STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions

Demonstration Track

Damian Kurpiewski
Institute of Computer Science, Polish Academy of Sciences
d.kurpiewski@ipipan.waw.pl

Witold Pazderski
Institute of Computer Science, Polish Academy of Sciences
w.pazderski@ipipan.waw.pl

Wojciech Jamroga
Institute of Computer Science, Polish Academy of Sciences and University of Luxembourg
w.jamroga@ipipan.waw.pl

Yan Kim
University of Luxembourg
yan.kim@uni.lu

ABSTRACT

We present a substantially expanded version of our tool **STV** for strategy synthesis and verification of strategic abilities. The new version adds user-definable models and support for model reduction through partial order reduction and checking for bisimulation.

KEYWORDS

formal methods; model checking; alternating-time temporal logic

ACM Reference Format:
Damian Kurpiewski, Witold Pazderski, Wojciech Jamroga, and Yan Kim. 2021. **STV+Reductions**: Towards Practical Verification of Strategic Ability Using Model Reductions: Demonstration Track. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 3 pages.

1 INTRODUCTION

Formal analysis of multi-agent systems is becoming increasingly important as the procedures, protocols, and technology that surround us get more and more complex. *Alternating-time temporal logic* ATL [3, 4, 43] is probably the most popular logic to describe interaction in MAS. Formulas of ATL allow to express statements about what agents (or groups of agents) can achieve. For example, $\langle\langle taxi\rangle\rangle \mathrm{G}\neg\mathsf{fatality}$ says that the autonomous cab can drive in such a way that nobody is ever killed, and $\langle\langle taxi, passg\rangle\rangle \mathrm{F}\,\mathsf{destination}$ expresses that the cab and the passenger have a joint strategy to arrive at the destination, no matter what any other agents do.

Algorithms and tools for verification of such properties have been in development for over 20 years [1, 2, 7, 8, 11, 13–15, 21, 26, 32, 33, 37, 38, 40]. Unfortunately, model checking of agents with imperfect information is Δ_2^{P}- to PSPACE-complete for memoryless strategies [9, 22, 43] and EXPTIME-complete to undecidable for agents with perfect recall [18, 20]; also, the problem does not admit simple incremental solutions [10, 16, 17]. This has been confirmed in experiments [11, 12, 21, 26, 40] and case studies [23, 25, 28].

Much of the complexity is due to the size of the model, and in particular to state space explosion [5]. To address the problem, we have extended our experimental tool **STV (STrategic Verifier)** [33] with support for *model reductions*. Two methods are used: (i) checking for equivalence of models according to a handcrafted relation of *A-bisimulation* [6], and (ii) fully automated *partial order reduction (POR)* [29, 31]. We also add a simple model specification language that allows the user to define their own inputs for verification, which was not available in the previous version [33].

The purpose of the extension is twofold. First, it should facilitate practical verification of MAS, as the theoretical and experimental results for POR and bisimulation-based reduction suggest [6, 31]. No less importantly, it serves a pedagogical objective, as we put emphasis on visualisation of the reductions, so that the tool can be also used in the classroom to show how the reduction works. Finally, checking strategic bisimulation by hand is difficult and prone to errors; here, the user can both see the idea of the bisimulation, and automatically check if it is indeed correct.

2 APPLICATION DOMAIN

STV+Reductions is aimed at verification of agents' abilities – in particular, synthesis of memoryless imperfect information strategies that guarantee a given temporal goal. This includes both model checking of *functionality requirements* (understood as the ability of legitimate users to achieve their goals), and *security properties* defined by the inability of an intruder to compromise the system.

A good example of a specific domain is formal verification of voting procedures and elections, with a number of classical requirements, such as *election integrity*, *ballot secrecy*, *receipt-freeness*, and *voter-verifiability* [42, 44]. Some recent case studies [23, 25, 28] have shown that practical verification of such scenarios is still outside of reach. Some tools do not support intuitive specification and validation of models; some others have limited property specification languages. In all cases, the state-space explosion is a major obstacle that prevents verification of anything but toy models.

3 SCENARIOS

The new version of **STV** provides a flexible specification language for asynchronous models. The following examples are included: Train-Gate-Controller (TGC) [2, 31, 45], Two-Stage Voting [6], and

---

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

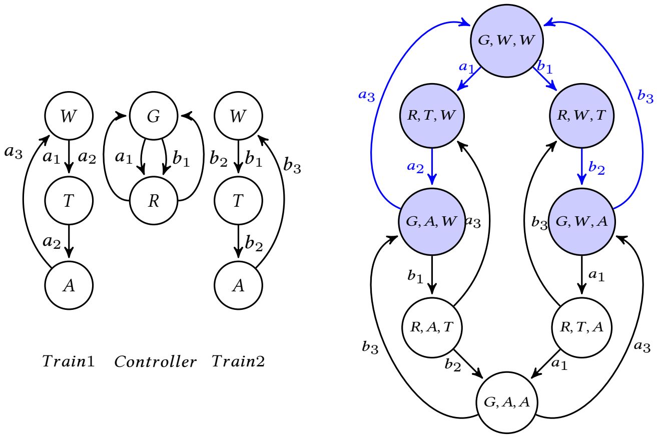

Figure 1: Trains, Gate, and Controller benchmark (TGC): asynchronous MAS (left); full and reduced model (right).

