Verifying Auctions as Artifact Systems: Decidability via Finite Abstraction

Francesco Belardinelli
Laboratoire IBISC, Université d’Evry

based on work with Alessio Lomuscio
Imperial College London, UK

and Fabio Patrizi
Sapienza Università di Roma, Italy

Toulouse – 21 February 2014
Model Checking in one slide

Model checking: technique(s) to automatically verify that a system design $S$ satisfies a property $P$ before deployment.

More formally, given
- a model $M_S$ of system $S$
- a formula $\phi_P$ representing property $P$
we check that

$$M_S \models \phi_P$$
Turing Award 2007

(a) E. Clarke  
(CMU, USA)

(b) A. Emerson  
(U. Texas, USA)

(c) J. Sifakis  
(IMAG, F)

• Jury justification

“For their roles in developing model checking into a highly effective verification technology, widely adopted in the hardware and software industries.”
Motivation and Background:

- Artifact Systems as data-aware systems
- Parallel English (ascending bid) Auctions as Artifact Systems (eBay, etc.)
Overview

Motivation and Background:
- Artifact Systems as *data-aware* systems
- Parallel English (ascending bid) Auctions as Artifact Systems (eBay, etc.)

Main task: *Formal verification of infinite-state AS*
- model checking is appropriate for control-intensive applications...
- ...but less suited for data-intensive applications (data typically range over infinite domains) [1].
Motivation and Background:
- Artifact Systems as *data-aware* systems
- Parallel English (ascending bid) Auctions as Artifact Systems (eBay, etc.)

Main task: *Formal verification of infinite-state AS*
- model checking is appropriate for control-intensive applications...
- ...but less suited for data-intensive applications (data typically range over infinite domains) [1].

Key contribution:
- Verification of *bounded* and *uniform* AS is decidable
- Verification of Parallel English Auctions is decidable
Artifact Systems

Outline

- Recent paradigm in Service-Oriented Computing [2].
- **Motto**: let’s give *data* and *processes* the same relevance!
- **Artifact**: data model + lifecycle
  - (nested) records equipped with actions
  - actions may affect several artifacts
  - evolution stemming from the interaction with other artifacts/external actors
- **Artifact System**: interacting artifacts, representing services, manipulated by agents.
- **Auctions as Artifact Systems**
Artifact Systems
Order-to-Cash Scenario

Customer

<table>
<thead>
<tr>
<th>Purchase Order</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Desk</td>
<td>1</td>
</tr>
<tr>
<td>Chair</td>
<td>4</td>
</tr>
</tbody>
</table>

Manufacturer

<table>
<thead>
<tr>
<th>Material Order</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Desk Legs</td>
<td>4</td>
</tr>
<tr>
<td>Chair Legs</td>
<td>16</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Material Order</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Hammer Nails</td>
<td>1000</td>
</tr>
<tr>
<td>Glue</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Supplier

Accept/reject

Accept/reject
a single *auctioneer* $A$ and a finite number of *bidders* $B_1, \ldots, B_\ell$. 
a single *auctioneer* $A$ and a finite number of *bidders* $B_1, \ldots, B_\ell$. 

1. the auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders;
a single *auctioneer* $A$ and a finite number of *bidders* $B_1, \ldots, B_{\ell}$.

1. the auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
a single auctioneer $A$ and a finite number of bidders $B_1, \ldots, B_\ell$.

1. the auctioneer puts on sale a finite number of items, starting from a base price that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
3. at the end of the bidding process, the item is assigned to the bidder with the highest offer.
a single auctioneer $A$ and a finite number of bidders $B_1, \ldots, B_\ell$.

1. the auctioneer puts on sale a finite number of items, starting from a base price that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
3. at the end of the bidding process, the item is assigned to the bidder with the highest offer.

Assumptions:
a single auctioneer $A$ and a finite number of bidders $B_1, \ldots, B_\ell$.

1. the auctioneer puts on sale a finite number of items, starting from a base price that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
3. at the end of the bidding process, the item is assigned to the bidder with the highest offer.

Assumptions:
- each bidder is rational;
Artifact Systems
Parallel English (ascending bid) Auctions

a single *auctioneer* $A$ and a finite number of *bidders* $B_1, \ldots, B_\ell$.

