Model Checking Strategic Ability Why, What, and Especially: How? (Extended Abstract)

Wojciech Jamroga[1]
Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
w.jamroga@ipipan.waw.pl

Abstract

Automated verification of discrete-state systems has been a hot topic in computer science for over 35 years. Model checking of temporal and strategic properties is one of the most prominent and most successful approaches here. In this talk, I present a brief introduction to the topic, and mention some relevant properties that one might like to verify this way. Then, I describe some recent results on approximate model checking and model reductions, which can be applied to facilitate verification of notoriously hard cases.

2012 ACM Subject Classification Computing methodologies → Multi-agent systems

Keywords and phrases model checking, strategic ability, alternating-time temporal logic, imperfect information games, approximate verification, model reductions

Digital Object Identifier 10.4230/LIPIcs.TIME.2018.3

Acknowledgements This extended abstract reports joint research with Francesco Belardinelli, Rodica Condurache, Piotr Dembiński, Cătălin Dima, Michał Knapik, Damian Kurpiewski, Antoni Mazurkiewicz, Łukasz Mikulski, and Wojciech Penczek. I gratefully acknowledge their contribution in what is presented below.

1 Multi-Agent Systems and Strategic Ability

More and more systems involve social as much as technological aspects, and even those that focus on technology are often based on distributed components exhibiting self-interested, goal-directed behavior. Moreover, the components act in environments characterized by incomplete information and uncertainty. The field of *multi-agent systems* studies the whole spectrum of phenomena ranging from agent architectures to communication and coordination in agent groups to agent-oriented software engineering. The theoretical foundations are mainly based on game theory and formal logic.

Many relevant properties of multi-agent systems refer to *strategic abilities* of agents and their groups. In particular, most functionality requirements can be specified as the ability of the authorized users to achieve their legitimate goals. At the same time, many security properties can be phrased in terms of the inability of unauthorized users to compromise the system. Properties of this kind can be conveniently specified in *modal logics of strategic ability*, of which alternating-time temporal logic (**ATL**) [2] is probably the most popular.

ATL modalities, possibly in combination with epistemic operators, allow e.g. to capture different flavors of coercion-resistance in voting systems [23, 17]. A simple hierarchy of

[1] The author acknowledges the support of the National Centre for Research and Development (NCBR), Poland, under the PolLux project VoteVerif (POLLUX-IV/1/2016).

25th International Symposium on Temporal Representation and Reasoning (TIME 2018).
Editors: Natasha Alechina, Kjetil Nørvåg, and Wojciech Penczek; Article No. 3; pp. 3:1–3:10
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

$$CR_1 \equiv \neg\langle\langle Coercer\rangle\rangle \, \mathrm{G} \, \big((\mathsf{closed} \wedge \bigwedge_{v\in A} \neg\mathsf{vote}_{v,1}) \rightarrow \mathsf{K}_{Coercer}(\bigvee_{v\in A} \neg\mathsf{vote}_{v,1})\big)$$
$$CR_2 \equiv \neg\langle\langle Coercer\rangle\rangle \, \mathrm{G} \, \big((\mathsf{closed} \wedge \bigwedge_{v\in A} \neg\mathsf{vote}_{v,1} \rightarrow \bigvee_{v\in A} \mathsf{K}_{Coercer}(\neg\mathsf{vote}_{v,1})\big)$$
$$CR_3 \equiv \neg\langle\langle Coercer\rangle\rangle \, \mathrm{G} \, \big((\mathsf{closed} \wedge \bigvee_{v\in A} \neg\mathsf{vote}_{v,1} \rightarrow \mathsf{K}_{Coercer}(\bigvee_{v\in A} \neg\mathsf{vote}_{v,1})\big)$$
$$CR_4 \equiv \neg\langle\langle Coercer\rangle\rangle \, \mathrm{G} \, \big((\mathsf{closed} \wedge \bigvee_{v\in A} \neg\mathsf{vote}_{v,1} \rightarrow \bigvee_{v\in A} \mathsf{K}_{Coercer}(\neg\mathsf{vote}_{v,1})\big)$$

Figure 1 Variants of coercion-resistance for coercion of multiple voters [17]

coercion-related specifications is presented in Figure 1. CR_1 expresses that the coercer cannot force all the agents in A to vote for candidate 1, or else he will know, at the closing of the election, that at least one of them disobeyed. CR_2 captures a stronger property: if someone in A disobeyed, the coercer will know the identity of at least one such agent. We leave the interpretation of CR_3 and CR_4 to the interested reader.