Asynchronous Simple Voting [31]. Some built-in synchronous models are also included, such as TianJi [37], Castles [41], Bridge Endplay [24], and Drones [27].

4 FORMAL BACKGROUND

Models. The main part of the input is given by an *asynchronous multi-agent system (AMAS)* [30, 31, 35], i.e., a network of local automata (one automaton per agent). From the AMAS, the global model is generated, where nodes are tuples of local states. The knowledge/uncertainty of an agent is defined by the agent's local state. An example AMAS is shown in Figure 1(left). The global model generated from the AMAS is shown in Figure 1(right).

Strategies. A strategy is a conditional plan that specifies what the agent(s) are going to do in every possible situation. Here, we consider the case of *imperfect information memoryless strategies*, represented by functions from the agent's local states (formally, abstraction classes of its indistinguishability relations) to its available actions. The *outcome* of a strategy from state q consists of all the infinite paths starting from q and consistent with the strategy.

Formulas. Given a model M and a state q in the model, the formula $\langle\langle A\rangle\rangle\varphi$ holds in (M, q) iff there exists a strategy for A that makes φ true on all the outcome paths starting from any state indistinguishable from q. For more details, we refer the reader to [4, 43].

Model reduction and bisimulation. State space explosion is a major factor that prevents practical model checking [5]. A possible way out is *model reduction*, i.e., using a smaller equivalent model for verification instead of the original one. A suitable notion of *A-bisimulation* has been proposed in [6]. Unfortunately, synthesizing a reduced A-bisimilar model is at least as hard as the verification itself [6]. However, checking if a handcrafted relation is an A-bisimulation can be done in polynomial time, which offers valuable help especially for larger models.

Partial-order reduction. A fully automated model reduction is possible if the state space explosion is due to asynchronous interleaving of agents' actions. The method is called *partial order reduction*, and has been been recently extended to verification of strategic abilities under imperfect information [31]. The reduced model for the TGC scenario is highlighted in blue color in Figure 1(right).

5 TECHNOLOGY

STV+Reductions does *explicit-state model checking*. That is, the global states and transitions of the model are represented explicitly in the memory of the verification process. The tool includes the following new functionalities.

User-defined input. The user can load and parse the input specification from a text file that defines: the local automata in the AMAS, the formula to be verified, the propositional variables, persistent propositions, agent names relevant for POR, and/or the mapping for bisimulation checking. Based on that, the global model is generated and displayed in the GUI and can be verified by means of *fixpoint approximation* [26] or *dominance-based strategy search* [34]. When using partial-order reduction, the reduced model is also displayed, and highlighted in the full model.

Partial-order reduction. The fully automated reduction method is based on POR [39] and implemented according to the algorithms proposed in [31, 36]. The reduced model is generated based on the AMAS specification, together with two additional parameters: the coalition and the set of proposition variables.

Bisimulation checking. The tool allows to check if two models are A-bisimilar for a given coalition A [6]. Apart from the specification of the two models, the bisimulation relation between the corresponding states must also be provided, along with the selected coalition.

6 USAGE

The current version of **STV+Reductions** is available for download here, and allows to:

- Select and display a model specification from a text file,
- Generate and display the explicit state-transition graph,
- Generate and display the reduced model using POR,
- Select specifications of two models and a relation from text files, and check if the models are A-bisimilar wrt the relation,
- Verify the selected full or reduced model by means of fixpoint approximation or dominance-based verification (DominoDFS),
- Alternatively, run the verification for a predefined parameterized model and formula,
- Display the verification result, including the relevant truth values and the winning strategy.

7 CONCLUSIONS

Model checking strategic abilities under imperfect information is notoriously hard. **STV+Reductions** addresses the state explosion problem by an implementation of partial-order reduction and bisimulation checking. This should not only facilitate verification, but also make the techniques easier to use and understand.

Acknowledgements. The authors acknowledge the support of the Luxembourg National Research Fund (FNR) and the National Centre for Research and Development (NCBiR), Poland, under the CORE/PolLux project STV (POLLUX-VII/1/2019).