1. the auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
3. at the end of the bidding process, the item is assigned to the bidder with the highest offer.

**Assumptions:**
- each bidder is rational;
- he has an *intrinsic value* for each item being auctioned;
a single *auctioneer* $A$ and a finite number of *bidders* $B_1, \ldots, B_\ell$.

1. the auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders;
2. at each round, the bidder can either choose to bid for a specific item or to skip the round;
3. at the end of the bidding process, the item is assigned to the bidder with the highest offer.

**Assumptions:**
- each bidder is rational;
- he has an *intrinsic value* for each item being auctioned;
- and he keeps this information private from other bidders and the auctioneer.
## Artifact Systems
### Auction Data Model

<table>
<thead>
<tr>
<th>Bidding</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>base_price</td>
</tr>
</tbody>
</table>

- \( init_A(item, base\_price) \)
- \( bid_i(item, bid) \)
- \( time\_out(item) \)
- \( skip_A \)
- \( skip_i \)
- ... 

<table>
<thead>
<tr>
<th>trueValue_i</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>item</td>
<td>true_value</td>
</tr>
</tbody>
</table>

- \( init_i(item, true\_value) \)
- ...
Artifact Systems
Auction Lifecycle

- Agents operate on artifacts.
  - e.g., the bidder sends a new bid to the auctioneer.

- Actions add/remove artifacts or change artifact attributes.
  - e.g., the auctioneer puts a new item on auction.

- The whole system can be seen as a *data-aware* dynamic system.
  - at every step, an action yields a change in the current state.
Research questions

1. Which syntax and semantics to specify AS?

2. Is verification of AS decidable?

3. If not, can we identify relevant fragments that are reasonably well-behaved?

4. How can we implement this?
Research questions

1. Which syntax and semantics to specify AS?
2. Is verification of AS decidable?
Research questions

1. Which syntax and semantics to specify AS?
2. Is verification of AS decidable?
3. If not, can we identify relevant fragments that are reasonably well-behaved?
Research questions

1. Which syntax and semantics to specify AS?
2. Is verification of AS decidable?
3. If not, can we identify relevant fragments that are reasonably well-behaved?
4. How can we implement this?
Challenges

Multi-agent systems, but . . .
Challenges

Multi-agent systems, but . . .

- . . . states have a relational structure,
Challenges

Multi-agent systems, but . . .

• . . . states have a relational structure,

• data are potentially infinite,
Challenges

Multi-agent systems, but . . .

• . . . states have a relational structure,
• data are potentially infinite,
• the state space is infinite in general.
Multi-agent systems, but . . .

- . . . states have a relational structure,
- data are potentially infinite,
- the state space is infinite in general.

⇒ The model checking problem cannot be tackled by standard techniques.
Artifact Systems

Results

Artifact-centric multi-agent systems (AC-MAS) as a formal model for AS.

Intuition: databases (?) that evolve in time and are manipulated by agents.
Artifact Systems

Results

1. **Artifact-centric multi-agent systems (AC-MAS)** as a formal model for AS.
   **Intuition:** databases (?) that evolve in time and are manipulated by agents.

2. FO-CTLK as a specification language:

   \[
   \text{AG } \forall \text{it}, \vec{bd}, s(\exists! bp \ Bidding(it, \vec{bd}, bp, s) \land \exists^{\leq 1} tv \ trueValue_i(it, tv))
   \]

   for each item there is exactly one base price, while bidders associate at most one true value to each item (possibly none).
1. **Artifact-centric multi-agent systems (AC-MAS)** as a formal model for AS.
   **Intuition:** databases (?) that evolve in time and are manipulated by agents.

2. FO-CTLK as a specification language:
   \[
   AG \forall it, \vec{bd}, s(\exists! bp \ Bidding(it, \vec{bd}, bp, s) \land \exists^{\leq 1} tv \ trueValue_i(it, tv))
   \]
   for each item there is exactly one base price, while bidders associate at most one true value to each item (possibly none).

