Parameterised Verification for Probabilistic Multi-agent Systems

Edoardo Pirovano
Presenting joint work with Alessio Lomuscio

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

Computer Science needs Logic!
15 January 2021

Drone Swarms

Introduction

- In some cases, such as when designing drone swarms, the number of participants in a system may not be known until run-time.
- Methods exist to verify multi-agent systems with a possibly unbounded number of agents [KL16]; this is known as parameterised verification.
- Separately, probabilistic model checking techniques have been developed to verify systems with stochastic transitions [Kat16].

Aim: To develop a methodology to verify properties in unbounded multi-agent systems that are also probabilistic.

Our Contribution

1. We define a number of different semantics to reason about **unbounded probabilistic multi-agent systems**.
2. We give procedures to solve the parameterised model checking problem for these systems.
3. We present open-source implementations of these procedures and case studies analysed using them.

Unbounded Probabilistic Multi-Agent Systems

- One of our semantics extends Parameterised Interleaved Interpreted Systems [KL15] to include probabilities.
- We assume that all agents are behaviourally identical.
- The agents synchronize with each-other in different ways depending on the type of action being performed:
  - *Asynchronous:* Performed by one agent on its own.
  - *Agent-environment:* Performed by *one* agent together with the environment.
  - *Global synchronous:* Performed by *all* agents together with the environment.

Agents and Environments

Definition (Probabilistic Agent Template)

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

- The set S represents a finite set of agent local states.
- $\iota \in S$ is a distinguished initial state.
- The non-empty set $Act = A \cup AE \cup GS$ defines the actions that can be performed by the agents.
- The agent's protocol function $P : S \to \mathcal{P}(Act) \setminus \{\emptyset\}$ defines the actions available in each state.
- The agent's transition function $t : S \times Act \times S \to [0, 1]$ is such that for every $s \in S$ and $a \in P(s)$ we have $\sum_{s' \in S} t(s, a, s') = 1$. This defines the probabilistic next state given the current state and action.

- The environment is similarly defined.

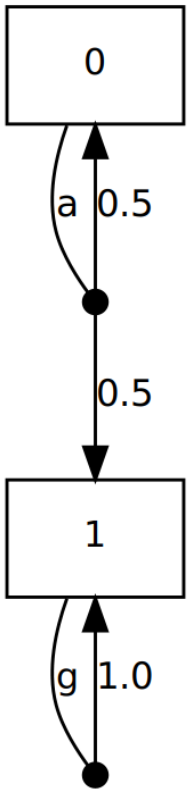
Our Semantics

Definition (PPIIS)

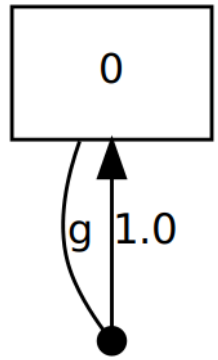
A *probabilistic parameterised interleaved interpreted system* (PPIIS) is a tuple $\mathcal{S} = \langle T, E, L \rangle$, where T is a probabilistic agent template, E is an environment and $L : S \times S_E \to \mathcal{P}(AP)$ is a labelling function for a set of atomic propositions AP.

- Each such parameterised system defines an infinite family of MDPs made by setting a different number of agents.
- We denote by $\mathcal{S}(n)$ the concrete system with n agents; notice each of these is a finite MDP.