REFERENCES

[1] R. Alur, L. de Alfaro, R. Grossu, T.A. Henzinger, M. Kang, C.M. Kirsch, R. Majumdar, F.Y.C. Mang, and B.-Y. Wang. 2001. jMocha: A Model-Checking Tool that Exploits Design Structure. In *Proceedings of International Conference on Software Engineering (ICSE)*. IEEE Computer Society Press, 835–836.

[2] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. 1998. MOCHA: Modularity in Model Checking. In *Proceedings of Computer Aided Verification (CAV) (Lecture Notes in Computer Science, Vol. 1427)*. Springer, 521–525.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. 1997. Alternating-Time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society Press, 100–109.

[4] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49 (2002), 672–713. https://doi.org/10.1145/585265.585270

[5] C. Baier and J.-P. Katoen. 2008. *Principles of Model Checking.* MIT Press.

[6] Francesco Belardinelli, Rodica Condurache, Catalin Dima, Wojciech Jamroga, and Michal Knapik. 2021. Bisimulations for verifying strategic abilities with an application to the ThreeBallot voting protocol. *Information and Computation* 276 (2021), 104552. https://doi.org/10.1016/j.ic.2020.104552

[7] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. 2017. Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic. In *Proceedings of IJCAI.* 91–97.

[8] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. 2017. Verification of Multi-agent Systems with Imperfect Information and Public Actions. In *Proceedings of AAMAS.* 1268–1276.

[9] N. Bulling, J. Dix, and W. Jamroga. 2010. Model Checking Logics of Strategic Ability: Complexity. In *Specification and Verification of Multi-Agent Systems*, M. Dastani, K. Hindriks, and J.-J. Meyer (Eds.). Springer, 125–159.

[10] N. Bulling and W. Jamroga. 2011. Alternating Epistemic Mu-Calculus. In *Proceedings of IJCAI-11.* 109–114.

[11] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2014. Improving the Model Checking of Strategies under Partial Observability and Fairness Constraints. In *Formal Methods and Software Engineering*. Lecture Notes in Computer Science, Vol. 8829. Springer, 27–42. https://doi.org/10.1007/978-3-319-11737-9_3

[12] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2015. Reasoning about memoryless strategies under partial observability and unconditional fairness constraints. *Information and Computation* 242 (2015), 128–156. https://doi.org/10.1016/j.ic.2015.03.014

[13] P. Cermak, A. Lomuscio, F. Mogavero, and A. Murano. 2014. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Proc. of Computer Aided Verification (CAV) (Lecture Notes in Computer Science, Vol. 8559)*. Springer, 525–532.

[14] Petr Cermák, Alessio Lomuscio, and Aniello Murano. 2015. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *Proceedings of AAAI.* 2038–2044.

[15] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. 2013. PRISM-games: A Model Checker for Stochastic Multi-Player Games. In *Proceedings of Tools and Algorithms for Construction and Analysis of Systems (TACAS) (Lecture Notes in Computer Science, Vol. 7795)*. Springer, 185–191.

[16] C. Dima, B. Maubert, and S. Pinchinat. 2014. The Expressive Power of Epistemic μ-Calculus. *CoRR* abs/1407.5166 (2014).

[17] C. Dima, B. Maubert, and S. Pinchinat. 2015. Relating Paths in Transition Systems: The Fall of the Modal Mu-Calculus. In *Proceedings of Mathematical Foundations of Computer Science (MFCS) (Lecture Notes in Computer Science, Vol. 9234)*. Springer, 179–191. https://doi.org/10.1007/978-3-662-48057-1_14

[18] C. Dima and F.L. Tiplea. 2011. Model-checking ATL under Imperfect Information and Perfect Recall Semantics is Undecidable. *CoRR* abs/1102.4225 (2011).

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

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

[21] X. Huang and R. van der Meyden. 2014. Symbolic Model Checking Epistemic Strategy Logic. In *Proceedings of AAAI Conference on Artificial Intelligence.* 1426–1432.

[22] W. Jamroga and J. Dix. 2006. Model Checking ATL$_{ir}$ is Indeed Δ_2^P-complete. In *Proceedings of EUMAS (CEUR Workshop Proceedings, Vol. 223)*.

[23] Wojciech Jamroga, Yan Kim, Damian Kurpiewski, and Peter Y. A. Ryan. 2020. Towards Model Checking of Voting Protocols in Uppaal. In *Proceedings of E-Vote-ID (Lecture Notes in Computer Science, Vol. 12455)*. Springer, 129–146. https://doi.org/10.1007/978-3-030-60347-2_9

[24] W. Jamroga, M. Knapik, and D. Kurpiewski. 2017. Fixpoint Approximation of Strategic Abilities under Imperfect Information. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 1241–1249.

[25] W. Jamroga, M. Knapik, and D. Kurpiewski. 2018. 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, Vol. 11143)*. Springer, 100–116.

[26] W. Jamroga, M. Knapik, D. Kurpiewski, and Ł. Mikulski. 2019. Approximate Verification of Strategic Abilities under Imperfect Information. *Artificial Intelligence* 277 (2019).