2 Practical Verification of Strategic Abilities

In the last 15 years, there has been a large number of works that study the syntactic and semantic variants of **ATL** for agents with imperfect information, cf. [5, 1] for an overview. The contributions are mainly theoretical, and concern the conceptual soundness of a given semantics, meta-logical properties, and the complexity of model checking and satisfiability problems. However, there is relatively little research on practical algorithms for verification.

This is somewhat easy to understand, since model checking of **ATL** variants with imperfect information is $\mathbf{\Delta_2^P}$- to **PSPACE**-complete for memoryless strategies [22, 4] and **EXPTIME**-complete to undecidable for agents with perfect recall [13, 14]. Moreover, the imperfect information semantics of **ATL** does not admit fixpoint characterizations [6, 11, 12], which makes incremental synthesis of strategies difficult to achieve. Experimental studies based on heuristic approaches [8, 15, 21, 9, 7] have also confirmed that the problem is hard, and dealing with it requires innovative techniques.

In this talk, I present several very recent attempts at overcoming the complexity barrier for model checking of $\mathbf{ATL}_{\mathrm{ir}}$, i.e., the **ATL** variant for memoryless strategies with imperfect information. The new techniques include approximate verification by fixpoint computation of upper- and lower bounds, model reduction based on locally defined model equivalence, and partial order reduction for simple strategic abilities in asynchronous systems. The main ideas are presented in the following sections, in a rather rudimentary form. For technical details, the reader is referred to the original papers [16, 18, 3, 19].

3 Approximate Model Checking

The proposal in [16, 18] is based on the idea that, instead of the exact model checking, it may suffice to provide a lower and an upper bound for the output. Given a formula φ, we construct two translations $LB(\varphi)$ and $UB(\varphi)$ such that $LB(\varphi) \Rightarrow \varphi \Rightarrow UB(\varphi)$. That is, if $LB(\varphi)$ is verified as true, then the original formula φ must also hold in the given model. Conversely, if $UB(\varphi)$ evaluates to false, then φ must also be false.

#cards	#states	Approximate verification				Exact verification
		tgen	lower	upper	match	
4	11	0.0007	0.00007	0.00004	100%	0.12
8	346	0.011	0.0008	0.0003	100%	2.42 h*
12	12953	0.73	0.07	0.01	100%	timeout
16	617897	35.19	348.37	0.72	100%	timeout
20*	2443467	132.00	8815.73	4.216	100%	timeout

Figure 2 Experimental results: solving endplay in bridge, formula $\langle\langle \mathbf{S} \rangle\rangle_{\mathrm{ir}} \mathrm{F}\mathsf{win}$ [18]

The construction of the upper bound is straightforward: instead of checking existence of an imperfect information strategy, we look for a perfect information strategy that obtains the same goal. If the latter is false, the former must be false too. The lower bound is defined by a fixpoint expression in alternating epistemic mu-calculus, with a nonstandard next-step strategic modality $\langle A \rangle^\bullet$ such that: (i) agents in A look for a short-term strategy that succeeds from the "common knowledge" neighborhood of the current state (rather than in the "everybody knows" neighborhood), and (ii) they are allowed to "steadfastly" pursue their goal in a variable number of steps within the indistinguishability class. Formally, the upper and the lower bounds are derived from φ through the following translations:

$$
\begin{array}{ll}
LB(p) = p, & UB(p) = p, \\
LB(\neg\phi) = \neg UB(\phi), & UB(\neg\phi) = \neg LB(\phi), \\
LB(\phi \wedge \psi) = LB(\phi) \wedge LB(\psi), & UB(\phi \wedge \psi) = UB(\phi) \wedge UB(\psi), \\
LB(\langle A \rangle \phi) = \langle A \rangle LB(\phi), & UB(\langle A \rangle \phi) = E_A \langle\langle A \rangle\rangle_{\mathrm{Ir}} \mathrm{X}\, UB(\phi), \\
LB(\langle\langle A \rangle\rangle \mathrm{G} \phi) = \nu Z.(C_A LB(\phi) \wedge \langle A \rangle^\bullet Z), & UB(\langle\langle A \rangle\rangle \mathrm{G} \phi) = E_A \langle\langle A \rangle\rangle_{\mathrm{Ir}} \mathrm{G}\, UB(\phi), \\
LB(\langle\langle A \rangle\rangle \psi \,\mathrm{U}\, \phi) = \mu Z.\big(E_A LB(\phi) \vee (C_A LB(\psi) \wedge \langle A \rangle^\bullet Z)\big) & UB(\langle\langle A \rangle\rangle \psi \,\mathrm{U}\, \phi) = E_A \langle\langle A \rangle\rangle_{\mathrm{Ir}} UB(\psi) \,\mathrm{U}\, UB(\phi)
\end{array}
$$