Example PPIIS



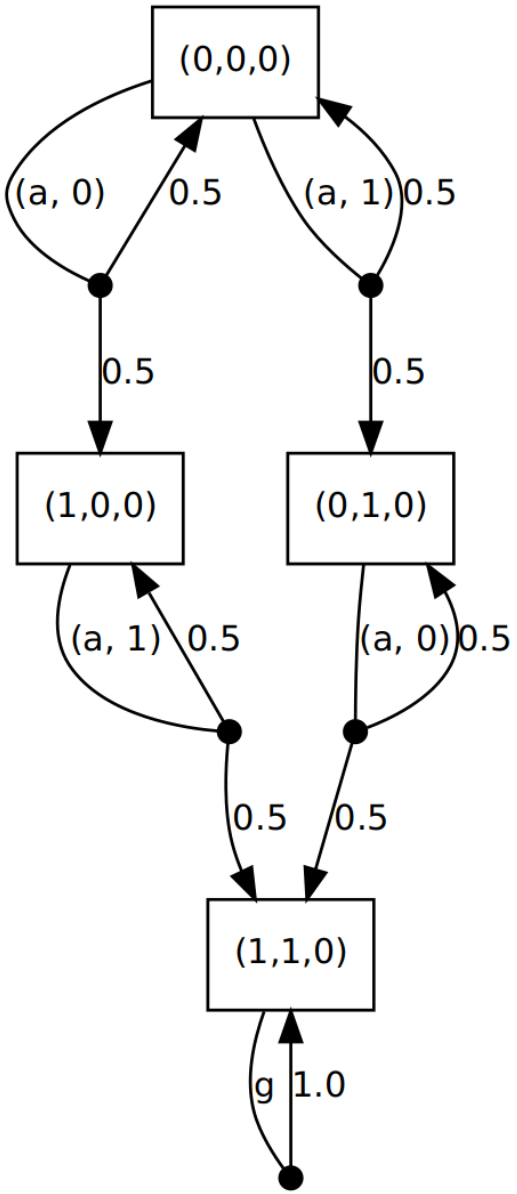
(a) An agent template.



(b) An environment.

Figure: An example PPIIS. The a action is asynchronous, while the g one is global synchronous.

Example Concrete System



Parameterised Model Checking

Definition (Parameterised Model Checking)

Given a system $\mathcal{S}$ and a formula ϕ, the parameterised model checking problem involves checking whether it is the case that $\mathcal{S}(n) \models \phi$ for all n.

- Note that this problem is a generalisation to probabilistic systems of a problem that is known to be undecidable in general [AK86], so it is certainly also undecidable in general.
- Nonetheless, we can identify partial decision procedures.

Abstract Models

- To verify formulas, we construct *abstract models*.
- Our abstract models only record which states have **one or more** agents in them instead of recording the exact number of agents in each state. This allows us to build a finite model for arbitrarily large systems.
- Every path in a concrete system of any size has a corresponding path in the abstract model.
- So when a property is verified in the abstract model, we know it will hold in systems of any size.

Implementations

- Our implementations, based on PRISM [KNP11], are available here: https://vas.doc.ic.ac.uk/software/probabilistic/
- One case study we modelled is a *foraging* protocol in which robots aim to find food and bring it back to a nest.
- We assume robots carrying food can drop it with probability p_f.
- We checked the property $P^{\max}_{\leq p}[F^{<k}\text{deposited}_2]$ which says that the probability 2 units of food are deposited within k steps is at most p.