3. Abstraction techniques and finite interpretation to tackle model checking.
   **Main result:** under specific conditions MC can be reduced to the finite case.
Artifact Systems

Results

1. **Artifact-centric multi-agent systems** (AC-MAS) as a formal model for AS.
   **Intuition:** databases (?) that evolve in time and are manipulated by agents.

2. FO-CTLK as a specification language:
   \[
   AG \forall \text{it}, \vec{bd}, s(\exists! bp \text{ Bidding}(\text{it}, \vec{bd}, \text{bp}, s) \land \exists^{\leq 1} tv \text{ trueValue}_i(\text{it}, tv))
   \]
   For each item there is exactly one base price, while bidders associate at most one true value to each item (possibly none).

3. Abstraction techniques and finite interpretation to tackle model checking.
   **Main result:** under specific conditions MC can be reduced to the finite case.

4. **Case study:** modelling and verifying auctions as AC-MAS.
Semantics: Databases

The data model of AS is given as a particular kind of database.

- a **database schema** is a finite set \( \mathcal{D} = \{ P_1/a_1, \ldots, P_n/a_n, Q_1/b_1, \ldots, Q_m/b_m \} \) of (typed) relation symbols \( R_i \) with arity \( c_i \in \mathbb{N} \).

- an **instance** on a domain \( U \) is a mapping \( D \) associating
  - each symbol \( P_i \) with a finite \( a_i \)-ary relation on \( U \)
  - each symbol \( Q_i \) with a (possibly infinite) \( b_i \)-ary relation on \( U \)

- the **active domain** \( \text{adom}(D) \) is the set of all \( u \in U \) appearing in some \( D(P_i) \).

- the **disjoint union** \( D \oplus D' \) is the \( (\mathcal{D} \cup \mathcal{D}') \)-interpretation s.t.
  \[
  \begin{align*}
  (i) & \quad D \oplus D'(R_i) = D(R_i) \\
  (ii) & \quad D \oplus D'(R'_i) = D'(R_i)
  \end{align*}
  \]

We consider untyped languages; the extension to types is not problematic.
Artifact-centric Multi-agent Systems

Agents have partial access (views) to the artifact system.

- An agent is a tuple $A_i = \langle D_i, Act_i, Pr_i \rangle$ where
  - $D_i$ is the local database schema
  - $Act_i$ is the set of local actions $\alpha(\vec{x})$ with parameters $\vec{x}$
  - $Pr_i : D_i(U) \mapsto 2^{Act_i(U)}$ is the local protocol function

- the setting is reminiscent of the interpreted systems semantics for MAS [3], ...

- ...but here the local state of each agent is relational.

Intuitively, agents manipulate artifacts and have (partial) access to the information contained in the global db schema $\mathcal{D} = \mathcal{D}_1 \cup \cdots \cup \mathcal{D}_\ell$. 
Example 1: Parallel English (ascending bid) Auction

- **Agents**: Auctioneer, Bidder$_1$, ..., Bidder$_\ell$

- **local db schema** $\mathcal{D}_A$ for the auctioneer
  - $\text{Bidding}(\text{item}, \text{base\_price}, \text{bid}_1, \ldots, \text{bid}_\ell, \text{status})$
  - $\prec$ on $\mathbb{Q}$

- **local db schema** $\mathcal{D}_i$ for the bidders
  - $\text{Bidding}(\text{item}, \text{base\_price}, \text{bid}_1, \ldots, \text{bid}_\ell, \text{status})$
  - $\text{TValue}_i(\text{item}, \text{true\_value})$
  - $\prec$ on $\mathbb{Q}$

- **then**, $\mathcal{D} = \{\text{Bidding}, \text{TValue}_1, \ldots, \text{TValue}_\ell, \prec\}$

- **Actions** introduce values from an infinite domain $U = \text{Items} \cup \mathbb{Q} \cup \{\text{active}, \text{term}\}$:
  - $\text{init}_A(\text{item}, \text{base\_price}), \text{time\_out}(\text{item}), \text{skip}_A$ belong to $\text{Act}_A$
  - $\text{init}_i(\text{item}, \text{true\_value}), \text{bid}_i(\text{item}, \text{bid}), \text{skip}_i$ belong to $\text{Act}_i$