▶ **Theorem 1** ([16]). *For every pointed model* (M, q) *and* $\mathbf{ATL}_{\mathrm{ir}}$ *formula* φ*:*

$$M, q \models LB(\varphi) \implies M, q \models \varphi \implies M, q \models UB(\varphi).$$

The effectiveness of the approximations has been evaluated experimentally on a number of benchmarks. The results for a scalable scenario of *Bridge endplay* are presented in Figure 2. The results in each row are averaged over 20 randomly generated instances, except for ($\star$) where only 1 hand-crafted instance was used. All the tests were conducted on a computer with an Intel Core i7-6700 CPU with dynamic clock speed of 2.60–3.50 GHz, 32 GB RAM, running 64bit Ubuntu 16.04 Linux. The running times are given in seconds. *Timeout* indicates that the process did not terminate in 48 hours. The computation of the lower and upper approximations was done with a straightforward implementation (in Python 3) of the fixpoint model checking algorithm. The exact $\mathbf{ATL}_{\mathrm{ir}}$ model checking was done with MCMAS 1.3.0 [20].

Notice that the results in Figure 2 have been obtained by a completely straigghtforward implementation of the lower/upper bound algorithms. After a simple optimization of data structures (based on the technique of *merge-find sets*) and operations on the data, the algorithms were able to generate and verify a model with over 70 million states in less than 75 minutes, cf. Figure 3. Perhaps more importantly, the verification of the handpicked $(5,5)^\star$ model (which marked the limit of our capability without the optimization) ran almost 3000 times (!) faster than with the straightforward implementation. This strongly suggests that the potential for further improvement is still large.

#cards	#states	Approximate verification				Exact
		tgen	lower	upper	match	verification
4	11	<0.0001	<0.0001	<0.0001	100%	0.12
8	346	<0.0001	<0.0001	<0.0001	100%	2.42 h*
12	12953	0.06	<0.0001	<0.0001	100%	timeout
16	617897	4.64	0.56	0.26	100%	timeout
20*	2443467	34.00	3.0	2.0	100%	timeout
20	1.5 e7	124.00	8.5	6.0	100%	timeout
24*	7 e7	3779.00	667.0	78.0	100%	timeout

Figure 3 Experimental results for the optimised approximate verification [18]

4 Bisimulation-Based Model Reduction

The main source of hardness in model checking for strategic ability is the size of the model. First, the space of available strategies is at least exponential in the number of states and transitions. To make it even worse, there is (as of now) no method to factorize the model so that the algorithm would look for optimal substrategies independently, and then combine them into a winning strategy. Secondly, the explicit state model is typically exponential in size with respect to a higher level description, e.g., by means of local automata or concurrent programs. This means that realistic models are huge, and their strategy spaces are enormous. In consequence, an effective model reduction can offer invaluable help in turning a hopeless verification task into a feasible one.

One way to obtain such reductions is to identify a suitable notion of model equivalence that preserves the logic. Then, whenever the need for verification arises, we can look for a smaller model that is provably equivalent to the input model. Locally definable model equivalences for logics of strategies are usually called *alternating bisimulations*. The first variant of alternating bisimulation for **ATL** with imperfect information has been recently proposed in [3]. The main definitions and preservation theorems are summarized below.

