A Counter Abstraction Technique for the Verification of Probabilistic Swarm Systems

Alessio Lomuscio, Edoardo Pirovano

Introduction

- Swarms of drones ("agents") follow certain protocols in order to co-operate and achieve an overall goal.
- These protocols are sometimes probabilistic.
- Existing research in probabilistic model checking allows us to check finite systems with a fixed number of agents.
- Separately, parameterised model checking techniques allow us to check systems with a possibly unbounded number of agents.

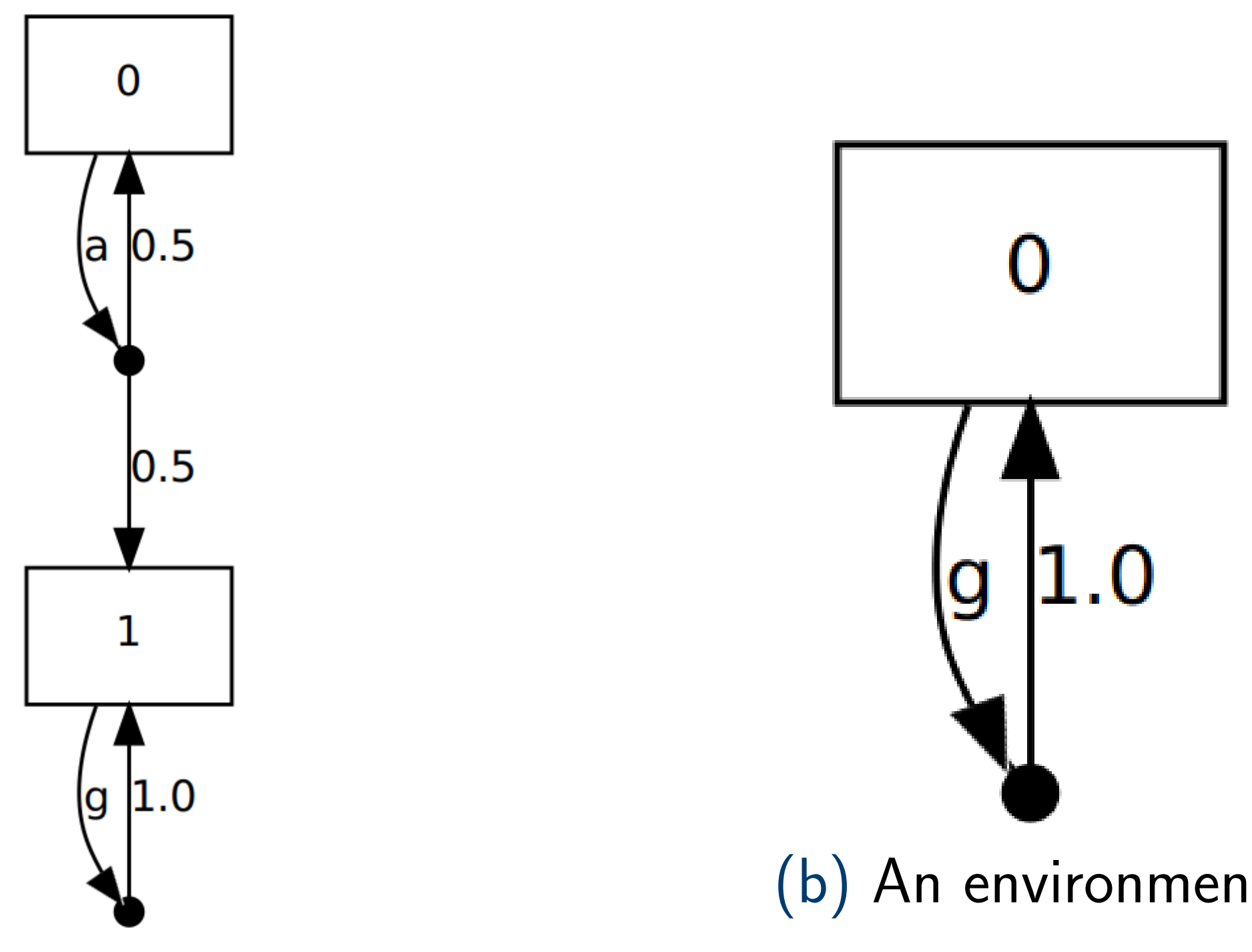
But what about probabilistic systems with a possibly unbounded number of agents?

