PSTV: Towards Practical Verification of Strategic Ability for Probabilistic Models with Imperfect Information

Demonstration Track

Mateusz Kamiński
Faculty of Mathematics and Computer Science & Institute of Advanced Studies, Nicolaus Copernicus University in Toruń
mathis@mat.umk.pl

Wojciech Jamroga
Institute of Computer Science, Polish Academy of Sciences
Faculty of Mathematics and Computer Science & Institute of Advanced Studies, Nicolaus Copernicus University in Toruń
w.jamroga@ipipan.waw.pl

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

ABSTRACT

We present **PSTV**, a new, probabilistic branch of our tool suite **STV** (STrategic Verifier). **PSTV** supports formal verification of strategic ability in multi-systems with imperfect information and probabilistic behavior. 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; probabilistic systems

ACM Reference Format:
Mateusz Kamiński, Wojciech Jamroga, and Damian Kurpiewski. 2026. **PSTV**: Towards Practical Verification of Strategic Ability for Probabilistic Models with Imperfect Information: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/GQVY6801

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, 7, 16, 23, 24], especially in the realistic case of systems with partial observability [5, 19, 25], and stochastic environments of action [4, 9, 15]. For example, one can address the ability of a voter v to ensure with at least 95% probability that her vote is eventually registered correctly for candidate j (*probabilistic voter-verifiability*), or to guarantee that a coercer c does not learn the vote value with probability exceeding 0.5 (*probabilistic vote privacy*). These requirements can be specified in *probabilistic alternating-time temporal logic* **PATL**. The former can be expressed by the formula $\langle\langle v\rangle\rangle^{\geq 0.95}(\mathrm{F}\,\mathrm{vote}_{\mathrm{v,j}})$. The latter is captured by $\langle\langle v\rangle\rangle^{\leq 0.5}(\mathrm{F}\,\mathrm{learns}_{\mathrm{c}})$. Here, we propose a new extension of our experimental tool **STV** [20, 21], enabling the verification of such **PATL** specifications for asynchronous multi-agent systems with imperfect information and imperfect recall.

Related work. Model checkers for agent logics has been an active research area since late 1990s [1, 6, 12, 20, 22]. Among those, tools for stochastic MAS are much scarcer, and include Storm [14], PRISM [13] (both for probabilistic temporal properties), and PRISM Games [8] (for probabilistic strategic properties in perfect information games). To our best knowledge, no existing tool supports model checking for **PATL** with imperfect information. Our new proposal, **PSTV**, fills the gap. As the first step, we focus on the least complex case of *memoryless deterministic strategies with imperfect information* (which is $\Delta_2^{\mathbf{P}}$-complete [3], as opposed to probabilistic and perfect recall strategies, which make the verification respectively **PSPACE**-hard [4] and undecidable [15]).

Application domain. PSTV provides a user-friendly environment for formal verification of MAS [11]. It enables the analysis of critical requirements like functionality, anonymity, and privacy through a graphical interface and a flexible modeling language. The tool also serves a strong pedagogical purpose, offering an intuitive introduction to probabilistic strategic reasoning and model checking.

2 FORMAL BACKGROUND

Modules. The main part of the input is given by a set of asynchronous modules [18, 21], where local states are labelled with valuations of state variables. Transitions are transformations of local states, possibly shared with 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, 25]. 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 possible probability spaces over infinite paths starting from q and consistent with the strategy.

Logic. Given a model M and state q, the **ATL** formula $\langle\langle A\rangle\rangle\varphi$ holds in M, q iff there is a strategy s_A for A that makes φ true on all the paths starting from any state indistinguishable from q and consistent with s_A [2, 25]. Moreover, the **PATL** formula $\langle\langle A\rangle\rangle^{\geq\alpha}\varphi$ holds in M, q iff there is a strategy for A such that, for every indistinguishable state and probability space in the resulting outcome, the probability of

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/GQVY6801

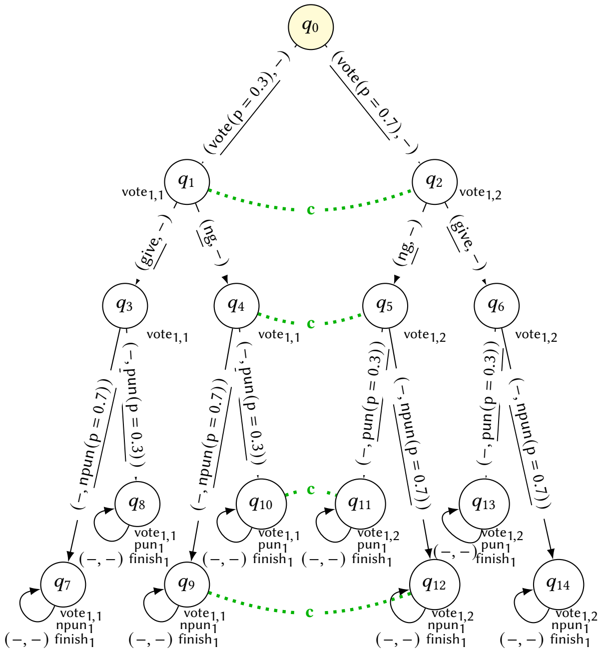

Figure 1: Simple voting model with 1 voter and 2 candidates.