Strategy simulators. Let M, M' be two models of $\mathbf{ATL}_{\mathrm{ir}}$. A *simulator of partial strategies for coalition* A from M into M' is a family $\mathbf{ST}$ of functions $\mathbf{ST}_{Q,Q'} : PStr_A(Q) \to PStr_A(Q')$ for some subsets of states $Q \subseteq St$ in model M and $Q' \subseteq St'$ in model M'. Intuitively, every $\mathbf{ST}_{Q,Q'}$ maps each partial strategy σ_A defined on set Q in M into a "corresponding" strategy σ'_A defined on Q' in M'. Typically, we will map strategies between the common knowledge neighborhoods of "bisimilar" states in M and M'. We formalize this idea as follows. Let $R \subseteq St \times St'$ be a relation between states in M and M'. A *simulator of partial strategies for coalition* A *with respect to relation* R is a family $\mathbf{ST}$ of functions $\mathbf{ST}_{C_A(q),C'_A(q')} : PStr_A(C_A(q)) \to PStr_A(C'_A(q'))$ such that qRq'.

Simulation and bisimulation. Let M, M' be two models defined on the same set $\mathbb{A}\mathrm{gt}$ of agents, and $A \subseteq \mathbb{A}\mathrm{gt}$ be a coalition. A relation $\Rrightarrow_A \subseteq St \times St'$ is a *simulation for* A iff

1. There exists a simulator $\mathbf{ST}$ of partial strategies for A w.r.t. $\Rrightarrow_A$, such that $q \Rrightarrow_A q'$ implies that:
   - **a.** $\pi(q) = \pi'(q')$;
   - **b.** for every $i \in A$, $r' \in St'$, if $q' \sim'_i r'$ then for some $r \in St$, $q \sim_i r$ and $r \Rrightarrow_A r'$.
   - **c.** For every states $r \in C_A(q)$, $r' \in C'_A(q')$ such that $r \Rrightarrow_A r'$, for every partial strategy $\sigma_A \in PStr_A(C_A(q))$, and every state $s' \in succ(r', \mathbf{ST}(\sigma_A))$, there exists a state $s \in succ(r, \sigma_A)$ such that $s \Rrightarrow_A s'$.

2. If $q_1 \Rrightarrow_A q'$ and $q_2 \Rrightarrow_A q'$, then $C_A(q_1) = C_A(q_2)$.

That is, state q' can only simulate q if (1a) q and q' agree on the interpretation of atoms; (1b) q simulates the epistemic transitions from q'; and (1c) for every partial strategy σ_A, defined on the common knowledge neighborhood $C_A(q)$, we are able to find some partial strategy $\mathbf{ST}(\sigma_A)$ (the same for all states in $C_A(q)$) such that the transition relations $\xrightarrow{\mathbf{ST}(\sigma_A)}$ and $\xrightarrow{\sigma_A}$ commute with the simulation relation $\Rrightarrow_A$. Moreover, (2) multiple states simulated by the same q' must be in the same common knowledge neighborhood.

Relation $\Leftrightarrow_A$ is a *bisimulation* iff both $\Leftrightarrow_A$ and its converse $\Leftrightarrow_A^{-1} = \{(q', q) \mid q \Leftrightarrow_A q'\}$ are simulations.

▶ **Theorem 2** (Preservation Theorem for A-$\mathbf{ATL}_{\mathrm{ir}}$ [3]). *If $\Leftrightarrow_A$ is a bisimulation for A and $q \Leftrightarrow_A q'$, then for every formula φ that contains only agents from A inside strategic modalities, we have:*

$$M, q \models \varphi \quad \textit{if and only if} \quad M', q' \models \varphi.$$

▶ **Theorem 3** (Preservation Theorem for $\mathbf{ATL}_{\mathrm{ir}}$ [3]). *If $\Leftrightarrow$ is a bisimulation for every $A \subseteq \mathbb{A}\mathrm{gt}$, and $q \Leftrightarrow q'$, then for every formula φ we have that:*

$$M, q \models \varphi \quad \textit{if and only if} \quad M', q' \models \varphi.$$

The bisimulation provides a strong notion of model equivalence, since it preserves the truth values of all $\mathbf{ATL}_{\mathrm{ir}}$ formulae. Moreover, it can lead to *very significant model reductions*. As an example, Figure 4a presents a fragment of the simple model of voting and coercion [18] for 1 coercer, 2 voters, and 2 candidates. A bisimilar, much sparser model is presented in Figure 4b.

5 Partial-Order Reduction

The bisimulation-based reduction can result in a significant trimming of states and transitions. However, the conditions of the bisimulation are quite strong, which means that the method has somewhat limited applicability. Moreover, the reduced model and the equivalence must be crafted and proved by hand. An automated method for reduction of models *that arise due to interleaving of asynchronously executed actions* has been recently proposed in [19].