A swarm of kilobots

Models

- We model our swarms by an agent template, which captures the possible states and behaviours of one of the agents, and an environment which captures the rest of the state of the system.
- A *probabilistic agent template* is a tuple $T = \langle L, \iota, Act, P, t \rangle$ where:
  - The set S represents a finite set of agent local states.
  - $\iota \in S$ is a distinguished initial state.
  - The non-empty set $Act = A \cup \mathbf{AE} \cup \mathbf{GS}$ gives the actions that can be performed by the agents.
  - The agent's protocol function $P : S \to \mathcal{P}(Act) \setminus \{\emptyset\}$ gives 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 gives the probabilistic next state given the current state and action.
- We similarly define an environment.

Example System



(a) An agent template.

(b) An environment.

PLTL

- For $a \in AP$ and $i \in \mathbb{N}$, the probabilistic LTL logic is the set of formulas ϕ defined by the following BNF:

  $\phi ::= P^{\max}_{\bowtie x}[\psi] \mid P^{\min}_{\bowtie x}[\psi]$ for $x \in [0,1]$ and $\bowtie \in \{\leq, <, \geq, >\}$

  $\psi ::= \top \mid (a,i) \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid \psi\, U\, \psi$
- The formula $P^{\max}_{\leq x}[\psi]$ is read as "with a scheduler (choice of action for each state) that maximises the probability of ψ occurring, this probability is $\leq x$."

Parameterised Model Checking

- Given a PPIIS $\mathcal{S}$ and an m-indexed PLTL formula ϕ, the parameterised model checking problem involves establishing whether it is the case that $\mathcal{S}(n) \models \phi$ for all $n \geq m$. We write $\mathcal{S} \models \phi$ if this is the case.
- We identify a partial decision procedure for this problem based on an abstract model that instead of keeping track of *how many* agents are in each state, will only keep track of which states have **one or more** agents at them.
- If the property holds in our abstract model, then it will hold in all concrete systems.

Implementation and Evaluation

- Our implementation is based on PRISM.
- We used our implementation to model a *foraging* protocol in which robot aim to find food and bring it back to a nest.
- We checked the property $P^{\max}_{\leq p}[F^{<k}\text{deposited2}]$ which says that for any choice of scheduling, the probability that 2 units of food are deposited within k steps does not exceed p.
- Note that when the property is true, we know it will be true in *all concrete systems of any size*. However, when it is false we cannot make any claim as our procedure is partial.

Results

		k				
		6	8	10	12	14
	0.25	False	False	False	False	False
	0.50	True	False	False	False	False
	0.75	True	True	False	False	False
p	0.90	True	True	True	False	False
	0.95	True	True	True	True	False
	0.99	True	True	True	True	True

Table: For different values of k and p, whether the property $P^{\max}_{\leq p}[F^{<k}\text{deposited2}]$ held in the abstract model.

- The abstract model takes around 50 seconds to construct and has 277,593 states and 8,880,150 transitions.
- The properties take a negligible amount of time to check (∼50ms).

Conclusion

- We have developed and implemented a method for verifying probabilistic swarm systems, and used our implementation to check a small example protocol.
- Our procedure targets the novel overlap of checking systems that are both probabilistic and have a possibly unbounded number of agents.
- We plan to continue work in this area by targetting more tractable procedures, further examples and richer specification logics.

VAS Group, Department of Computing, Imperial College London
a.lomuscio@ic.ac.uk, e.pirovano17@ic.ac.uk