

Parameterised Verification of Strategic Properties in Probabilistic Multi-Agent Systems

Alessio Lomuscio and Edoardo Pirovano

Verification of Autonomous Systems Group
Imperial College London, UK
<https://vas.doc.ic.ac.uk/>

AAMAS 2020

Introduction

- Methods exist to verify multi-agent systems (MAS) with a possibly unbounded number of agents [KL16]; this is known as parameterised verification.
- Recently this work has been extended to incorporate probabilities [LP19].
- However, in some situations (for example, when considering security scenarios) we wish to reason about the abilities of different agents to achieve goals.

Aim: To develop a methodology to verify strategic properties in probabilistic multi-agent systems.

Our Contribution

- 1 We present a novel semantics to reason about strategic properties in **probabilistic multi-agent systems**. Its parameterised verification problem is undecidable in general.
- 2 We give a partial decision procedure for the problem, and prove its correctness. We also identify a variant of the specification language on which our procedure is complete.
- 3 We present an open-source implementation of this procedure and evaluate it against an example scenario.

Probabilistic Multi-Agent Systems

- Our probabilistic multi-agent systems (PMAS) semantics is based on interpreted systems [Fag+95] and their extension to handle unbounded systems [KL16].
- In our system we define the behaviour of an agent (of which there may be arbitrarily many copies) and an environment.
- We assume that all agents are behaviourally identical. Our results would also hold for a finite number of different agent behaviours.
- Further, we assume there is a null action that is always available to every agent and cannot be observed by other agents or the environment.

Definition

A *probabilistic agent template* is a tuple $T = \langle S, \iota, Act, P, t \rangle$ where:

- The finite set $S \neq \emptyset$ represents the agent's local states.
- $\iota \in S$ is a distinguished initial state.
- $Act \neq \emptyset$ is a finite set of possible local actions.
- The agent's protocol function $P : S \rightarrow 2^{Act}$ gives the set of possible actions in each state.
- The agent's transition function $t : S \times Act_E \times 2^{Act} \times Act \rightarrow Dist(S)$ returns a distribution on the agent's next state given its current state, the environment's action, the set of actions performed by all the agents (including the one performed by the agent being considered) and the action performed by this agent at this time-step.
- The environment is similarly defined.

Example Agent

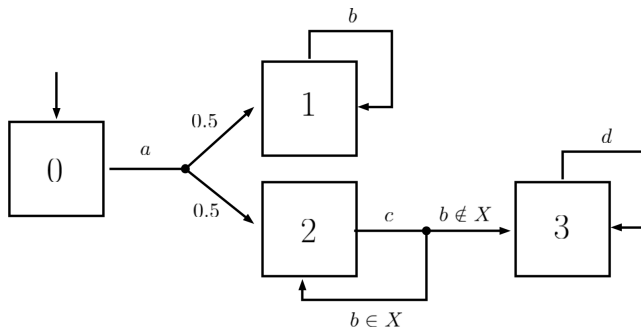


Figure: An example of an agent template.

Probabilistic Multi-Agent Systems

- We put an agent template and an environment together with a labelling function to give a system, defined below.

Definition

A *probabilistic multi-agent system* (PMAS) is a tuple $\mathcal{S} = \langle T, E, \mathcal{V} \rangle$, where T is a probabilistic agent template, E is an environment and $\mathcal{V} : S \times S_E \rightarrow 2^{AP}$ is a labelling function on a set of atomic propositions AP .

- We denote by $\mathcal{S}(n)$ the system obtained by fixing a number n of agents.

- We consider specifications based on a fragment of PATL* which we call P[ATL*].

Definition

Given a set AP of atomic propositions, P[ATL*] formulae are defined by the following grammar:

$$\begin{aligned}\phi &::= \langle\langle A \rangle\rangle P_{\bowtie r}[\psi] \\ \psi &::= \top \mid (p, i) \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid \psi U \psi,\end{aligned}$$

where $A \subset \mathbb{Z}^+ \cup \{E\}$ is a finite set of agents (and possibly the environment), $p \in AP$, $i \in \mathbb{Z}^+$, $\bowtie \in \{<, \leq, \geq, >\}$ and $r \in [0, 1]$.

Example

Consider an opinion formation protocol where a group of robots have to agree on some choice of option. Then, the P[ATL*] formula

$$\langle\langle 2, E \rangle\rangle P_{\geq 0.5}[G \neg(\textit{decisionReached}, 1)]$$

represents that agent 2 and the environment have a strategy that ensures with probability at least 0.5 that agent 1 does not reach a decision.

- We say a formula is m -indexed if it refers to agents with index at most m . For example, the above formula is 2-indexed.

Parameterised Model Checking Problem

Definition

Given a PMAS \mathcal{S} and an m -indexed formula ϕ , the PMCP is to determine whether $\mathcal{S}(n) \models \phi$ for all $n \geq m$. If this is the case we write $\mathcal{S} \models \phi$.

