Model Checking Auctions as Artifact Systems: Decidability via Finite Abstraction

Francesco Belardinelli [1]

Abstract. The formal verification of auctions has recently received considerable attention in the AI and logic community. We tackle this problem by adopting methodologies and techniques originally developed for Artifact Systems, a novel paradigm in Service Oriented Computing. Specifically, we introduce a typed version of artifact-centric multi-agent systems (AC-MAS), a multi-agent setting for Artifact Systems, and consider the model checking problem against typed first-order temporal epistemic specifications. Notably, this formal framework is expressive enough to capture a relevant class of auctions: parallel English (ascending bid) auctions. We prove decidability of the model checking problem for AC-MAS via finite abstraction. In particular, we put forward a methodology to formally verify interesting properties of auctions.

1 Introduction

The formal verification of game structures is a topic of growing interest in the AI and logic community [2, 18, 29]. Particularly, the model checking problem for auctions has received considerable attention recently [3, 20, 31, 32]. Indeed, it is hard to overestimate the relevance of auctions and auction-based mechanisms in a wide range of distributed systems. Task scheduling [27], power grid management [11], and resource allocation [21] are all areas where auctions have found successful applications. However, with some notable exceptions, most of the research on this topic has focus on the design of auctioning mechanisms and the analysis of their formal properties, while the automated verification of these designs has only partially been addressed [3, 20, 32, 30].

In this paper we tackle the issues pertaining to model checking auctions by adopting methodologies and techniques originally developed for Artifact Systems, a novel paradigm for the specification and implementation of business processes [23, 24]. Artifact Systems (AS) are best described in terms of interacting modules, or *artifacts*, which typically consist of a *data model*, accounting for the relational structure of data, and a *lifecycle*, describing the evolution of the system over time. In order to develop secure and reliable business processes, automated verification procedures by model checking have been investigated in relation to Artifact Systems; thus producing several results on the formal verification of data-aware systems [9, 14, 23]. However, to keep the verification task tractable, most contributions disregard the data content of artifacts as well as the agents implementing the services. Still, in Artifact Systems and auctions alike it is crucial to reason about the actions agents can perform, the knowledge they possess, as well as the states they can jointly reach. Hence, the formal verification of both AS and auctions can benefit from techniques developed in the area of *reasoning about knowledge*. Indeed, knowledge representation and reasoning analyses formally the epistemic properties of rational and proactive actors, or *agents*. This line of research has generated a considerable body of work [17], including the verification of complex temporal epistemic specifications [25, 26].

Taking inspiration from the works above, this paper aims at providing a twofold contribution. Firstly, we put forward an agent-based abstraction techniques to model check Artifact Systems. Secondly, we apply this methodology to the formal verification of auctions. This endeavour requires a significant effort and generates interesting theoretical results. Indeed, the presence of data leads to a potentially infinite state space, thus making the verification problem undecidable in the most general setting. In this paper we focus on parallel English auctions and model these as artifact-centric multi-agent systems (AC-MAS) [6, 7], a multi-agent setting for Artifact Systems. Then, we tackle the model checking problem against specifications written in a first-order temporal epistemic logic suitable to describe the agents' information state during the auction bidding process. Notably, the specification language includes predicates whose interpretation might be infinite (for example, total orders on rational numbers). This modelling choice, while allowing to express background information shared by agents, calls for novel abstraction techniques with respect to the state-of-the-art. Specifically, the notion of *uniformity*, which has proved to be sufficient for finite abstractions [6, 13], has to be recast to account for this more complex setting. We then describe an abstraction techniques for AC-MAS, and prove that a specification is satisfied by a concrete, infinite-state AC-MAS iff it is satisfied by its finite abstraction. In particular, this result applies to parallel English auctions.

Related Work. To our knowledge [3, 20, 31, 32, 30] are among the first contributions to consider the formal verification of auctions. In [30] the authors implement a simple auction model in a BDI-based programming language, to which they apply agent verification techniques. In [20] the problem of model checking strategy-proofness of Vickrey auctions is investigated; while [31, 32] propose a formal approach to check for shilling behaviours in auctions. Overall, [3] is the contribution most closely related to the present work in spirit, as the authors also analyze the verification of agent-based English auctions, but a key difference is that their models abstract from the data content of auctions. We also remark that the references above discuss limited classes of auctions, and the solutions proposed are tailored to the cases of interest. On the more general subject of Artifact Systems verification, in [12, 14] this problem is investigated in relation to first-order linear-time specifications; while [22] considers data-centric dynamic systems. In both cases the specification language is synctactically restricted, while no such restriction is here considered. Also, differently from the above, we adopt an agent-based perspective, as

[1] Laboratoire IBISC, Université d'Evry, France, email: belardinelli@ibisc.fr

reflected in the modular abstraction methodology. Other works considering these features are [5, 6, 7], upon which this paper builds. However, the task of formally verifying parallel English auctions, in particular the need for infinite total orders, calls for novel abstraction techniques with respect to the cited references.

