STV+KH: Towards Practical Verification of Strategic Ability for Knowledge and Information Flow

Demonstration Track

Mateusz Kamiński
Institute of Computer Science, Polish Academy of Sciences
Faculty of Mathematics and Computer Science, Nicolaus Copernicus University in Toruń
m.kaminski@ipipan.waw.pl

Damian Kurpiewski
Institute of Computer Science, Polish Academy of Sciences
Faculty of Mathematics and Computer Science, Nicolaus Copernicus University in Toruń
d.kurpiewski@ipipan.waw.pl

Wojciech Jamroga
Interdisciplinary Centre for Security, Reliability and Trust, SnT, University of Luxembourg
Institute of Computer Science, Polish Academy of Sciences
wojciech.jamroga@uni.lu

ABSTRACT

We present an expanded version of our tool **STV** for model checking of strategic abilities. The new version adds support for knowledge and uncertainty operators, thus enabling the verification of properties such as privacy, anonymity, and strategic information flow. All of that is available through a web interface, with no need to install or configure the software by the user.

KEYWORDS

model checking; strategic ability; alternating-time temporal logic

ACM Reference Format:
Mateusz Kamiński, Damian Kurpiewski, and Wojciech Jamroga. 2024. **STV+KH**: Towards Practical Verification of Strategic Ability for Knowledge and Information Flow: Demonstration Track. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

Model checking of multi-agent systems allows for formal verification of their relevant properties. An important group of such properties concerns the ability of agents to achieve (or prevent) a given state of affairs [2, 5, 18, 20]. This is often combined with requirements regarding the agents' knowledge or uncertainty. For example, one can address the ability of a voter v to eventually know whether her vote has been registered correctly for candidate j (*voter-verifiability*), or to maintain the coercer's uncertainty about the value of the vote (*vote privacy*). The former requirement can be specified in the epistemic extension of alternating-time temporal logic **ATLK** [12, 22] by the formula $\langle\langle v\rangle\rangle \mathrm{F}(K_v \mathrm{vote}_{v,j} \vee K_v \neg \mathrm{vote}_{v,j})$. The latter is captured in **ATLH** by $\langle\langle v\rangle\rangle \mathrm{G} H_c^{\geq k}(\mathrm{vote}_{v,1}, \ldots, \mathrm{vote}_{v,n})$, where H is the uncertainty modality proposed recently in [21], n is the number of candidates in the election, and k says how many bits of uncertainty we want on the side of the coercer. Here, we propose a new extension of our experimental tool **STV** [13, 15], that allows to verify such specifications for models of asynchronous MAS.

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

Related work. A number of model checkers for agent logics have been proposed over the last 25 years. Of those, MCK [9] addresses only epistemic properties; Mocha [1], the PRISM family [6], STV [13, 15], and most Strategy Logic extensions of MCMAS [4] admit only strategic-temporal operators. MCMAS [16] and MCMAS-SLK [3] allow for strategic and epistemic modalities, but concentrate on perfect information strategies. Our new proposal, **STV+KH**, combines verification of memoryless imperfect information strategies with specifications of agents' knowledge. Even more importantly, it allows for the analysis of how agents can influence the information flow and the quantitative uncertainty in the system.

Application domain. STV+KH addresses formal verification of MAS, which is a nontrivial problem [7]. Anonymity, privacy, and effective information exchange are essential requirements for many systems. **STV+KH** offers a user-friendly environment for the analysis of such requirements, including a GUI and a flexible model specification language. Moreover, **STV+KH** has a strong pedagogical valor, as it can be used for an intuitive introduction to the complicated subject of strategic reasoning and model checking of strategic logics. The previous versions of **STV** have already been used in tutorials and graduate courses at IJCAI, PRIMA, and ESSAI.

2 FORMAL BACKGROUND

Modules. The main part of the input is given by a set of asynchronous modules [11, 17], where local states are labelled with valuations of state variables. The transitions are valuations of input variables controlled by the other modules. The global model of the MAS is defined by the asynchronous product of its modules.

Strategies. A strategy is a conditional plan that specifies what the agent(s) are going to do in every possible situation [2, 20]. Here, we consider the case of *imperfect information memoryless strategies*, represented by functions from the agent's local states 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.

Logic. 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 agents A that makes φ true on all the outcome paths starting from any state indistinguishable from q [2, 20]. Moreover, $K_a\varphi$ holds in M, q iff φ is true in every state q' indistinguishable from q for a [8]. Finally, $H_a^{\leq r}(\varphi_1, \ldots, \varphi_n)$ holds iff the number of possible valuations for $(\varphi_1, \ldots, \varphi_n)$ in a's indistinguishability class can be represented on r bits. I.e., the Hartley measure of uncertainty for a is at most r [21].