Results

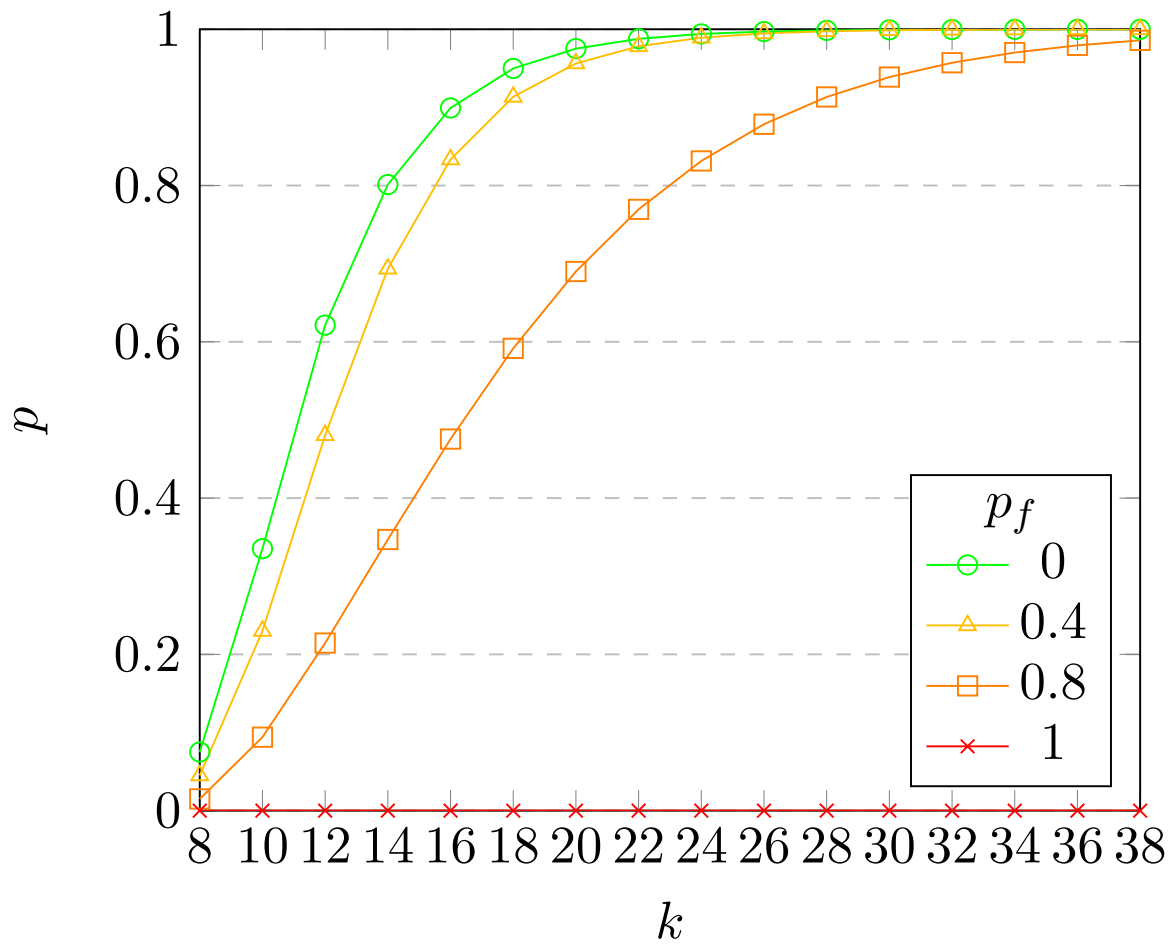


Figure: For different fault probabilities p_f and time-steps k, the maximum value of p for which $P^{\max}_{\leq p}[F^{<k}\text{deposited}_2]$ holds in arbitrarily large systems.

Conclusions

- In this line of work, we proposed procedures for checking multi-agent systems that are **unbounded** and **probabilistic**.
- We developed our verification procedures into a practical toolkit and used it to verify example scenarios from swarm robotics and other application domains.
- One area for future work is improving the scalability of our tool in order to apply it to further examples.

Summary of Publications

- [LP18]: Introduced a synchronous semantics, where agents all take actions simultaneously in turns.
- [LP19]: Introduced an asynchronous semantics with different action types, as presented here.
- [LP20a]: Extends our synchronous semantics to support the checking of strategic properties in which we reason about what properties different groups of agents can enforce.
- [LP20b]: Extends our asynchronous semantics to allow us to model systems where some of the agents may exhibit faults.

References I

K. Apt and D. C. Kozen. "Limits for automatic verification of finite-state concurrent systems". In: *Information Processing Letters* 22.6 (1986), pp. 307–309.

J. Katoen. "The Probabilistic Model Checking Landscape". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS16)*. ACM, 2016, pp. 31–45.

P. Kouvaros and A. Lomuscio. "A Counter Abstraction Technique for the Verification of Robot Swarms". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI15)*. AAAI Press, 2015, pp. 2081–2088.

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

M. Kwiatkowska, G. Norman, and D. Parker. "PRISM 4.0: Verification of Probabilistic Real-time Systems". In: *Proceedings of the 23rd International Conference on Computer Aided Verification (CAV11)*. Vol. 6806. Lecture Notes in Computer Science. Springer, 2011, pp. 585–591.

References II

A. Lomuscio and E. Pirovano. "Verifying Emergence of Bounded Time Properties in Probabilistic Swarm Systems". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence and 23rd European Conference on Artificial Intelligence (IJCAI-ECAI18)*. AAAI Press, 2018, pp. 403–409.

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.

A. Lomuscio and E. Pirovano. "Parameterised Verification of Strategic Properties in Probabilistic Multi-Agent Systems". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS20)*. IFAAMAS Press, 2020, pp. 762–770.

A. Lomuscio and E. Pirovano. "Verifying Fault-Tolerance in Probabilistic Swarm Systems". In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI-PRICAI20)*. ijcai.org, 2020, pp. 325–331.