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

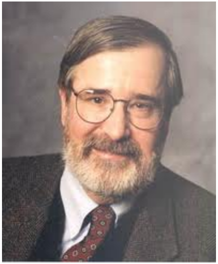
- a model $\mathcal{M}_S$ of system S
- a formula ϕ_P representing property P

we check that

$$\mathcal{M}_S \models \phi_P$$

Turing Award 2007

www.acm.org/press-room/news-releases-2008/turing-award-07



(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."

Overview

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

Overview

1. Motivation and Background:
   - Artifact Systems as *data-aware* systems
   - Parallel English (ascending bid) Auctions as Artifact Systems (eBay, etc.)
2. 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].

Overview

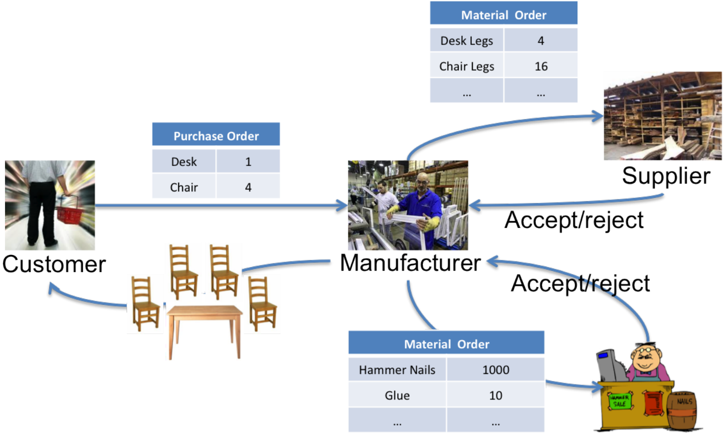
1. Motivation and Background:
   - Artifact Systems as *data-aware* systems
   - Parallel English (ascending bid) Auctions as Artifact Systems (eBay, etc.)
2. Main task: *Formal* verification of *infinite-state* AS
   - model checking is appropriate for control-intensive applications...
   - ...but less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

Artifact Systems

- 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



Purchase Order	
Desk	1
Chair	4

Material Order	
Desk Legs	4
Chair Legs	16
...	...

Material Order	
Hammer Nails	1000
Glue	10
...	...

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;

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;

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;
- and he keeps this information private from other bidders and the auctioneer.

Artifact Systems

Auction Data Model

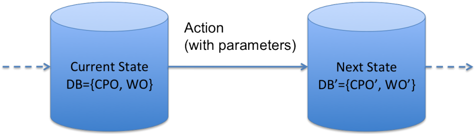
Bidding					
item	*base_price*	bid_1	...	bid_ℓ	*status*

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

$trueValue_i$	
item	*true_value*

- $init_i(item, true\_value)$
- ...

- 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?

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.

Challenges

Multi-agent systems, but . . .

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

$\Rightarrow$ The model checking problem cannot be tackled by standard techniques.

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.

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).

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:
   $$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ trueValue_i(it, tv))$$
   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.

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:

$$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ trueValue_i(it, tv))$$

   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.

4. Case study: modelling and veryfing 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* $adom(D)$ 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.
  (i) $D \oplus D'(R_i) = D(R_i)$
  (ii) $D \oplus D'(R_i') = D'(R_i)$

We consider untyped languages; the extension to types is not problematic.

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

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

- local db schema $\mathcal{D}_A$ for the auctioneer
  - *Bidding(item, base_price,* bid_1, ..., bid_ℓ, *status)*
  - $<$ on $\mathbb{Q}$
- local db schema $\mathcal{D}_i$ for the bidders
  - *Bidding(item, base_price,* bid_1, ..., bid_ℓ, *status)*
  - $TValue_i$*(item, true_value)*
  - $<$ on $\mathbb{Q}$
- then, $\mathcal{D} = \{Bidding, TValue_1, \ldots, TValue_\ell, <\}$

- Actions introduce values from an infinite domain $U = Items \cup \mathbb{Q} \cup \{active, term\}$:
  - $init_A$*(item, base_price), time out(item),* $skip_A$ belong to Act_A
  - $init_i$*(item, true_value),* bid_i*(item, bid),* $skip_i$ belong to Act_i

- The protocol function specifies the preconditions for actions:
  - e.g., $bid_i(item, bid) \in Pr_i(D)$ whenever *item* appears in $D(TValue_i)$, the highest bid bid_j in *Bidding*, $j \neq i$, for *item* is $<$ *true_value* for bidder B_i, $bid_j < bid \leq true\_value$, and $D(status) = active$ for *item*.
  - the *skip* actions are always enabled.