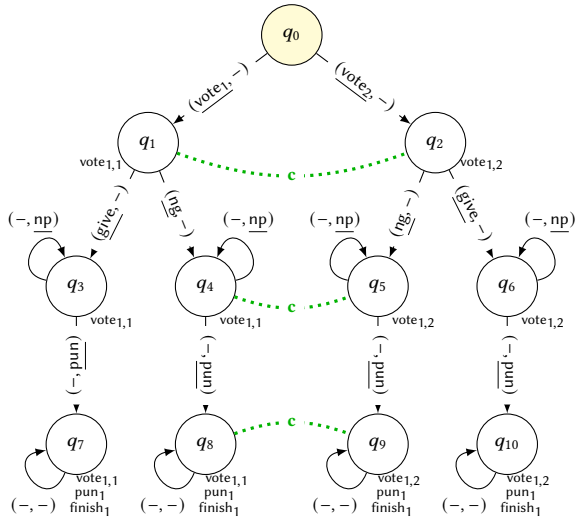

Figure 1: Simple voting model with 1 voter and 2 candidates.

Example scenario. As a working example, we use the Asynchronous Simple Voting scenario [10]. The model consists of k voters and a single coercer. Figure 1 presents the global model with one voter. There are several propositional variables in the model: $\mathsf{vote}_{i,j}$: whether the voter i has voted for the candidate j; pun_i: whether the voter i was punished or not; finish_i: whether the voter i has finished the voting process and her interactions with 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 coercer has two available actions per voter: to punish (or not) the voter.

3 TECHNOLOGY AND USAGE

STV+KH 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 user can load and parse the input specification from a text file that defines the modules, i.e., local automata representing the agents. The generated models and the verification results are visualised in an intuitive web-based graphical interface. The verification algorithms are implemented in C++, and the GUI in Typescript, using the Angular framework.

The tool is available at stv.cs-htiew.com. The video demonstration of the tool is available at youtu.be/yDuK_Vsr8sQ. Example specifications can be found at stv-docs.cs-htiew.com. The current version of **STV+KH** allows to: generate and display the composition of a set of modules into the model of a multi-agent system; provide local specifications for modules, and compute the global specification as their conjunction; verify an **ATLK** and/or **ATLH** reachability or safety formula with knowledge and/or uncertainty operators (nested strategic operators are not allowed); display the verification result including the relevant truth values.

4 EXPERIMENTAL EVALUATION

We evaluate the performance of the new operators on two benchmarks: the Simple Voting example from Section 2, and the much more sophisticated family of models for the voting protocol Selene [14, 19]. All times are given in seconds. The timeout was set to 3*h*. The test platform was a server with ninety-six 2.40 GHz Intel Xeon Platinum 8260 CPUs, 991 GB RAM, and 64-bit Linux.

#V	States	Time	Result
1	15	0.005	False
2	133	0.008	False
3	1071	0.097	False
4	8461	1.559	False
5	66855	52.493	False
6	timeout		

Table 1: Results for Simple Voting: 2 candidates, formula ϕ_1

#V	States	ϕ_1		ϕ_2		ϕ_3	
		Time	Res.	Time	Res.	Time	Res.
1	1267	0.1	False	0.1	True	0.1	False
2	38530	2.5	False	2.7	True	2.6	False
3	2195950	180.7	False	200.4	True	184.3	False
4	timeout						

Table 2: Results for Selene with 3 candidates

Simple Voting. For Simple Voting, we used the **ATLK** formula:

$$\phi_1 \equiv \langle\langle c \rangle\rangle \mathrm{G}((\mathsf{finish} \wedge \mathsf{vote}_{i,j}) \rightarrow K_c \mathsf{vote}_{i,j})$$

Thus, ϕ_1 expresses the undesirable property of *strategic anonymity breach*, saying that the coercer can ensure that the coercer knows the value of i's vote whenever the election comes to an end and voter i has voted for candidate j. We use $i = j = 1$ in all experiments. The experimental results are shown in Table 1. **STV+KH** was able to verify up to 6 agents (5 voters and 1 coercer). The formula was *false* in all instances, i.e., no anonymity breach was found.

