer, 1984.

[32] P.Y. Schobbens. Alternating-time logic with imperfect recall. *Electr. Not. Theor. Comput. Sci.*, 85(2):82–93, 2004.