[27] Wojciech Jamroga, Beata Konikowska, Wojciech Penczek, and Damian Kurpiewski. 2020. Multi-valued Verification of Strategic Ability. *Fundamenta Informaticae* 175, 1-4 (2020), 207–251. https://doi.org/10.3233/FI-2020-1955

[28] Wojciech Jamroga, Damian Kurpiewski, and Vadim Malvone. 2020. Natural Strategic Abilities in Voting Protocols. *CoRR* abs/2007.12424 (2020). arXiv:2007.12424 https://arxiv.org/abs/2007.12424

[29] W. Jamroga, W. Penczek, P. Dembiński, and A. Mazurkiewicz. 2018. Towards Partial Order Reductions for Strategic Ability. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 156–165.

[30] W. Jamroga, W. Penczek, and T. Sidoruk. 2020. Strategic Abilities of Asynchronous Agents: Semantic Paradoxes and How to Tame Them. *CoRR* abs/2003.03867 (2020). arXiv:2003.03867 [cs.LO] https://arxiv.org/abs/2003.03867

[31] W. Jamroga, W. Penczek, T. Sidoruk, P. Dembiński, and A. Mazurkiewicz. 2020. Towards Partial Order Reductions for Strategic Ability. *Journal of Artificial Intelligence Research* 68 (2020), 817–850. https://doi.org/10.1613/jair.1.11936

[32] M. Kacprzak and W. Penczek. 2004. Unbounded Model Checking for Alternating-Time Temporal Logic. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IEEE Computer Society, 646–653. https://doi.org/10.1109/AAMAS.2004.10089

[33] D. Kurpiewski, W. Jamroga, and M. Knapik. 2019. STV: Model Checking for Strategies under Imperfect Information. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2019.* IFAAMAS, 2372–2374.

[34] Damian Kurpiewski, Michał Knapik, and Wojciech Jamroga. 2019. On Domination and Control in Strategic Ability. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2019.* IFAAMAS, 197–205.

[35] A. Lomuscio, W. Penczek, and H. Qu. 2010. Partial Order Reductions for Model Checking Temporal-epistemic Logics over Interleaved Multi-agent Systems. *Fundamenta Informaticae* 101, 1-2 (2010), 71–90. https://doi.org/10.3233/FI-2010-276

[36] Alessio Lomuscio, Wojciech Penczek, and Hongyang Qu. 2010. Partial order reductions for model checking temporal epistemic logics over interleaved multi-agent systems. In *9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010), Toronto, Canada, May 10-14, 2010, Volume 1-3.* IFAAMAS, 659–666.

[37] A. Lomuscio, H. Qu, and F. Raimondi. 2017. MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. *International Journal on Software Tools for Technology Transfer* 19, 1 (2017), 9–30. https://doi.org/10.1007/s10009-015-0378-x

[38] A. Lomuscio and F. Raimondi. 2006. MCMAS : A Model Checker for Multi-Agent Systems. In *Proceedings of Tools and Algorithms for Construction and Analysis of Systems (TACAS) (Lecture Notes in Computer Science, Vol. 4314)*. Springer, 450–454.

[39] Doron A. Peled. 1993. All from One, One for All: on Model Checking Using Representatives. In *Proceedings of CAV (Lecture Notes in Computer Science, Vol. 697)*, Costas Courcoubetis (Ed.). Springer, 409–423. https://doi.org/10.1007/3-540-56922-7_34

[40] J. Pilecki, M.A. Bednarczyk, and W. Jamroga. 2014. Synthesis and Verification of Uniform Strategies for Multi-Agent Systems. In *Proceedings of CLIMA XV (Lecture Notes in Computer Science, Vol. 8624)*. Springer, 166–182.

[41] J. Pilecki, M.A. Bednarczyk, and W. Jamroga. 2017. SMC: Synthesis of Uniform Strategies and Verification of Strategic Ability for Multi-Agent Systems. *Journal of Logic and Computation* 27, 7 (2017), 1871–1895. https://doi.org/10.1093/logcom/exw032

[42] P.Y.A. Ryan. 2010. The Computer Ate My Vote. In *Formal Methods: State of the Art and New Directions.* Springer, 147–184.

[43] P. Y. Schobbens. 2004. Alternating-Time Logic with Imperfect Recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93.

[44] M. Tabatabaei, W. Jamroga, and Peter Y. A. Ryan. 2016. 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.* ACM, 1:1–1:8. https://doi.org/10.1145/2970030.2970039

[45] W. van der Hoek and M. Wooldridge. 2002. Tractable Multiagent Planning for Epistemic Goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*, C. Castelfranchi and W.L. Johnson (Eds.). ACM Press, New York, 1167–1174.