- **The protocol function** specifies the preconditions for actions:
  - e.g., $\text{bid}_i(\text{item}, \text{bid}) \in \text{Pr}_i(D)$ whenever $\text{item}$ appears in $D(\text{TValue}_i)$, the highest bid $\text{bid}_j$ in $\text{Bidding}$, $j \neq i$, for $\text{item}$ is $\prec \text{true\_value}$ for bidder $B_i$, $\text{bid}_j < \text{bid} \leq \text{true\_value}$, and $D(\text{status}) = \text{active}$ for $\text{item}$.
  - the $\text{skip}$ actions are always enabled.
Artifact-centric Multi-agent Systems

AC-MAS

Agents are modules that can be composed together to obtain AC-MAS.

- **Global states** are tuples \( s = \langle D_0, \ldots, D_\ell \rangle \in \mathcal{D}(U) \).

- An **AC-MAS** is a tuple \( \mathcal{P} = \langle \text{Ag}, s_0, \rightarrow \rangle \) where
  - \( \text{Ag} = \{A_0, \ldots, A_\ell\} \) is a **finite set of agents**
  - \( s_0 \in \mathcal{D}(U) \) is the **initial global state**
  - \( s \xrightarrow{\alpha(\overline{u})} s' \) is the **transition relation**

- **Epistemic relation**: \( s \sim_i s' \) iff \( D_i = D'_i \)

- An AC-MAS \( \mathcal{P} \) is **rigid** iff for all states \( s, s' \), symbols \( Q \), and agents \( A_i, A_j \in \text{Ag} \), \( D_i(Q) = D'_j(Q) \).

- AC-MAS are infinite-state systems in general

AC-MAS are first-order temporal epistemic structures.  
\( \Rightarrow \) FO-CTLK can be used as a specification language.
Example 2: the Auction AC-MAS

The Auction AC-MAS $\mathcal{A} = \langle Ag, s_0, \rightarrow \rangle$ is defined as

- $Ag = \{A, B_1, \ldots, B_\ell\}$
- $s_0$ is the empty interpretation of $D = \{Bidding, TValue_1, \ldots, TValue_\ell, <\}$ but for $<$
- $\rightarrow$ is the transition relation s.t. $s \xrightarrow{\alpha(\bar{u})} s'$ whenever
  - $\alpha_i = bid_i(item, bid')$ and $s'$ modifies $s$ by replacing any tuple $(item, \ldots, bid_i, \ldots, status)$ in $D_s(Bidding)$ with $(item, \ldots, bid'_i, \ldots, status)$
  - $\alpha_A = timeout(item)$ and the value of $status$ in $D_{s'}(Bidding)$ for $item$ is term
  - $\ldots$

Notice:

- the auction AC-MAS $\mathcal{A}$ is rigid
- actions preserve the consistency of the underlying database
- the active domain $adom(s_0)$ is empty
Syntax: FO-CTLK

- Data call for First-order Logic.
- Evolution calls for Temporal Logic.
- Agents (operating on artifacts) call for Epistemic Logic.

The specification language **FO-CTLK**:

\[ \varphi ::= R(t_1, \ldots, t_c) \mid t = t' \mid \neg \varphi \mid \varphi \rightarrow \varphi \mid \forall x \varphi \mid AX \varphi \mid A\varphi U \varphi \mid E \varphi U \varphi \mid K_i \varphi \]

Alternation of free variables and modal operators is enabled.
Semantics of FO-CTLK

Formal definition

An AC-MAS $\mathcal{P}$ satisfies an FO-CTLK-formula $\varphi$ in a state $s$ for an assignment $\sigma$, iff