Scheme of the Paper. In Section 2 we present parallel English auctions and AC-MAS, a framework for Artifact Systems in a multi-agent setting. Also, we introduce the typed first-order temporal epistemic logic tFO-CTLK and state the model checking problem. In Section 3 we show that AC-MAS are expressive enough to model parallel English auctions. Sections 4 and 5 contain the main theoretical results of the paper: in Section 4 we define a notion of bisimulation for AC-MAS and in Section 5 we state sufficient conditions for the model checking problem to be decidable via finite abstraction. The technique is then applied to the formal verification of parallel English auctions. We conclude by discussing the proposed approach and point to future work.

2 Preliminaries

In this section we introduce the notation and formal notions that will be used in the rest of the paper.

2.1 Auctions

Hereafter we focus on a particular form of auction: the parallel English (ascending bid) auction [16]. This kind of auction is of particular interest in the present context, as it is common to a number of distributed scenarios, including popular auctioning websites. In parallel English auctions we typically have a single auctioneer A and a finite number of bidders $B_1, \ldots, B_\ell$. The auctioneer puts on sale a finite number of items, starting from a *base price* that is public to all bidders. For sake of presentation, we consider the bidding process as structured in discrete rounds. At each round, the bidder can either choose to bid for a specific item or to skip the round. At the end of the bidding process, each item is assigned to the bidder with the highest offer. We assume that our bidders are rational and each of them has an intrinsic value for each item being auctioned: she is willing to buy the item for a price up to her *true value*, but not for any higher price. Also, each bidder keeps this information private from other bidders and the auctioneer.

We are interested in verifying auctions against properties concerning the evolution of the bidding process and the knowledge acquired by bidders. For instance, we might want to check that *(i)* the base price for each item is indeed known to all agents, and not only this but that the base price is actually common knowledge. Also, we might want to express that *(ii)* the true value of each bidders for each item is indeed unknown to the actioneer and the other bidders, and it remains so throughout the bidding process. Other specifications of interest might be liveness properties such as *(iii)* the bidders are always able to make a higher bid, unless they have already hit their true value.

We remark that model checking such properties is extremely complex, given the distributed nature of auctions. Indeed, bidders can draw their bids from a set of values that is infinite in general. Also, the auctioneer can in principle choose any base price for the auctioned items. Since prices are usually represented by real or rational numbers, we obtain that auctions typically belong to the realm of infinite-state systems.

In what follows we provide a formal model for auctions and show that we can model check properties such as *(i)-(iii)* above by considering finite abstractions of concrete infinite-state auctions.

2.2 Artifact-centric Multi-agent Systems

We now fix the basic notation for databases used hereafter [1]. In what follows we assume a finite number of types $T_1, \ldots, T_k$.

Definition 1 (Db schema and instance) *A* (typed) 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 *with arity* $c \in \mathbb{N}$ *and type* $T_{k_1}, \ldots, T_{k_c}$.

Given a countable interpretation domain U_h *for each type* T_h*, a* $\mathcal{D}$-instance *over* $U_1, \ldots, U_k$ *is a mapping* D *associating in a type-consistent way* (i) *each relation symbol* $P \in \mathcal{D}$ *with a* finite a*-ary relation* $D(P)$ *over* $U_{k_1} \times \ldots \times U_{k_a}$*, and* (ii) *each relation symbol* $Q \in \mathcal{D}$ *with a (*possibly infinite*) b-ary relation* $D(Q)$ *over* $U_{k_1} \times \ldots \times U_{k_b}$.

In Def. 1 we depart from the standard notion of db instance as, while the intepretation $D(P)$ of a symbol $P \in \mathcal{D}$ is *finite*, the interpretation $D(Q)$ of a symbol $Q \in \mathcal{D}$ can be *infinite* in principle. Intuitively, the symbols Q are used to model background information on the interpretation domains, such as the total order $<$ on the set $\mathbb{Q}$ of rational numbers.