- This problem is undecidable in general. Nonetheless, we aim to develop a partial decision procedure for it.

Definition

Let \mathcal{S} be a PMAS, A a coalition of agents and ψ a path formula. Then we use $\langle\langle A \rangle\rangle P_{n, \max=?}[\psi]$ to denote the maximal value of $r \in [0, 1]$ for which it is the case that $\mathcal{S}(n) \models \langle\langle A \rangle\rangle P_{\geq r}[\psi]$.

- Intuitively, this is the maximum probability of ψ that the agents A can achieve in a system of size n .
- Note that if we can compute the range of $\langle\langle A \rangle\rangle P_{n, \max=?}[\psi]$ as n varies then we would have a decision procedure for the PMCP.

Upper Bound

Lemma

Let S be a PMAS. Then, for any set of agents A and path formula ψ

$$\langle\langle A \rangle\rangle P_{n, \max=?}[\psi] \geq \langle\langle A \rangle\rangle P_{n+1, \max=?}[\psi]$$

for values of n larger than the index of the formula.

- Intuitively, adding a new agent that is not referred to in the formula can only make it less likely that ψ is satisfied since the additional agent will be aiming to achieve $\neg\psi$.
- So, the upper bound on the probability for an m -indexed formula is simply $\langle\langle A \rangle\rangle P_{m, \max=?}[\psi]$, which we can compute.

Lower Bound

- For the lower bound, we give a method based on constructing an abstract model, the details of which can be found in the paper.
- In this model, we have a component that can capture the behaviour of an arbitrarily large number of agents.
- The bound computed by this method is not tight. Thus, the decision procedure is incomplete. However, it still enables the verification of some systems.
- We can replace the the $\psi_1 U \psi_2$ operator in our logic with $\psi_1 U^{\leq k} \psi_2$ which is read as “at some point within k time-steps ψ_2 holds and before then ψ_1 is true”. In this case, we can give a complete procedure.

Implementation

- Our implementation, called PSV-S, is based on PRISM-games [Che+13]. It is open-source and available to download here: <https://vas.doc.ic.ac.uk/software/probabilistic/>
- In order to verify the functionality and scalability of our tool, we used it to model a channel jamming security protocol [Zhu+10].
- In this scenario, users can either jam channels (making it more difficult for other users of that channel to transmit along them) or transmit messages.
- We studied the property $\langle\langle 1 \rangle\rangle P_{\geq p}[F^{\leq 15}(\text{transmitted}_3, 1)]$ which represents that agent 1 can ensure with probability at least p that 3 messages are transmitted within the first 15 time steps.

Results

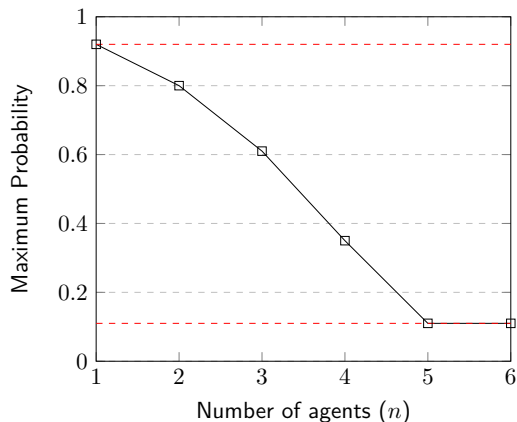


Figure: Graph showing the probability $\langle\langle 1 \rangle\rangle P_{n, \max=?}[F^{\leq 15}(\text{transmitted}_3, 1)]$ for different values of n . The red dashed lines show the expected bounds computed by our procedure.

Conclusions

- We have proposed a semantics to reason about strategies in probabilistic multi-agent systems with a possibly unbounded number of agents.
- While our verification problem is undecidable in general, we have presented a partial decision procedure which is complete for a less expressive variant of the specification logic.
- We plan to continue work in this area by identifying further decidable fragments of the verification problem and building decision procedures for these.

References



T. Chen et al. “Automatic Verification of Competitive Stochastic Systems”. In: *Formal Methods in System Design* 43.1 (2013), pp. 61–92.



R. Fagin et al. *Reasoning about Knowledge*. Cambridge: MIT Press, 1995.



P. Kouvaros and A. Lomuscio. “Parameterised Verification for Multi-Agent Systems”. In: *Artificial Intelligence* 234 (2016), pp. 152–189.



A. Lomuscio and E. Pirovano. “A Counter Abstraction Technique for the Verification of Probabilistic Swarm Systems”. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS19)*. IFAAMAS Press, 2019, pp. 161–169.



Q. Zhu et al. “A Stochastic Game Model for Jamming in Multi-Channel Cognitive Radio Systems”. In: *Proceedings of IEEE International Conference on Communications, (ICC10)*. IEEE, 2010, pp. 1–6.