Many real-life systems are inherently asynchronous, and do not operate on a global clock that perfectly synchronizes the atomic steps of all the components. As an example, consider robots interacting in an environment with faulty communication or non-negligible delays in execution of actions. No less importantly, many systems that are synchronous at the implementation level (say, the level of the virtual machine) can be more conveniently modeled as asynchronous on a more abstract level, because when we abstract away the implementation details it is not clear anymore how transitions initiated by different agents are exactly arranged in a particular temporal order. For instance, the actual implementation of a soccer match in the simulated RoboCup competition can be executed on a single computer with a global clock ticking every 0.3 ns, but the corresponding synchronous model would be huge and in consequence useless for analysis. Instead, one can remove a lot of unnecessary details by assuming that the players execute their actions asynchronously – without clear temporal relationship between their execution times – and synchronize only when a particular event has to be executed *jointly*.

In many scenarios, both aspects combine. For example, when modeling an election, one must take into account both the truly asynchronous nature of events happening at different

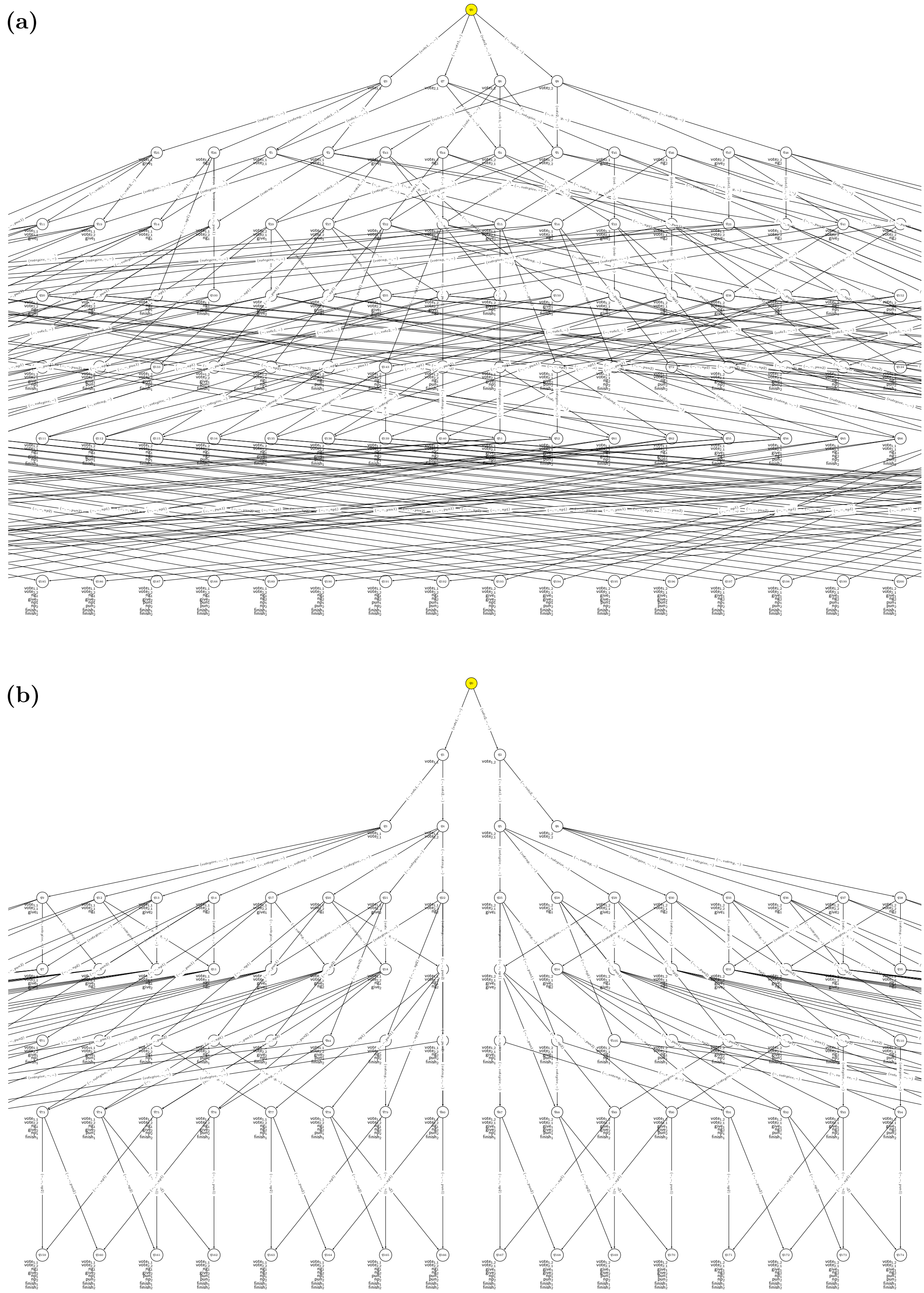

Figure 4 Bisimulation-based reduction of a simple voting model: before and after the reduction

polling stations, and the best level of granularity for modeling the events happening within a single polling station.

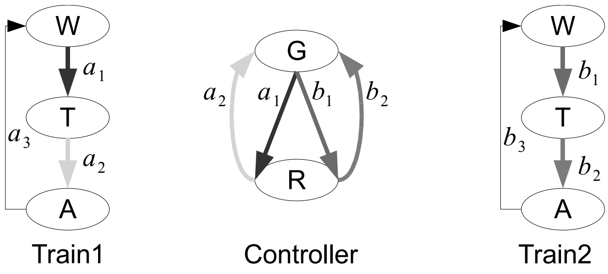

Figure 5 Asynchronous multi-agent system for the Trains, Gate and Controller benchmark

In [19], we have made the first step towards strategic analysis of such systems. Most importantly, we adapted *partial order reduction (POR)* to model checking of strategic abilities for agents with imperfect information. In fact, we showed that the most efficient variant of POR, defined for linear time logic **LTL**, can be applied almost directly. Interestingly, the scheme does *not* work for verification of agents with perfect information.