The set of all $\mathcal{D}$-instances over $U_1, \ldots, U_k$ is denoted as $\mathcal{D}(\vec{U})$. Also, the interpretation domains are normally assumed to be disjoint and the db schema $\mathcal{D}$ is omitted whenever clear by the context. The *active domain* $adom(D) = \langle adom_1(D), \ldots, adom_k(D)\rangle$ of a db instance D is a tuple where each $adom_h(D)$ is the set of all individuals in U_h occurring in some relation $D(P)$. Since $\mathcal{D}$ and each $D(P)$ are finite, so is each $adom_h(D)$. Notice that the relations $D(Q)$ do not contribute to the definition of the active domain. Finally, with an abuse of notation we write $f : U_h \to U'_h$ to express that f is a function s.t. for each type T_h, $f(u) \in U'_h$ if $u \in U_h$.

We now introduce the disjoint union $\oplus$ of db instances. Let the *primed version* of the db schema $\mathcal{D}$ above be the db schema $\mathcal{D}' = \{P'_1/a_1, \ldots, P'_n/a_n, Q'_1/b_1, \ldots, Q'_m/b_m\}$.

Definition 2 (Disjoint union $\oplus$) *Given* $\mathcal{D}$*-instances* D *and* D'*,* $D \oplus D'$ *is the* $(\mathcal{D} \cup \mathcal{D}')$*-instance s.t. for every relation symbol* R*,* $D \oplus D'(R) = D(R)$ *and* $D \oplus D'(R') = D'(R)$.

We now introduce a notion of *agent* inspired to multi-agent systems [6, 17].

Definition 3 (Agent) *Given a countable interpretation domain* U_h *for each type* T_h*, an* agent *is a tuple* $A = \langle \mathcal{D}, Act, Pr\rangle$ *such that*

- $\mathcal{D}$ *is the* local database schema*;*
- Act *is a finite set of* (typed) actions $\alpha(\vec{T})$*, where the tuple* $\vec{T}$ *of types are the* formal parameters *of* α*;*
- $Pr : \mathcal{D}(\vec{U}) \mapsto 2^{Act(\vec{U})}$ *is the* local protocol function*, where* $Act(\vec{U})$ *is the set of* ground actions $\alpha(\vec{u})$*, for* $\alpha(\vec{T}) \in Act$ *and* $\vec{u} \in \vec{U}^{|\vec{T}|}$ *a tuple of* (type-consistent) ground parameters.

As standard in multi-agent systems (MAS), each agent A performs the actions in Act according to the protocol function Pr. Moreover, we assume that A is in some local state $D \in \mathcal{D}(\vec{U})$, that is, the information she possesses is structured as a database.

As agents can interact among themselves, we consider their composition.

Definition 4 (AC-MAS) *Given a countable interpretation domain* U_h *for each type* T_h *and a set* $Ag = \{A_0, \ldots, A_\ell\}$ *of agents* $A_i = \langle \mathcal{D}_i, Act_i, Pr_i\rangle$*, an* Artifact-centric Multi-agent System *is a tuple* $\mathcal{P} = \langle Ag, s_0, \tau\rangle$ *such that*

- $s_0 \in \mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U})$ *is the* initial global state*;*
- $\tau : \mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U}) \times Act(\vec{U}) \mapsto 2^{\mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U})}$ *is the* global transition function*, where* $Act(\vec{U}) = Act_0(\vec{U}) \times \ldots \times Act_\ell(\vec{U})$ *is the set of* joint (ground) actions*, and the transition* $\tau(s, \langle \alpha_0(\vec{u}_0), \ldots, \alpha_\ell(\vec{u}_\ell) \rangle)$ *is defined iff* $\alpha_i(\vec{u}_i) \in Pr_i(D_i)$ *for all* $i \leq \ell$*.*

The assumption of a single interpretation domain for each type does not limit the generality of the approach, as agents' domains can always be extended before composition. Also, AC-MAS are rich enough to formalize the framework of Artifact Systems, as it was shown in [5, 7] for instance.

