Cutoff Techniques in the Verification of Open Multi-Agent Systems

Panagiotis Kouvaros and Alessio Lomuscio

Department of Computing, Imperial College London, UK

Parameterized Verification Workshop - 6 September 2014
Multi-agent systems (MAS)

Systems comprised of multiple components, namely agents, and their environment.

- Agents act autonomously and can act socially.
- The autonomous and social behaviour of MAS has been utilised in diverse applications.
  - Search and rescue,
  - Web-services,
  - Personal negotiation assistants.
- Growing need for verification before deployment.
Verification of MAS

Model checking: a leading verification technique.

- MAS model checking follows the AI tradition of reasoning about autonomous systems by ascribing high level properties.
- MAS specifications expressed as epistemic, BDI, alternating-time logic formulas.

Specification example: If an agent knows that its current goal is no longer achievable, then it will drop it at the next tick and begin replanning.
A MAS is composed of a set of agents $A = \{1, \ldots, n\}$ and an environment $e$.

Each agent is described by

- A set of local states $L_i$,
- A set of local actions $\text{Act}_i$,
- A local protocol function $P : L_i \rightarrow 2^{\text{Act}_i}$,
- An evolution function $\tau_i : L_i \times \text{Act}_1 \times \ldots \times \text{Act}_n \times \text{Act}_e \rightarrow L_i$.

Evolution by synchronous composition of $\tau_i$. 
Efficient Model Checking MAS

- Bounded Model Checking [PL03].
- BDD-based Model Checking [GvdM04,LQR05].
- Abstraction [CDLR08].
- Partial Order Reduction [LPQ10].

Implementations: MCMAS (Imperial), VerICS (Warsaw), MCK (Sydney). Widely used but number of agents is fixed.
MAS are often open systems.

- Swarm robotics,
- Multi-party negotiation protocols and auctions,
- Voting protocols,
- e-institutions.

**The problem:** Parameterised model checking for open MAS.

**Our contribution:** Cutoffs in the context of open MAS.
Parameterised Interpreted Systems

- All agents encoded by a single agent template.
- One environment modelled by an environment template.
- The agent and environment template define a generic system $S$, representing an arbitrary number of concrete systems.
- Given an $n \in \mathbb{N}$, a concrete system $S(n)$ is made of $n$ indexed instantiations of the template agent composed with a parameterised environment.
Actions and Synchronisation in PIS

PIS: Essential feature is how the agents interact between them and the environment.

- Agent template $\mathcal{T} = \langle L, \iota, Act, P, t \rangle$;
- Environment template $\mathcal{E} = \langle L_E, \iota_E, Act_E, P_E, t_E \rangle$.
- From IIS [LPQ10]: Only one local action performed in a concrete system at a given round; every agent that is potentially able to perform said action has to perform it at that round.
- Agent’s actions partitioned in
  - asynchronous (each concrete action admitted by exactly one agent),
  - agent-environment (each concrete action shared by exactly one agent and the environment)
  - global-synchronous (each concrete action shared by all the agents and the environment)
### Actions and Synchronisation in PIS

<table>
<thead>
<tr>
<th>Template action</th>
<th>( a \in A )</th>
<th>( b \in AE )</th>
<th>( c \in GS )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent 1</td>
<td>( a_1 \in A_1 )</td>
<td>( b_1 \in AE_1 )</td>
<td>( c \in GS_1 )</td>
</tr>
<tr>
<td>Agent ( i )</td>
<td>( a_i \in A_i )</td>
<td>( b_i \in AE_i )</td>
<td>( c \in GS_i )</td>
</tr>
<tr>
<td>Agent ( n )</td>
<td>( a_n \in A_n )</td>
<td>( b_n \in AE_n )</td>
<td>( c \in GS_n )</td>
</tr>
<tr>
<td>Environment</td>
<td>( {b_1, \ldots, b_n} \subseteq AE_E )</td>
<td>( c \in GS_E )</td>
<td></td>
</tr>
</tbody>
</table>
Global transitions from a concrete global state $g$ to a successor $g'$ can happen if:

- An asynchronous action is enabled for some agent at $g$.
- An agent-environment action is enabled for some agent and the environment at $g$.
- A global-synchronous action is enabled for all agents and the environment at $g$. 
Agent and environment templates of the untimed version of the Train-Gate-Controller.

\[ \text{Modelling Example} \]

\[ \text{Agent and environment templates of the untimed version of the Train-Gate-Controller.} \]

\[ \text{approach} \in A \]

\[ \text{enter} \in AE \]

\[ \text{exit} \in AE \]
Specs built from the logic indexed ACTL*_{K\setminus X}

- Combines indexed ACTL*_{K\setminus X} with indexed epistemic modalities.
- Epistemic modalities are interpreted over epistemic accessibility relations defined on local equalities for the agents’ states.
- Specifications range over $m$-tuples of distinct agents for $m \geq 1$.

$$\forall v_1 \ldots \forall v_m \phi(v_1, \ldots, v_m)$$

- Example specification for the TGC.

$$\forall i \forall j AG(T_i \rightarrow K_i \neg T_j)$$

“whenever an agent $i$ is in the tunnel, it knows agent $j$ is not”.
Consider the set $\Gamma(m)$ of all specifications with at most $m$ variables and a PIS $S$. $c \in \mathbb{N}$ is said to be a MAS cutoff for $S$ if

$$S(c) \models \phi \iff \forall n \geq c : S(n) \models \phi$$

for any $\phi \in \Gamma(m)$. 
Only a Sufficient Condition

**Lemma**

There are PIS that admit no cutoffs for any $\Gamma(m)$

---

where $a, b \in AE$ (agent-environment) and $d \in A$ (asynchronous). Specifications can count the loops on the composition above thereby counting the number of agents.
Simulations between Agent and Env Templates

Agent-environment simulations

A relation $\sim_{aes} \subseteq L \times L_E$ is an agent-environment simulation between $\mathcal{T}$ and $\mathcal{E}$ if

1. $\iota \sim_{aes} \iota_E$ and

2. whenever $l \sim_{aes} l_E$ then if there is $l', l'' \in L$ such that $l \xrightarrow{A*} l' \xrightarrow{a} l''$ for some $a \in AE \cup GS$, then $a \in P_E(l_E)$ and $l'' \sim_{aes} t_E(l_E, a)$.

1. $\iota \sim_{aes} \iota_E$

2. $l \sim_{aes} l_E$

$A$ transitions

\[ l \sim_{aes} l_E \]

\[ a \in S \]

\[ l' \sim_{aes} l_E \]

\[ l'' \sim_{aes} \]

$A$ transitions

\[ l' \sim_{aes} l_E \]

\[ a \in S \]

\[ l'' \sim_{aes} \]
Cutoff Identification

Cutoff theorem

For $\Gamma(m)$, if $T \leq_{aes} E$ then $c = \max(2, m)$.

The result relies on the semantical constraint that template actions for the environment are enabled at exactly one template state.

The constraint can be relaxed in the absence of agent-environment actions.

Cutoff theorem

For $\Gamma(m)$, if $AE = \emptyset$, then $c = m$. 
Implementation: MCMAS-P

Available for download from http://vas.doc.ic.ac.uk/.
Conclusions

- Model checking MAS now a relatively mature area of research, but many scalability issues remain.
- Current techniques to deal with state-explosion problem do not tackle unbounded number of components leading to limited applicability in many MAS applications.
- Novel parametric semantics.
- Cutoff technique for parametric temporal-epistemic specifications.
- Implementation.
Future work

- Generalised semantics.
- Multiple roles.
- Synchronous MAS and CTLK$\backslash X$. 