Algorithm for partial-order reduction. Given a collection of local automata $\mathcal{S}$ with its underlying model M, the reduction proceeds by iterative unfolding of $\mathcal{S}$ into its reduced model M'. The unfolding is done by means of the following Depth-First Search (DFS) algorithm:

1. Identify the set $en(q_n) \subseteq Act$ of enabled actions.
2. Heuristically select a subset $E(q_n) \subseteq en(q_n)$ of possible actions (see below).
3. For any action $a \in E(q_n)$, compute the successor state q' such that $q_n \xrightarrow{a} q'$, and add q' to the stack, thus generating the path $\pi' = q_0 a_0 q_1 a_1 \cdots q_n a q'$. Recursively proceed to explore the submodel originating at q'.
4. Remove q_n from the stack.

The algorithm begins with the stack consisting solely of the initial state of M, and terminates when the stack is empty.

Heuristics. Let $A \subseteq \mathbb{A}\mathrm{gt}$. The conditions **C1** – **C3** below define a heuristics for the selection of $E(q) \subseteq en(q)$ in the DFS algorithm.

C1 Along each path π in M that starts at q, each action that is dependent on an action in $E(q)$ cannot be executed in π without an action in $E(q)$ is executed first in π. Formally, $\forall \pi \in \Pi_M(q)$ such that $\pi = q_0 a_0 q_1 a_1 \dots$ with $q_0 = q$, and $\forall b \in Act$ such that $(b, c) \notin I_A$ for some $c \in E(q)$, if $a_i = b$ for some $i \geq 0$, then $a_j \in E(q)$ for some $j < i$.
C2 If $E(q) \neq en(q)$, then $E(q) \subseteq Invis_A$.
C3 For every cycle in M' there is at least one node q in the cycle for which $E(q) = en(q)$, i.e., for which all the successors of q are expanded.

Preservation theorems. Let $\mathcal{S}$ be a collection of local automata representing an asynchronous multi-agent system. $I_\emptyset$ is the standard independence of actions, used in partial-order reduction for **LTL** [10]. I_A is a slightly modified variant of $I_\emptyset$, that treats all the actions of agents in A as visible. The following theorems show that the **LTL** partial-order reduction can be directly (or almost directly) applied to $\mathbf{sATL}^*_{\mathrm{ir}}$, i.e., the fragment of $\mathbf{ATL}^*_{\mathrm{ir}}$ without nested strategic modalities and the temporal operator X.