We now introduce some basic terminology. We denote a joint (ground) action as $\alpha(\vec{u})$ for $\alpha = \langle \alpha_0(\vec{T}_0), \ldots, \alpha_\ell(\vec{T}_\ell) \rangle$ and $\vec{u} = \langle \vec{u}_0, \ldots, \vec{u}_\ell \rangle$, and define the *transition relation* $\rightarrow$ on global states so that $s \rightarrow s'$ iff $s \xrightarrow{\alpha(\vec{u})} t$, i.e., $s' \in \tau(s, \alpha(\vec{u}))$ for some $\alpha(\vec{u}) \in Act(\vec{U})$. An *s-run* r is an infinite sequence $s^0 \rightarrow s^1 \rightarrow \cdots$, with $s^0 = s$. For $n \in \mathbb{N}$, we set $r(n) = s^n$. A state s' is *reachable from* s if there is an s-run r s.t. $r(i) = s'$ for some $i \geq 0$. We can safely assume that the relation $\rightarrow$ is serial by considering for each agent A_i a $skip_i$ action enabled at every local state. Finally, we introduce $\mathcal{S} \subseteq \mathcal{D}_0(\vec{U}) \times \ldots \times \mathcal{D}_\ell(\vec{U})$ as the set of global states reachable from the initial state s_0. The following class of AC-MAS will feature prominently in the paper.

Definition 5 (Rigidity) *An AC-MAS* $\mathcal{P}$ *is* rigid *iff for every* $Q \in \mathcal{D}$*, and* $s, s' \in \mathcal{S}$*,* $D_i \in s$*,* $D_j \in s'$*,* $D_j(Q) = D_i(Q)$*.*

In rigid AC-MAS the symbols $Q \in \mathcal{D}$ have the same interpretation in all global states and for all agents, consistently with the intuition that these represent persistent properties of the interpretation domains known to all agents. We refer to this relation as $\mathcal{P}(Q)$. Further, two global states $s = \langle D_0, \ldots, D_\ell \rangle$ and $s' = \langle D'_0, \ldots, D'_\ell \rangle$ in $\mathcal{S}$ are *indistinguishable* for agent A_i, or $s \sim_i s'$, if $D_i = D'_i$ [17]. Finally, for technical reasons we refer to the *global* db schema $\mathcal{D} = \bigcup_{A_i \in Ag} \mathcal{D}_i$ of an AC-MAS. Then, each state s is associated with the $\mathcal{D}$-instance $D_s \in \mathcal{D}(\vec{U})$ s.t. $D_s(R) = \bigcup_{A_i \in Ag} D_i(R)$. Also, we write $adom(s)$ for $adom(D_s)$. Notice that for every state s, the associated D_s is unique, whereas the converse is not true in general. Furthermore, we lift the disjoint union operator $\oplus$ to global states so that $s \oplus s'$ is defined as $\langle D_0 \oplus D'_0, \ldots, D_\ell \oplus D'_\ell \rangle$.

2.3 The Typed Logic tFO-CTLK

We now consider the specification language for AC-MAS. For each type T_h let Var_h (resp. Con_h) be a countable set of *(typed) individual variables* (resp. *constants*). A *(typed) term* is then any element $t \in Var_h \cup Con_h$.

Definition 6 (tFO-CTLK) *The typed first-order CTLK formulas* φ *over a db schema* $\mathcal{D}$ *are defined by the following BNF:*

$$\varphi ::= t = t' \mid R(\vec{t}) \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \forall x\varphi \mid AX\varphi \mid A\varphi U\varphi \mid E\varphi U\varphi \mid K_i\varphi \mid C\varphi$$

where $R \in \mathcal{D}$*,* $\vec{t}$ *is a type-consistent tuple of terms,* t, t' *are terms of the same type,* $x \in Var_h$*, and* $i \leq \ell$*.*

We introduce the abbreviations $\exists$, $\wedge$, $\vee$, $\neq$, and define free and bound variables as standard. For a formula φ, $var_h(\varphi)$ (resp. $fr_h(\varphi)$ and $con_h(\varphi)$) denotes the set of its variables (resp. free variables and constants) of type T_h. A *sentence* is a formula with no free variables. The temporal formulas $AX\varphi$ and $A\varphi U\varphi'$ (resp. $E\varphi U\varphi'$) are read as "for all runs, at the next step φ" and "for all runs (resp. some run), φ until φ'". The epistemic formulas $K_i\varphi$ and $C\varphi$ intuitively mean that "agent A_i knows φ" and "it is common knowledge that φ" respectively. We use the standard abbreviations $EX\varphi$, $AF\varphi$, $AG\varphi$, $EF\varphi$, and $EG\varphi$. By Def. 6 free variables can occur within the scope of modal operators; this is a major feature of the present framework in comparison with, for instance, [9, 19]. Hereafter we consider also the non-modal fragment of tFO-CTLK, i.e., the typed first-order logic tFO defined by the following BNF:

$$\varphi ::= t = t' \mid R(\vec{t}) \mid \neg\varphi \mid \varphi \rightarrow \varphi \mid \forall x\varphi$$