\begin{align*}
(\mathcal{P}, s, \sigma) &\models R(t) \quad \text{iff} \quad \langle \sigma(t_1), \ldots, \sigma(t_c) \rangle \in D_s(R) \\
(\mathcal{P}, s, \sigma) &\models t = t' \quad \text{iff} \quad \sigma(t) = \sigma(t') \\
(\mathcal{P}, s, \sigma) &\models \neg \varphi \quad \text{iff} \quad (\mathcal{P}, s, \sigma) \not\models \varphi \\
(\mathcal{P}, s, \sigma) &\models \varphi \rightarrow \psi \quad \text{iff} \quad (\mathcal{P}, s, \sigma) \not\models \varphi \text{ or } (\mathcal{P}, s, \sigma) \models \psi \\
(\mathcal{P}, s, \sigma) &\models \forall x \varphi \quad \text{iff} \quad \text{for all } u \in \text{adom}(s), \ (\mathcal{P}, s, \sigma^x_u) \models \varphi \\
(\mathcal{P}, s, \sigma) &\models AX \varphi \quad \text{iff} \quad \text{for all runs } r, r(0) = s \text{ implies } (\mathcal{P}, r(1), \sigma) \models \varphi \\
(\mathcal{P}, s, \sigma) &\models A\varphi U \varphi' \quad \text{iff} \quad \text{for all runs } r, r(0) = s \text{ implies } (\mathcal{P}, r(k), \sigma) \models \varphi' \text{ for some } k \geq 0, \text{ and } (\mathcal{P}, r(k'), \sigma) \models \varphi \text{ for all } 0 \leq k' < k \\
(\mathcal{P}, s, \sigma) &\models E\varphi U \varphi' \quad \text{iff} \quad \text{there exists } r \text{ s.t. } r(0) = s, \ (\mathcal{P}, r(k), \sigma) \models \varphi' \text{ for some } k \geq 0, \text{ and } (\mathcal{P}, r(k'), \sigma) \models \varphi \text{ for all } 0 \leq k' < k \\
(\mathcal{P}, s, \sigma) &\models K_i \varphi \quad \text{iff} \quad \text{for all states } s', s \sim_i s' \text{ implies } (\mathcal{P}, s', \sigma) \models \varphi
\end{align*}

- Active-domain semantics, but...
  - ...we can refer to individuals that no longer exist
  - the number of states is infinite in general
Semantics of FO-CTLK

Intuition

(d) $AX\varphi$

(e) $A\varphi U\psi$

(f) $E\varphi U\psi$
Verification of AC-MAS

How do we verify FO-CTLK specifications on auctions?

- the true value of items for each bidder is secret to all other bidders and to the auctioneer:
  \[ AG \forall \text{item} \neg \exists \text{true\_value} \bigvee_{j \neq i \vee j = A} K_j \text{TVvalue}_i(\text{item, true\_value}) \]

- for each bidder, each bid is less or equal to her true value:
  \[ AG \forall \text{it}, \vec{x}, \text{bd}_i, \vec{y}, \text{tv}(\text{Bidding}(\text{it, } \vec{x}, \text{bd}_i, \vec{y}) \land \text{TVvalue}_i(\text{it, tv}) \rightarrow \text{bd}_i \leq \text{tv}) \]

- each bidder can raise her bid unless she has already hit her true value:
  \[ AG \forall \text{it}, \vec{x}, \text{bd}_i, \vec{y}(\text{Bidding}(\text{it, } \vec{x}, \text{bd}_i, \vec{y}) \rightarrow \rightarrow (\text{TVvalue}_i(\text{it, bd}_i) \lor \text{EF } \exists \vec{x}', \text{bd}'_i, \vec{y}'(\text{bd}'_i > \text{bd}_i \land \text{Bidding}(\text{it, } \vec{x}', \text{bd}'_i, \vec{y}'))) ) \]

**Problem:** the infinite domain \( U \) may generate infinitely many states!

**Investigated solution:** can we *simulate* the concrete values from \( U \) with a finite set of *abstract* symbols?
Abstraction: Isomorphism and Bisimulation

- two states $s, s'$ are *isomorphic*, or $s \simeq s'$, if there is a bijection

$$\iota : \text{dom}(s) \cup C \mapsto \text{dom}(s') \cup C$$

such that

- $\iota$ is the identity on $C$
- for every $\bar{u}$ in $\text{dom}(s)$, $A_i \in Ag$, $\bar{u} \in D_i(R) \iff \iota(\bar{u}) \in D'_i(R)$

<table>
<thead>
<tr>
<th></th>
<th>$D(R)$</th>
<th>$D'(R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$a$  $b$</td>
<td>$A_1$  $1$  $2$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$b$  $c$</td>
<td>$A_2$  $2$  $c$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$d$  $e$</td>
<td>$A_3$  $4$  $5$</td>
</tr>
</tbody>
</table>