▶ **Theorem 4** ([19]). *Let M be the model of $\mathcal{S}$, and M' be the reduced model generated by DFS with the choice of $E(q')$ for $q' \in St'$ given by conditions* **C1, C3** *and the independence relation $I_\emptyset$. Then, M' satisfies exactly the same formulae of* $\mathbf{sATL}^*_{\mathrm{ir}}$ *as M under the concurrency-fairness assumption.*

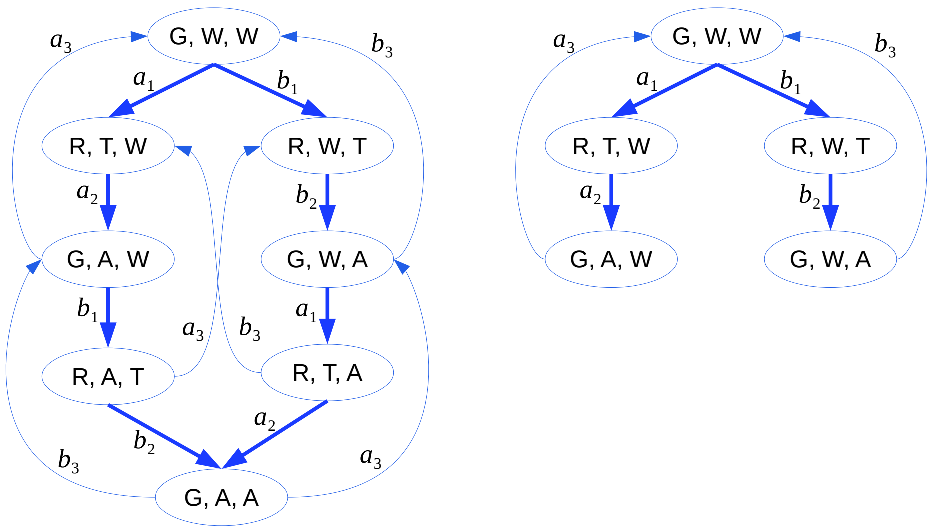

Figure 6 Interleaved interpreted systems for TGC: (a) full model, (b) reduced model. Visible transitions are depicted by bold arrows

▶ **Theorem 5** ([19]). *Let $A \subseteq \mathbb{A}\text{gt}$, M be the model of S, and M' the reduced model generated by DFS with the choice of $E(q')$ given by conditions* **C1, C2, C3** *and the independence relation I_A. Then, M' satisfies exactly the same formulae of A-$\mathbf{sATL}^*_{\text{ir}}$ as M without the concurrency-fairness assumption.*[2]

How big is the gain? As an example, consider the well known TGC benchmark (Trains, Gate, and Controller). The local automata representing the system for $k = 2$ trains are shown in Figure 5, and the full logical model in Figure 6a. The reduced model obtained by POR is depicted in Figure 6b (same as for the **LTL** partial-order reduction). It is easy to see that the reduced state space is exponentially smaller than the size of the full model.

Of course, such optimistic outcomes are not guaranteed for all asynchronous agent systems. Still, it is important to note that $\mathbf{ATL}_{\text{ir}}$ model checking is **NP**-hard *in the size of the model* (and not the size of the representation), and all the attempts at actual algorithms so far run in exponential time. So, even a linear reduction of the state space is likely to produce an exponential improvement of the performance.

Perfect information. The reduction scheme does not work for the standard variant of alternating-time logic, based on perfect information strategies:

▶ **Theorem 6** ([19]). *The analogues of Theorems 4 and 5 do not hold for* $\mathbf{sATL}_{\text{Ir}}$.

This negative result is especially interesting because, until now, virtually all the results have been in favor of solving games with perfect information.

6 Conclusions

Verification by model checking is one of the top success stories in computer science and AI. Many important properties of multi-agent systems are underpinned by the ability of some agents (or groups) to achieve a given goal. However, model checking of strategic ability in

[2] where A-$\mathbf{sATL}^*_{\text{ir}}$ is the fragment of $\mathbf{sATL}^*_{\text{ir}}$ containing only agents from A in strategic modalities.

realistic systems is a notoriously hard problem. In this short paper, I have tried to summarize some of the very recent developments that can contribute to overcoming the complexity barrier, and extending the scope of formal verification.