We now assign a meaning to tFO-CTLK formulas by using AC-MAS. Given countable interpretation domains U_h s.t. $Con_h \subseteq U_h$, a *(type-consistent) assignment* is a function $\sigma : Var_h \mapsto U_h$. Also, we denote by σ^x_u the assignment s.t. *(i)* $\sigma^x_u(x) = u \in U_h$; and *(ii)* $\sigma^x_u(x') = \sigma(x')$ for every $x' \in Var_h$ different from x. For convenience, we extend assignments to constants so that $\sigma(t) = t$ whenever $t \in Con_h$, The semantics of tFO-CTLK formulas is then defined as follows.

Definition 7 (Satisfaction) *We define whether an AC-MAS* $\mathcal{P}$ satisfies *a tFO-CTLK formula* φ *in a state* $s \in \mathcal{S}$ *for assignment* σ*, or* $(\mathcal{P}, s, \sigma) \models \varphi$*, as follows (the clauses for propositional connectives are straightforward and thus omitted):*

$(\mathcal{P}, s, \sigma) \models R(\vec{t})$ *iff* $\langle \sigma(t_1), \ldots, \sigma(t_c) \rangle \in D_s(R)$
$(\mathcal{P}, s, \sigma) \models t = t'$ *iff* $\sigma(t) = \sigma(t')$
$(\mathcal{P}, s, \sigma) \models \forall x\varphi$ *iff for all* $u \in adom_h(s)$*,* $(\mathcal{P}, s, \sigma^x_u) \models \varphi$
$(\mathcal{P}, s, \sigma) \models AX\varphi$ *iff for all runs* r*, if* $r(0) = s$ *then* $(\mathcal{P}, r(1), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models A\varphi U\varphi'$ *iff for all runs* r*, if* $r(0) = s$ *then there is* $k \geq 0$ *s.t.* $(\mathcal{P}, r(k), \sigma) \models \varphi'$*, and for all* j*,* $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models E\varphi U\varphi'$ *iff for some run* r*,* $r(0) = s$ *and there is* $k \geq 0$ *s.t.* $(\mathcal{P}, r(k), \sigma) \models \varphi'$*, and for all* j*,* $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models K_i\varphi$ *iff for all* s'*,* $s \sim_i s'$ *implies* $(\mathcal{P}, s', \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models C\varphi$ *iff for all* s'*,* $s \sim s'$ *implies* $(\mathcal{P}, s', \sigma) \models \varphi$

where $\sim$ *is the transitive closure of* $\bigcup_{A_i \in Ag} \sim_i$*.*

Notice that the interpretation of temporal operators is standard of the braching-time logic CTL; while epistemic modalities are interpreted as in the multi-modal logic $S5_{|Ag|}$. Further, a formula φ is *true* in s, or $(\mathcal{P}, s) \models \varphi$, if $(\mathcal{P}, s, \sigma) \models \varphi$ for all σ; while φ is *true* in $\mathcal{P}$, or $\mathcal{P} \models \varphi$, if $(\mathcal{P}, s_0) \models \varphi$.

We adopt an *active-domain* semantics, that is, in each state s quantified variables range only over the active domain of s, which is finite. Nonetheless, by the unconstrained alternation of free variables and modal operators, we can refer to these "active" individuals in successive states, where they might no longer be active. In Section 3 we argue that this form of quantification is sufficient for specifying interesting properties of auctions.

The key concern of this paper is to investigate the model checking problem for AC-MAS against tFO-CTLK specifications defined as follows.

Definition 8 (Model Checking Problem) *Model checking an AC-MAS* $\mathcal{P}$ *against a tFO-CTLK formula* φ *amounts to finding an assignment* σ_0 *such that* $(\mathcal{P}, s_0, \sigma_0) \models \varphi$*.*

If all U_h are finite, the model checking problem is decidable, as $\mathcal{P}$ is a finite-state system. However, this is not the case in general, as the following result related to Theorem 4.10 in [15] shows.