- $\iota : a \mapsto 1$
  - $b \mapsto 2$
  - $c \mapsto c$
  - $d \mapsto 4$
  - $e \mapsto 5$
Abstraction: Isomorphism and Bisimulation

- two states $s, s'$ are **bisimilar**, or $s \approx s'$, if
  1. $s \approx s'$
  2. if $s \rightarrow t$ then there is $t'$ s.t. $s' \rightarrow t'$, $s \oplus t \approx s' \oplus t'$, and $t \approx t'$
two states \( s, s' \) are *bisimilar*, or \( s \approx s' \), if

1. \( s \approx s' \)
2. if \( s \rightarrow t \) then there is \( t' \) s.t. \( s' \rightarrow t' \), \( s \oplus t \approx s' \oplus t' \), and \( t \approx t' \)

The other direction holds as well

similar conditions for the epistemic relation \( \sim_i \)
Abstraction: Isomorphism and Bisimulation

However, bisimulation is not sufficient to preserve FO-CTLK formulas:

\[ \phi = \forall x \ (P(x) \rightarrow AX AG \neg P(x)) \]
Uniformity

• Intuitively, the behaviour of uniform AC-MAS is *independent* from data not explicitly named in the system description.
Uniformity

- Intuitively, the behaviour of uniform AC-MAS is *independent* from data not explicitly named in the system description.
- More formally, an AC-MAS $\mathcal{P}$ is *uniform* iff for $s, t, s' \in S$ and $t' \in D(U)$:

$$s \rightarrow t \text{ and } s \oplus t \simeq s' \oplus t' \text{ imply } s' \rightarrow t'$$

\[
\begin{array}{cc}
\text{s} & \text{t} \\
\hline 
a & b \\
b & c \\
d & e \\
\end{array}
\quad \quad \quad
\begin{array}{cc}
\text{t} & \\
\hline 
a & f \\
f & c \\
\end{array}
\]

\[
\begin{array}{cc}
\text{s'} & \\
\hline 
1 & 2 \\
2 & c \\
4 & 5 \\
\end{array}
\quad \quad \quad
\begin{array}{cc}
\text{t'} & \\
\hline 
1 & 6 \\
6 & c \\
\end{array}
\]
Uniformity

- Intuitively, the behaviour of uniform AC-MAS is *independent* from data not explicitly named in the system description.
- More formally, an AC-MAS $\mathcal{P}$ is *uniform* iff for $s, t, s', t' \in S$ and $t' \in D(U)$:

1. $s \rightarrow t$ and $s \oplus t \simeq s' \oplus t'$ imply $s' \rightarrow t'$

Also, rigid AC-MAS must satisfy a condition akin to density of $<$ on $\mathbb{Q}$.
Uniformity

- Intuitively, the behaviour of uniform AC-MAS is *independent* from data not explicitly named in the system description.
- More formally, an AC-MAS $\mathcal{P}$ is *uniform* iff for $s, t, s' \in S$ and $t' \in \mathcal{D}(U)$:

$$s \rightarrow t \text{ and } s \oplus t \simeq s' \oplus t' \text{ imply } s' \rightarrow t'$$

1. Also, rigid AC-MAS must satisfy a condition akin to density of $<$ on $\mathbb{Q}$.

- Uniform AC-MAS cover a number of interesting cases [2, 4], including the auction AC-MAS $\mathcal{A}$.
Consider

- bisimilar and uniform AC-MAS $P$ and $P'$
- an FO-CTLK formula $\varphi$

If

1. $|U'| \geq 2 \cdot \sup_{s \in P} \{|\text{adom}(s)| + |C| + |\text{vars}(\varphi)|\}$
2. $|U| \geq 2 \cdot \sup_{s' \in P'} \{|\text{adom}(s')| + |C| + |\text{vars}(\varphi)|\}$

then

$P \models \varphi \iff P' \models \varphi$

Can we apply this result to finite abstraction?
Abstraction

Abstractions are defined in an agent-based, modular way.