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 $\mathcal{P} = \langle Ag, s_0, \rightarrow \rangle$ where
  - $Ag = \{A_0, \ldots, A_\ell\}$ is a *finite set of agents*
  - $s_0 \in \mathcal{D}(U)$ is the *initial global state*
  - $s \xrightarrow{\alpha(\vec{u})} s'$ 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, $A_j \in Ag$, $D_i(Q) = D'_j(Q)$.
- 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 $\mathcal{D} = \{Bidding, TValue_1, \ldots, TValue_\ell, <\}$ but for $<$
- $\rightarrow$ is the *transition relation* s.t. $s \xrightarrow{\alpha(\vec{u})} s'$ 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:

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

Alternation of free variables and modal operators is enabled.

Semantics of FO-CTLK

Formal definition

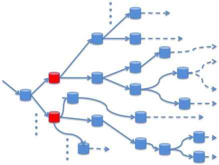
An AC-MAS $\mathcal{P}$ satisfies an FO-CTLK-formula φ in a state s for an assignment σ, iff

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

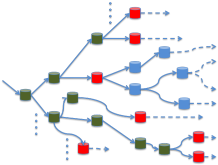
- Active-domain semantics, but...
  - ...we can refer to individuals that no longer exist
  - the number of states is infinite in general

Semantics of FO-CTLK

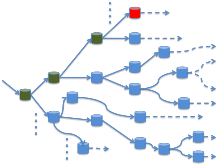
Intuition



(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:

$$AG\ \forall item\ \neg\exists true\_value \bigvee_{j \neq i \vee j = A} K_j\ TValue_i(item, true\_value)$$

- for each bidder, each bid is less or equal to her true value:

$$AG\ \forall it, \vec{x}, bd_i, \vec{y}, tv(Bidding(it, \vec{x}, bd_i, \vec{y}) \wedge TValue_i(it, tv) \rightarrow bd_i \leq tv)$$

- each bidder can raise her bid unless she has already hit her true value:

$$\begin{aligned} & AG\ \forall it, \vec{x}, bd_i, \vec{y}(Bidding(it, \vec{x}, bd_i, \vec{y}) \rightarrow \\ & \rightarrow (TValue_i(it, bd_i) \vee EF\ \exists \vec{x}', bd_i', \vec{y}'(bd_i' > bd_i \wedge Bidding(it, \vec{x}', bd_i', \vec{y}')))) \end{aligned}$$

<u>Problem</u>: the infinite domain U may generate infinitely many states!

<u>Investigated solution</u>: 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

$$\iota : adom(s) \cup C \mapsto adom(s') \cup C$$

such that

  - ι is the identity on C
  - for every $\vec{u}$ in $adom(s)$, $A_i \in Ag$, $\vec{u} \in D_i(R) \Leftrightarrow \iota(\vec{u}) \in D'_i(R)$

	$D(R)$	
A_1	a	b
A_2	b	c
A_3	d	e

$\simeq$

	$D'(R)$	
A_1	1	2
A_2	2	c
A_3	4	5

  - $$\begin{aligned} \iota : a &\mapsto 1 \\ b &\mapsto 2 \\ c &\mapsto c \\ d &\mapsto 4 \\ e &\mapsto 5 \end{aligned}$$

Abstraction: Isomorphism and Bisimulation

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

$$
\begin{tikzcd}
\boxed{s} \arrow[r] \arrow[d, phantom, "\approx"] & \boxed{t} \\
\boxed{s'} &
\end{tikzcd}
$$

Abstraction: Isomorphism and Bisimulation

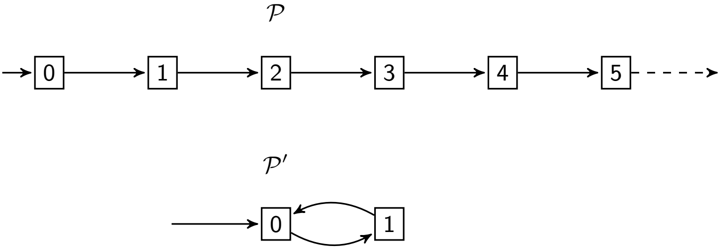
- two states s, s' are *bisimilar*, or $s \approx s'$, if
  1. $s \simeq s'$
  2. if $s \longrightarrow t$ then there is t' s.t. $s' \longrightarrow t'$, $s \oplus t \simeq s' \oplus t'$, and $t \approx t'$

$$\begin{tikzcd}
\boxed{s} \arrow[r] \arrow[d, phantom, "\approx"] & \boxed{t} \arrow[d, phantom, "\approx"] \\
\boxed{s'} \arrow[r] & \boxed{t'}
\end{tikzcd}$$

  3. the other direction holds as well
  4. similar conditions for the epistemic relation $\sim_i$

Abstraction: Isomorphism and Bisimulation

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



$$\phi \quad = \quad AG\ \forall x\ (P(x) \rightarrow AX\ AG\ \neg P(x))$$

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 $s, t, s' \in \mathcal{S}$ and $t' \in \mathcal{D}(U)$:

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

s	
a	b
b	c
d	e

$\longrightarrow$

t	
a	f
f	c

s'	
1	2
2	c
4	5

t'	
1	6
6	c

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 $s, t, s' \in \mathcal{S}$ and $t' \in \mathcal{D}(U)$:

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

s	
a	b
b	c
d	e

$\longrightarrow$

t	
a	f
f	c

s'	
1	2
2	c
4	5

$\longrightarrow$

t'	
1	6
6	c

2. 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 $s, t, s' \in \mathcal{S}$ and $t' \in \mathcal{D}(U)$:

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

s	
a	b
b	c
d	e

$\longrightarrow$

t	
a	f
f	c

s'	
1	2
2	c
4	5

$\longrightarrow$

t'	
1	6
6	c

2. 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 $\mathcal{A}$.

Bisimulation and Equivalence w.r.t. FO-CTLK

Theorem

Consider

- bisimilar *and* uniform *AC-MAS* $\mathcal{P}$ *and* $\mathcal{P}'$
- *an FO-CTLK formula* φ

If

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

then

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

Can we apply this result to finite abstraction?

Abstraction

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

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

- Let $\mathcal{P} = \langle Ag, s_0, \rightarrow \rangle$ be an AC-MAS.
  The *abstraction* $\mathcal{P}' = \langle Ag', s'_0, \rightarrow' \rangle$ of $\mathcal{P}$ 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 $s \xrightarrow{\alpha(\vec{u})} t$, and $s \oplus t \simeq s' \oplus t'$ for some witness ι, then $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$ for some constant-preserving bijection ι' extending ι to $\vec{u}$.

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

Abstraction

- Let $N_{Ag} = \sum_{A_i \in Ag} \max_{\{\alpha(\vec{x}) \in Act_i\}} |\vec{x}|$ 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.* $|U'| \geq 2 \sup_{s \in \mathcal{P}} |adom(s)| + |C| + N_{Ag}$

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

Theorem

Consider

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

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

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

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

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

In particular,

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

$\Rightarrow$ 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 $\mathcal{A}$ is bounded by $b = |Items|(2|Ag| - 1) + 2$
- Consider a finite $U' \geq 2b + vars(\phi)$
- Abstract agents <u>A</u>uctioneer A' and <u>B</u>idders B'_i
  - the local db schemas $\mathcal{D}'_A$ and $\mathcal{D}'_i$ are the same as for A and B_i
  - the sets of actions Act'_A and 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 $<$ $true\_value$ for bidder B_i, and bid is an abstract value that does not represent any bid in D', and for $item$, $D'(status) = active$.

Finite Abstract Auction II

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 $\mathcal{D}$
- $\rightarrow'$ mimics $\rightarrow$
  - e.g., if $\alpha_i = bid_i(item, bid)$, then $s \xrightarrow{\alpha(\vec{u})}{}' \, t$ 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

$$\mathcal{A} \models \varphi \quad \text{iff} \quad \mathcal{A}' \models \varphi$$

Extensions

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

$$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ \ TValue_i(it, tv))$$

Extensions

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

$$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ \ TValue_i(it, tv))$$

2. Non-uniform and unbounded AC-MAS: one-way preservation result for FO-ACTLK$^-$.

Theorem

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

$$\mathcal{P}' \models \varphi \quad \Rightarrow \quad \mathcal{P} \models \varphi$$

Extensions

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

$$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ \ TValue_i(it, tv))$$

2. Non-uniform and unbounded AC-MAS: one-way preservation result for FO-ACTLK$^-$.

Theorem

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

$$\mathcal{P}' \models \varphi \quad \Rightarrow \quad \mathcal{P} \models \varphi$$

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.

$$AG\ \forall it, \vec{bd}, s(\exists! bp\ Bidding(it, \vec{bd}, bp, s) \wedge \exists^{\leq 1} tv\ \ TValue_i(it, tv))$$

2. Non-uniform and unbounded AC-MAS: one-way preservation result for FO-ACTLK$^-$.

Theorem

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

$$\mathcal{P}' \models \varphi \quad \Rightarrow \quad \mathcal{P} \models \varphi$$

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 perfom the boundedness check.

Merci!

References

eamericonarticle Christel Baier and Joost-Pieter Katoen.
Principles of Model Checking.
MIT Press, 2008.

eamericonarticle D. Cohn and R. Hull.
Business Artifacts: A Data-Centric Approach to Modeling Business Operations and Processes.
IEEE Data Eng. Bull., 32(3):3–9, 2009.

eamericonarticle R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi.
Reasoning About Knowledge.
The MIT Press, 1995.

eamericonarticle B. Bagheri Hariri, D. Calvanese, G. De Giacomo, R. De Masellis, and P. Felli.
Foundations of Relational Artifacts Verification.
In *Proc. of BPM*, 2011.