References

1. T. Ågotnes, V. Goranko, W. Jamroga, and M. Wooldridge. Knowledge and ability. In H.P. van Ditmarsch, J.Y. Halpern, W. van der Hoek, and B.P. Kooi, editors, *Handbook of Epistemic Logic*, pages 543–589. College Publications, 2015.
2. R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002. doi:10.1145/585265.585270.
3. F. Belardinelli, R. Condurache, C. Dima, W. Jamroga, and A.V. Jones. Bisimulations for verification of strategic abilities with application to ThreeBallot voting protocol. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2017*, pages 1286–1295. IFAAMAS, 2017.
4. N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In M. Dastani, K. Hindriks, and J.-J. Meyer, editors, *Specification and Verification of Multi-Agent Systems*, pages 125–159. Springer, 2010.
5. N. Bulling, V. Goranko, and W. Jamroga. Logics for reasoning about strategic abilities in multi-player games. In J. van Benthem, S. Ghosh, and R. Verbrugge, editors, *Models of Strategic Reasoning. Logics, Games, and Communities*, volume 8972 of *Lecture Notes in Computer Science*, pages 93–136. Springer, 2015. doi:10.1007/978-3-662-48540-8.
6. N. Bulling and W. Jamroga. Alternating epistemic mu-calculus. In *Proceedings of IJCAI-11*, pages 109–114, 2011.
7. S. Busard. *Symbolic Model Checking of Multi-Modal Logics: Uniform Strategies and Rich Explanations*. PhD thesis, Universite Catholique de Louvain, 2017.
8. 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 *Lecture Notes in Computer Science*, pages 27–42. Springer, 2014. doi:10.1007/978-3-319-11737-9_3.
9. S. Busard, C. Pecheur, H. Qu, and F. Raimondi. Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Information and Computation*, 242:128–156, 2015. doi:10.1016/j.ic.2015.03.014.
10. E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
11. C. Dima, B. Maubert, and S. Pinchinat. The expressive power of epistemic μ-calculus. *CoRR*, abs/1407.5166, 2014.
12. C. Dima, B. Maubert, and S. Pinchinat. Relating paths in transition systems: The fall of the modal mu-calculus. In *Proceedings of Mathematical Foundations of Computer Science (MFCS)*, volume 9234 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2015. doi:10.1007/978-3-662-48057-1_14.
13. C. Dima and F.L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.
14. D.P. Guelev, C. Dima, and C. Enea. An alternating-time temporal logic with knowledge, perfect recall and past: axiomatisation and model-checking. *Journal of Applied Non-Classical Logics*, 21(1):93–131, 2011.
15. X. Huang and R. van der Meyden. Symbolic model checking epistemic strategy logic. In *Proceedings of AAAI Conference on Artificial Intelligence*, pages 1426–1432, 2014.
16. W. Jamroga, M. Knapik, and D. Kurpiewski. Fixpoint approximation of strategic abilities under imperfect information. In *Proceedings of the 16th International Conference*

on Autonomous Agents and Multiagent Systems (AAMAS), pages 1241–1249. IFAAMAS, 2017.

17 W. Jamroga, M. Knapik, and D. Kurpiewski. Model checking the selene e-voting protocol in multi-agent logics. In *Proceedings of the 3rd International Joint Conference on Electronic Voting (E-VOTE-ID)*, Lecture Notes in Computer Science. Springer, 2018. To appear.

18 W. Jamroga, M. Knapik, D. Kurpiewski, and Łukasz Mikulski. Approximate verification of strategic abilities under imperfect information. *Artificial Intelligence*, 2018. To appear.

19 W. Jamroga, W. Penczek, P. Dembiński, and A. Mazurkiewicz. Towards partial order reductions for strategic ability. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 156–165. IFAAMAS, 2018.

20 A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer*, 2015. Availabe online. `doi:10.1007/s10009-015-0378-x`.

21 J. Pilecki, M.A. Bednarczyk, and W. Jamroga. Synthesis and verification of uniform strategies for multi-agent systems. In *Proceedings of CLIMA XV*, volume 8624 of *Lecture Notes in Computer Science*, pages 166–182. Springer, 2014.

22 P. Y. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2):82–93, 2004.

23 M. Tabatabaei, W. Jamroga, and Peter Y. A. Ryan. Expressing receipt-freeness and coercion-resistance in logics of strategic ability: Preliminary attempt. In *Proceedings of the 1st International Workshop on AI for Privacy and Security, PrAISe@ECAI 2016*, pages 1:1–1:8. ACM, 2016. `doi:10.1145/2970030.2970039`.