- Let $A = \langle D, Act, Pr \rangle$ be an agent defined on the domain $U$. Given a domain $U'$, the abstract agent $A' = \langle D, Act, Pr' \rangle$ on $U'$ is s.t.
  - $Pr'$ is the smallest function s.t. if $\alpha(\bar{u}) \in Pr(D)$, $D' \in D'(U')$ and $D' \simeq D$ for some witness $\iota$, then $\alpha(\bar{u}') \in Pr'(D')$ where $\bar{u}' = \iota'(\bar{u})$ for some constant-preserving bijection $\iota'$ extending $\iota$ to $\bar{u}$.

- Let $P = \langle Ag, s_0, \rightarrow \rangle$ be an AC-MAS. The abstraction $P' = \langle Ag', s'_0, \rightarrow' \rangle$ of $P$ is an AC-MAS s.t.
  - $Ag'$ be the set of abstract agents on $U'$
  - $s'_0 \simeq s_0$
  - $\rightarrow'$ is the smallest function s.t. if $s \xrightarrow{\alpha(\bar{u})} t$, and $s \oplus t \simeq s' \oplus t'$ for some witness $\iota$, then $s' \xrightarrow{\alpha(\iota'(\bar{u}))} t'$ for some constant-preserving bijection $\iota'$ extending $\iota$ to $\bar{u}$.

The abstraction of a rigid AC-MAS is not necessarily rigid!
Abstraction

Let \( N_{Ag} = \sum_{A_i \in Ag} \max_{\alpha(\vec{x}) \in \text{Act}_i} |\vec{x}| \) be the sum of the maximum numbers of parameters contained in the action types of each agent.

**Lemma**

Consider

- a uniform and rigid AC-MAS \( \mathcal{P} \)
- a set \( U' \supseteq C \) s.t. \( |U'| \geq 2 \sup_{s \in \mathcal{P}} |\text{dom}(s)| + |C| + N_{Ag} \)

Then, there exists an abstraction \( \mathcal{P}' \) of \( \mathcal{P} \) that is uniform and bisimilar to \( \mathcal{P} \).

How can we define finite abstractions?
Bounded Models and Finite Abstractions

- An AC-MAS $\mathcal{P}$ is \textit{b-bounded} iff for all $s \in \mathcal{P}$, $|\text{adom}(s)| \leq b$.
- Bounded systems can still be infinite!

**Theorem**

Consider

- a \textit{b-bounded}, uniform and rigid AC-MAS $\mathcal{P}$ on an infinite domain $U$
- an FO-CTLK formula $\varphi$

Given a finite $U' \supseteq C$ s.t.

$$|U'| \geq 2b + |C| + \max\{|\text{vars}(\varphi)|, N_{Ag}\}$$

there exists a \textit{finite abstraction} $\mathcal{P}'$ of $\mathcal{P}$ s.t.

- $\mathcal{P}'$ is uniform and bisimilar to $\mathcal{P}$

In particular,

$$\mathcal{P} \models \varphi \iff \mathcal{P}' \models \varphi$$

$\Rightarrow$ Under specific circumstances, we can model check an infinite-state system by verifying its finite abstraction.
Finite Abstract Auction I

- **Notice:** the auction AC-MAS $A$ is bounded by $b = |\text{Items}|(2|\text{Ag}| - 1) + 2$

- Consider a finite $U' \geq 2b + \text{vars}(\phi)$

- Abstract agents $\text{Auctioneer} A'$ and $\text{Bidders} B_i'$
  - the local db schemas $D_A'$ and $D_i'$ are the same as for $A$ and $B_i$
  - the sets of actions $\text{Act}_A'$ and $\text{Act}_i'$ are the same as for $A$ and $B_i$
  - the protocol function $Pr_A'$ is the same as for $A$
  - as to $Pr_i', bid_i(item, bid) \in Pr_i'(D')$ whenever item appears in $D'(TValue_i)$, the highest bid $bid_j$ in $Bidding$, $j \neq i$, for item is $<\text{true\_value}$ for bidder $B_i$, and bid is an abstract value that does not represent any bid in $D'$, and for item, $D'(status) = \text{active}$.
The abstract auction $AC-MAS \; \mathcal{A}' = \langle Ag', s'_0, \tau' \rangle$ is defined as