φ is at least α, and analogously for $\langle\langle A\rangle\rangle^{\leq\alpha}$ [4]. We also consider the uncertainty operators $H_a^{\leq\alpha}(\varphi_, \ldots, \varphi_k)$ stating that a's Shannon entropy about the values of $\varphi_, \ldots, \varphi_k$ is at most α [26].

Example scenario. As a working example, we use a modified version of Asynchronous Simple Voting [17]. The ASV model consisting of 1 voter and a single coercer is shown in Figure 1. 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. The voter first casts her vote. With probability 0.3 she casts her vote for the first candidate, and with probability 0.7 for the second candidate. Then she 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. This decision is also probabilistic: with probability 0.3 the coercer punishes the voter, and with probability 0.7 he does not punish her. The coercer has two available actions per voter: to punish (or not) the voter.

3 TECHNOLOGY AND USAGE

PSTV 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 tool is available at stv.cs-htiew.com. Example specifications can be found at stv-docs.cs-htiew.com. The video demonstration of the tool is available at jmp.sh/share/9s8RrRQuxaXKViesRlQr. **PSTV** allows to: generate and display the product of a set of modules; verify an **ATL** or **PATL** formula with non-nested strategic operators, possibly augmented with knowledge and/or uncertainty; display the verification result including the relevant truth values.

#V	States	model gen. time	ϕ_1 (ATL) v. time	ϕ_1 (ATL) v. result	ϕ_{1p} (PATL) v. time	ϕ_{1p} (PATL) v. result
1	15	<0.01	<0.01	FALSE	0.01	TRUE
2	133	<0.01	0.02	FALSE	0.05	TRUE
3	1071	0.05	0.32	FALSE	1.20	TRUE
4	8461	0.82	4.90	FALSE	84.12	TRUE
5	66855	3.12	81.52	FALSE	527.63	TRUE
6	-	timeout	-	-	-	-

Table 1: Results for Simple Voting with 2 candidates

#V	States	model gen. time	ϕ_2 (ATLH) v. time	ϕ_2 (ATLH) v. result	ϕ_{2p} (PATLH) v. time	ϕ_{2p} (PATLH) v. result
1	887	0.05	0.04	FALSE	0.16	TRUE
2	39028	1.58	1.05	FALSE	192.35	TRUE
3	1717232	102.75	80.49	FALSE	timeout	-

Table 2: Results for vVote

4 EXPERIMENTAL EVALUATION

We evaluate the performance of the new operators on two benchmarks: ASV (see Section 2), and the much more sophisticated family of models for the voting protocol vVote [10] with added probabilistic components for preference-based voting model, optional verification compliance and WBB retrieval errors. All times are given in seconds with timeout set to $3h$. The test platform was a server with ninety-six 2.40 GHz Intel Xeon Platinum 8260 CPUs, 991 GB RAM, and 64-bit Linux.

ASV. We used the following formulas of **ATL** (ϕ_1) and **PATL** (ϕ_{1p}):

$$\phi_1 \equiv \langle\langle v1\rangle\rangle \mathrm{G}(\mathsf{finish}_1 \rightarrow (\mathsf{vote}_{1,2} \wedge \mathsf{npun}_1))$$

$$\phi_{1p} \equiv \langle\langle v1\rangle\rangle^{\geq 0.4} \mathrm{G}(\mathsf{finish}_1 \rightarrow (\mathsf{vote}_{1,2} \wedge \mathsf{npun}_1))$$

Formula ϕ_1 says that voter v1 can ensure that if she finishes the vote, then she has voted for candidate 2 and was not punished. Formula ϕ_{1p} expresses that voter v1 can ensure the same with at least 40% probability. The experimental results are shown in Table 1.