Theorem 1 *The model checking problem for AC-MAS w.r.t. tFO-CTLK is undecidable.*

In Section 4 and 5 we develop an abstraction technique to tackle this issue. But first we introduce an auction scenario to illustrate the formal machinery.

3 Auctions as AC-MAS

In this section we apply the formal framework of AC-MAS developed in Section 2.2 to model the parallel English auctions in Section 2.1. In particular, bids can be thought of as artifacts exchanged between the bidders and the auctioneer, and agents' actions depend on the data content of these artifacts. The relatively small size of the data model in auction AC-MAS will allow us to outline in Section 5 the verification procedure for tFO-CTLK specifications via finite abstraction. As detailed above, hereafter we consider a single auctioneer A and a finite number of bidders $B_1, \ldots, B_\ell$. The domains of interpretation include a finite set $Items$ of items, as well as the set $\mathbb{Q}$ of rational numbers to represent values for base prices, true values and bids. For sake of presentation we use the same names to denote interpretation domains and types. We start by formally defining the auctioneer as an agent according to Def. 3.

Definition 9 (Auctioneer) *The auctioneer $A = \langle \mathcal{D}_A, Act_A, Pr_A \rangle$ is defined as*

- $\mathcal{D}_A = \{Base/2, \{Bid_i/2\}_{i \leq \ell}, Status/2, </2\}$ *where $Base(it, bp)$ represents the base price $bp \in \mathbb{Q}$ for item $it \in Items$, each $Bid_i(it, bd)$ represents the bid $bd \in \mathbb{Q}$ of bidder B_i for item it, $Status(it, st)$ keeps track of the status of items; status st has two possible values: active if item it is actively traded, or term if the bidding phase for item it has terminated. Finally, $<$ is the standard "strictly less" symbol on $\mathbb{Q}$.*
- $Act_A = \{init_A(it, bp), time\_out(it), skip_A\}$.
- $init_A(it, bp) \in Pr_A(D)$ *if item it does not appear in any tuple in $D(Status)$; $time\_out(it) \in Pr_A(D)$ if $(it, \textsf{active}) \in D(Status)$; while the action $skip_A$ is always enabled.*

Intuitively, the auctioneer non-deterministically chooses to put some item it up for auctioning by performing action $init_A(it, bp)$. The base price bp is then registered in $Base$. She keeps track of bidder B_i's offers in Bid_i and non-deterministically stops the bidding phase for a specific item it by action $time\_out(it)$. At that point, the item is withdrawn and can no longer be put on sale.

Further, each bidder B_i can be represented as the following agent.

Definition 10 (Bidder) *Each bidder $B_i = \langle \mathcal{D}_i, Act_i, Pr_i \rangle$ is defined as*

- $\mathcal{D}_i = \{TValue_i/2, Base/2, \{Bid_i/2\}_{i \leq \ell}, Status/2, </2\}$ *where $TValue_i(it, tv)$ represents the true value $tv \in \mathbb{Q}$ of item it for bidder B_i, while $Base$, Bid_i, $Status$ and $<$ are defined as for the auctioneer.*
- $Act_i = \{init_i(it, tv), bid_i(it, bd), skip_i\}$.
- $init_i(it, tv) \in Pr_i(D)$ *if $(it, \textsf{active}) \in D(Status)$ and item it does not appear in $D(TValue_i)$; $bid_i(it, bd) \in Pr_i(D)$ whenever the item it appears in $D(TValue_i)$, the highest bid bd_j in some Bid_j ($j \neq i$) for item it is strictly less than the true value tv for bidder B_i, $(it, \textsf{active}) \in D(Status)$, and $bd_j < bd \leq tv$. The action $skip_i$ is always enabled.*

By Def. 10 it is apparent that each bidder can bid only for actively traded items, whenever bids have not exceeded her true value. After that point, she stops bidding. Notice that symbols $Base$, Bid_i, $Status$ and $<$ are shared by all agents. However, each relation can be modified by at most one agent ($Base$ and $Status$ by the auctioneer; Bid_i by bidder B_i). Hence, the consistency of db instances is preserved. Also, the information contained in $TValue_i$ is private to each agent B_i.

We can now introduce the formal definition of an auction as an AC-MAS.