- $Ag' = \{ A', B'_1, \ldots, B'_\ell \}$

- $s'_0$ is the empty interpretation of $D$

- $\rightarrow'$ mimics $\rightarrow$
  
  - e.g., if $\alpha_i = bid_i(item, bid)$, then $s \xrightarrow{\alpha(\bar{u})'} t$ whenever $t$ is the db instance that modifies $s$ by replacing any tuple $(item, \ldots, bid_i, \ldots, status)$ in $D_s(Bidding)$ with $(item, \ldots, bid'_i, \ldots, status)$, where the value $bid' \in U'$ has been found as above. In particular, $bid < bid' \leq true\_value$ in $t$.

- By the assumption that $U' \geq 2b + vars(\phi)$ and Theorem 3 we have that $\mathcal{A}'$ is a finite abstraction of $\mathcal{A}$. In particular,
  
  - $\mathcal{A}'$ is uniform and bisimilar to $\mathcal{A}$ (but not rigid) and

  $$\mathcal{A} \models \varphi \iff \mathcal{A}' \models \varphi$$
Extensions

Non-uniform AC-MAS: for sentence-atomic FO-CTL the results above still hold.

\[ \forall it, \vec{bd}, s (\exists! bp \ Bidding(it, \vec{bd}, bp, s) \land \exists^{\leq 1} tv \ TValue_i(it, tv)) \]
Extensions

1. Non-uniform AC-MAS: for *sentence-atomic* FO-CTL the results above still hold.

   \[ \text{AG } \forall it, \vec{bd}, s(\exists!bp \ Bidding(it, \vec{bd}, bp, s) \land \exists \leq 1 tv \ TValue_i(it, tv)) \]


**Theorem**

*For every AC-MAS \( \mathcal{P} \) and \( \varphi \in \text{FO-ACTLK}^- \), there exists a finite abstraction \( \mathcal{P}' \) s.t.*

\[ \mathcal{P}' \models \varphi \implies \mathcal{P} \models \varphi \]
Extensions

1. Non-uniform AC-MAS: for sentence-atomic FO-CTL the results above still hold.

   \[ AG \forall it, \vec{bd}, s(\exists! bp \text{ Bidding}(it, \vec{bd}, bp, s) \land \exists_{\leq 1} tv \text{ TValue}_i(it, tv)) \]


**Theorem**

*For every AC-MAS \( P \) and \( \varphi \in \text{FO-ACTLK}^- \), there exists a finite abstraction \( P' \) s.t.*

\[ P' \models \varphi \implies P \models \varphi \]

3. *Model checking bounded* AC-MAS w.r.t. FO-CTL is undecidable.
Extensions

1. Non-uniform AC-MAS: for sentence-atomic FO-CTL the results above still hold.

\[ AG \forall it, \vec{bd}, s(\exists!bp \ Bidding(it, \vec{bd}, bp, s) \land \exists^{\leq 1} tv \ TValue_{i}(it, tv)) \]


**Theorem**

For every AC-MAS \( P \) and \( \varphi \in FO-ACTLK^- \), there exists a finite abstraction \( P' \) s.t.

\[ P' \models \varphi \Rightarrow P \models \varphi \]

3. Model checking bounded AC-MAS w.r.t. FO-CTL is undecidable.

4. Complexity result:

**Theorem**

The model checking problem for finite AC-MAS w.r.t. FO-CTLK is EXPSPACE-complete in the size of the formula and data.
Results and main limitations

• We are able to model check AC-MAS w.r.t. full FO-CTLK...

• ...however, our results hold only for rigid, uniform and bounded systems.

• This class includes many interesting systems (AS programs, [2, 4]).

• The model checking problem is EXPSPACE-complete.
Next Steps

- Techniques for finite abstraction.
- Model checking techniques for finite-state systems are effective on the abstract system?
- How to perform the boundedness check.
References

Christel Baier and Joost-Pieter Katoen.  
*Principles of Model Checking.*  

D. Cohn and R. Hull.  

*Reasoning About Knowledge.*  

B. Bagheri Hariri, D. Calvanese, G. De Giacomo, R. De Masellis, and P. Felli.  
Foundations of Relational Artifacts Verification.  