vVote. For vVote, we used the following formulas of **ATL** (ϕ_2) and **PATL** (ϕ_{2p}) augmented with Shannon uncertainty operator H:

$$\phi_2 \equiv \langle\langle v1\rangle\rangle \mathrm{G}(\mathsf{end}_{\mathsf{v1}} \wedge \mathsf{voted1}_{\mathsf{v1}} \implies H_{v1}^{\leq 1}(\mathsf{wbb\_voted1}_{\mathsf{v1}}, \mathsf{wbb\_voted2}_{\mathsf{v1}}, \mathsf{wbb\_voted3}_{\mathsf{v1}}))$$

$$\phi_{2p} \equiv \langle\langle v1\rangle\rangle^{\geq 0.9} \mathrm{G}(\mathsf{end}_{\mathsf{v1}} \wedge \mathsf{voted1}_{\mathsf{v1}} \implies H_{v1}^{\leq 1}(\mathsf{wbb\_voted1}_{\mathsf{v1}}, \mathsf{wbb\_voted2}_{\mathsf{v1}}, \mathsf{wbb\_voted3}_{\mathsf{v1}}))$$

Formula ϕ_2 says that voter v1 can ensure that, if she completes the voting and votes for candidate 1, then her uncertainty about how her vote has been recorded (as per the announcement on the Web Bulletin Board) is never more than 1 bit. Formula ϕ_{2p} captures that voter v1 can ensure the same with at least 90% probability. The experimental results are shown in Table 2.

Notice that probabilistic requirements take more time to verify, but they are more likely to be satisfied than "all-or-nothing" properties that can only be expressed in **ATL**.

5 CONCLUSIONS

We present **PSTV**, a substantial extension of the **STV** model checker for the verification of **PATL** with imperfect information. While verifying probabilistic strategic properties incurs higher computational cost than non-probabilistic strategic model checking, **PSTV** makes it feasible, which is a significant step forward for practical verification of multi-agent scenarios.

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). For the purpose of open access, and in fulfilment of 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] Francesco Belardinelli, Wojciech Jamroga, Munyque Mittelmann, and Aniello Murano. 2023. Strategic Abilities of Forgetful Agents in Stochastic Environments. In *Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR*. 726–731. https://doi.org/10.24963/KR.2023/71

[4] Francesco Belardinelli, Wojtek Jamroga, Munyque Mittelmann, and Aniello Murano. 2024. Verification of Stochastic Multi-Agent Systems with Forgetful Strategies. In *Proceedings of AAMAS*. IFAAMAS/ACM, 160–169. https://doi.org/10.5555/3635637.3662863

[5] R. Berthon, B. Maubert, A. Murano, S. Rubin, and M. Y. Vardi. 2017. Strategy logic with imperfect information. In *Proceedings of LICS*. 1–12. https://doi.org/10.1109/LICS.2017.8005136

[6] 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.

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

[8] 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.

[9] Taolue Chen and Jian Lu. 2007. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *Proceedings of FSKD, Volume 2*. IEEE Computer Society, 35–39. https://doi.org/10.1109/FSKD.2007.458

[10] C. Culnane, P.Y.A. Ryan, S.A. Schneider, and V. Teague. 2015. vVote: A Verifiable Voting System. *ACM Trans. Inf. Syst. Secur.* 18, 1 (2015), 3:1–3:30. https://doi.org/10.1145/2746338

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

[12] Angelo Ferrando and Vadim Malvone. 2024. Hands-on VITAMIN: A Compositional Tool for Model Checking of Multi-Agent Systems. In *Proceedings of the 25th Workshop "From Objects to Agents" (CEUR Workshop Proceedings, Vol. 3735)*. CEUR-WS.org, 148–160.

[13] Vojtech Forejt, Marta Z. Kwiatkowska, Gethin Norman, and David Parker. 2011. Automated Verification Techniques for Probabilistic Systems. In *Formal Methods for Eternal Networked Software Systems - Proceedings of SFM (Lecture Notes in Computer Science, Vol. 6659)*. Springer, 53–113. https://doi.org/10.1007/978-3-642-21455-4_3

[14] Christian Hensel, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. 2022. The probabilistic model checker Storm. *Int. J. Softw. Tools Technol. Transfer* 24 (2022), 589–610. https://doi.org/10.1007/s10009-021-00633-z

[15] X. Huang, K. Su, and C. Zhang. 2012. Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall. In *Proceedings of AAAI-12*.

[16] Wojciech Jamroga. 2015. *Logical Methods for Specification and Verification of Multi-Agent Systems*. ICS PAS Publishing House.

[17] 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

[18] 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

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

[20] 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.

[21] 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.

[22] 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

[23] Artur Meski, Maciej Koutny, Łukasz Mikulski, Ion Petre, Wojciech Penczek, and Marcin Piatkowski. 2025. Model checking for distributed reaction systems with temporal-epistemic properties. *Natural Computing* 24, 4 (2025), 1101–1117.

[24] 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.

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

[26] 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