Definition 11 (Auction AC-MAS) *Given the set $Ag = \{A, B_1, \ldots, B_\ell\}$ of agents on sets Items, $\mathbb{Q}$, and $\{\textsf{active}, \textsf{terms}\}$, the* auction AC-MAS *is a tuple $\mathcal{A} = \langle Ag, s_0, \tau \rangle$ where*

- $s_0 = \langle D_A, D_1, \ldots, D_\ell \rangle$ *is the global state where for all $j \in \{A, 1, \ldots, \ell\}$, $D_j(<)$ is the "strictly less" relation on $\mathbb{Q}$, while all other relations are empty;*
- *τ is the global transition function s.t. $s \xrightarrow{\alpha(\vec{u})} s'$ iff*
  - *$\alpha_A = init_A(it, bp)$ and s' modifies s by adding tuples (it, bp) and $(it, \textsf{active})$ to relations $D'_A(Base)$ and $D'_A(Status)$ respectively;*
  - *$\alpha_i = init_i(it, tv)$ and s' modifies s by adding tuple (it, tv) to relation $D'_i(TValue_i)$ for bidder B_i;*
  - *$\alpha_i = bid_i(it, bd')$ and s' modifies s by replacing any tuple (it, bd) in $D_j(Bid_i)$ with (it, bd');*
  - *$\alpha_A = time\_out(it)$ and $(it, \textsf{active}) \notin D'_j(Status)$ and $(it, \textsf{term}) \in D'_j(Status)$;*
  - *$\alpha_A = skip_A$ or $\alpha_i = skip_i$ for some $i \leq \ell$, and $D'_i = D_i$.*

Notice that the auction AC-MAS $\mathcal{A}$ in Def. 11 respects the intuitions on the progress of an auction for multiple items in parallel. Items are put on sale by the auctioneer and bidders can offer up to their true value tv. Since bidders can bid any value in $\mathbb{Q}$ (up to tv), there can be an infinite number of bids in principle, so the AC-MAS $\mathcal{A}$ is really an infinite-state system. Of course, in our presentation we made a number of conceptual abstractions. For instance, agents are assumed to be perfectly rational and perfect reasoners, as such they drop from the auction as soon as they hit their true value. Also, bids can be incremented by any small amount. While, an actual auction might not allow for some of these behaviours, we maintain that the present formalisation satisfies an idealised notion of parallel English auction as it has been successfully analysed in game theory and rational choice theory [16]. Finally, notice that the interpretation of symbol $<$ is rigid as it represents the "strictly less" relation on $\mathbb{Q}$ in the initial state s_0, and this interpretation is not modified by τ. Since $<$ is the only symbol with an infinite interpretation graph, the auction AC-MAS $\mathcal{A}$ is *rigid*.

While $\mathcal{A}$ intuitively fulfils the informal description of an auction, we need to develop formal verification techniques to check this fact. Hence, we turn to considering properties of interest that can be expressed in the specification language tFO-CTLK. First, one feature of the auction we can check is that for each item $it \in Items$ there is exactly one base price bp registered in the relation $Base$, while bidders associate *at most* one true value tv to each item it (possibly none). This can be expressed as

$$AG\, \forall it (\exists! bp\ Base(it, bs) \wedge \exists^{\leq 1} tv\ TValue_i(it, tv))$$

where the quantifiers $\exists!$ and $\exists^{\leq 1}$ are defined as standard in first-order logic with identity.

In tFO-CTLK we can also express what agents know or ignore about the information content of the auction. For instance, specification *(i)* in Section 2.1 requires that the base prices of items remain

common knowledge throughout the auction:

$$AG\ \forall it\ \exists bp\ C\ Base(it, bp)$$

On the contrary, according to *(ii)* the true value of items for each bidder B_i is secret to all other bidders and to the auctioneer:

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

Further, we can express properties on the progress of the auctioning process. As an example, for each bidder B_i, each bid is less or equal to her true value:

$$AG\ \forall it, bd, tv((Bid_i(it, bd) \wedge TValue_i(it, tv)) \rightarrow bd \leq tv)$$

Also, specification *(iii)* states that each bidder B_i can raise her bid unless she has already hit her true value:

$$AG\ \forall it, bd(Bid_i(it, bd) \rightarrow \\ \rightarrow (TValue_i(it, bd) \vee EF\ \exists bd'(bd' > bd \wedge Bid_i(it, bd'))))$$