Selene. For Selene, we first verified ϕ_1, showing that the coercer cannot gain exact knowledge about the voter's vote also in that case (see the results in Table 2). Then, we verified the **ATLH** formulas

$$\phi_2 \equiv \langle\langle c \rangle\rangle \mathrm{G}(\mathsf{finish} \rightarrow H_c^{\leq 2}(\mathsf{vote}_{i,1}, \mathsf{vote}_{i,2}, \ldots, \mathsf{vote}_{i,n})),$$

which turned to be true in all instances, and

$$\phi_3 \equiv \langle\langle c \rangle\rangle \mathrm{G}(\mathsf{finish} \rightarrow H_c^{\leq 1}(\mathsf{vote}_{i,1}, \mathsf{vote}_{i,2}, \ldots, \mathsf{vote}_{i,n})),$$

which was always false. Thus, the coercer can reduce his uncertainty about the voter's vote to at most 2 bits, but not further down to 1 bit. Throughout the experiments, we used $i = 1$ and $n = 3$. We were able to verify up to 4 agents (3 voters and 1 coercer).

5 CONCLUSIONS

We present **STV+KH**: a substantial extension of the **STV** model checker, augmented with modalities for agents' knowledge and quantitative uncertainty. The experiments show that the verification of anonymity-related properties using **STV+KH** performs similarly to model checking "vanilla" strategic properties, reported in [13–15]. Thus, we gain significant expressivity with little price in terms of complexity and performance.

ACKNOWLEDGMENTS

The work has been supported by NCBR Poland and FNR Luxembourg under the PolLux/FNR-CORE project SpaceVote (POLLUX-XI/14/SpaceVote/2023 and C22/IS/17232062/SpaceVote), by NCN Poland under the CHIST-ERA grant SAI (CHIST-ERA-19-XAI-010, 2020/02/Y/ST6/00064), by FNR Luxembourg under the CORE project PABLO (C21/IS/16326754/PABLO), and by the CNRS IEA project MoSART. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the authors have applied CC BY 4.0 license to any Author Accepted Manuscript version arising from this submission.

REFERENCES

[1] 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.

[2] 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

[3] 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.

[4] 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.

[5] K. Chatterjee, T.A. Henzinger, and N. Piterman. 2010. Strategy Logic. *Information and Computation* 208, 6 (2010), 677–693.

[6] 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.

[7] M. Dastani, K. Hindriks, and J.-J. Meyer (Eds.). 2010. *Specification and Verification of Multi-Agent Systems*. Springer.

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

[9] P. Gammie and R. van der Meyden. 2004. MCK: Model Checking the Logic of Knowledge. In *Proc. of the 16th Int. Conf. on Computer Aided Verification (CAV'04) (LNCS, Vol. 3114)*. Springer-Verlag, 479–483.

[10] Wojciech Jamroga, Michał Knapik, Damian Kurpiewski, and Łukasz Mikulski. 2019. Approximate Verification of Strategic Abilities under Imperfect Information. *Artificial Intelligence* 277 (2019). https://doi.org/10.1016/j.artint.2019.103172

[11] 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

[12] W. Jamroga and W. van der Hoek. 2004. Agents that Know how to Play. *Fundamenta Informaticae* 63, 2–3 (2004), 185–219.

[13] Damian Kurpiewski, Wojciech Jamroga, and Michał 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.

[14] Damian Kurpiewski, Wojciech Jamroga, Lukasz Masko, Lukasz Mikulski, Witold Pazderski, Wojciech Penczek, and Teofil Sidoruk. 2022. Verification of Multi-Agent Properties in Electronic Voting: A Case Study. In *Advances in Modal Logic*. College Publications, 531–556.

[15] Damian Kurpiewski, Witold Pazderski, Wojciech Jamroga, and Yan Kim. 2021. STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions. In *Proceedings of AAMAS*. ACM, 1770–1772.

[16] 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

[17] Alessio Lomuscio, Ben Strulo, Nigel G. Walker, and Peng Wu. 2013. Assume-Guarantee Reasoning with Local Specifications. *Int. J. Found. Comput. Sci.* 24, 4 (2013), 419–444. https://doi.org/10.1142/S0129054113500123

[18] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic* 15, 4 (2014), 1–42.

[19] P.Y.A. Ryan, P.B. Rønne, and V. Iovino. 2016. Selene: Voting with Transparent Verifiability and Coercion-Mitigation. In *Financial Cryptography and Data Security: Proceedings of FC 2016. Revised Selected Papers (Lecture Notes in Computer Science, Vol. 9604)*. Springer, 176–192. https://doi.org/10.1007/978-3-662-53357-4_12

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

[21] Masoud Tabatabaei and Wojciech Jamroga. 2023. Playing to Learn, or to Keep Secret: Alternating-Time Logic Meets Information Theory. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, AAMAS*. ACM, 766–774. https://doi.org/10.5555/3545946.3598710

[22] W. van der Hoek and M. Wooldridge. 2003. Cooperation, Knowledge and Time: Alternating-time Temporal Epistemic Logic and its Applications. *Studia Logica* 75, 1 (2003), 125–157.