Finally, in tFO-CTLK we can define when an agent has won the auction for a specific item, and reason about the knowledge the other agents have of this fact. These features can be stated as *(i)* at the beginning it is common knowledge that there will be a winner eventually, and *(ii)* the identity of the winner will eventually be common knowledge. We first introduce the formula $Win_i(it)$, which intuitively means that the bidder B_i wins the auction for item it, as follows:

$$\begin{aligned} Win_i(it) = Status(it, \mathsf{term}) \\ \wedge \exists bd(Bid_i(it, bd) \\ \wedge \textstyle\bigwedge_{j\neq i} \forall bd'(Bid_j(it, bd') \rightarrow bd' < bd)) \end{aligned}$$

Then, statements *(i)* and *(ii)* can be respectively formalised as

$$CAG\ \forall it AF \bigvee_{B_i\in Ag} Win_i(it) \qquad AG\ \forall it AF \bigvee_{B_i\in Ag} C\ Win_i(it)$$

Hence, the interplay of quantifiers and epistemic and temporal modalities allows to express precisely subtly different concepts such as *(i)* and *(ii)* above.

We observe that in all specifications above the quantifiers bind variables of type $\mathbb{Q}$ that appear also as arguments of some non-rigid symbol ($Base$, Bid_i and $TValue_i$). Thus, these terms are meant to receive values taken from the active domain of each state. This remark motivates our choice of restricting quantification to the active domain in first place. Indeed, in the specifications above we are not interested in expressing general properties of the order $<$ on $\mathbb{Q}$ by comparing generic rational numbers, rather we want to confront values for bids, base prices and true values for items. Thus, tFO-CTLK suffices to express properties of auctions.

In the next sections we develop the theory that will allow us to model check specifications as above on a particular class of artifact-centric multi-agent systems that includes the auction AC-MAS $\mathcal{A}$.

4 Bisimulation

In this section we introduce a notion of bisimulation for AC-MAS. Similar notions have already appeared in the literature [6, 7]. However, in this paper we consider typed languages and, most importantly, relations $Q \in \mathcal{D}$ with a possibly infinite interpretation. This extended framework has an impact notably on the key concept of *uniformity*, which allows us to prove that bisimilar AC-MAS satisfy the same tFO-CTLK specifications. Intuitively, the behaviour of uniform AC-MAS does not depend on data that are not explicitly named in the systems description. In the rest of the section we let $\mathcal{P} = \langle Ag, s_0, \tau\rangle$ and $\mathcal{P}' = \langle Ag', s_0', \tau'\rangle$ be AC-MAS and assume that $s = \langle D_0, \ldots, D_\ell\rangle \in \mathcal{S}$ and $s' = \langle D_0', \ldots, D_\ell'\rangle \in \mathcal{S}'$. Also, each $\mathcal{C}_h$ is a finite set of constants of type T_h. We start by introducing a notion of isomorphism on db instances and global states that accounts also for symbols $Q \in \mathcal{D}$.

Definition 12 (Isomorphism) *The db instances $D, D' \in \mathcal{D}(\vec{U})$ are* isomorphic*, or $D \simeq D'$, iff there is a type-consistent bijection ι : $adom_h(D) \cup \mathcal{C}_h \mapsto adom_h(D') \cup \mathcal{C}_h$ such that*

(i) *ι is the identity on each $\mathcal{C}_h$;*
(ii) *for every $R \in \mathcal{D}$ and $\vec{u} \in U_{k_1} \times \ldots \times U_{k_c}$, $\vec{u} \in D(R)$ iff $\iota(\vec{u}) \in D'(R)$.*

When this is the case, we say that ι is a witness *for $D \simeq D'$.*

The global states s and s' are isomorphic*, or $s \simeq s'$, iff there exists a type-consistent bijection ι : $adom_h(s) \cup \mathcal{C}_h \mapsto adom_h(s') \cup \mathcal{C}_h$ s.t. for every $A_i \in Ag$, ι is a witness for $D_i \simeq D_i'$. Any function ι as above is a* witness *for $s \simeq s'$.*

Isomorphisms preserve the interpretation of individual constants as well as of relation symbols $P \in \mathcal{D}$. As to symbols $Q \in \mathcal{D}$, the witness ι preserves the interpretation only for the individuals in the active domain. This feature of isomorphisms is key to obtain finite abstractions. Clearly, $\simeq$ is an equivalence relation. Also, given a function $f : U_h \mapsto U_h'$ defined on each $adom_h(s)$, $f(s)$ denotes the instance in $\mathcal{D}(\vec{U'})$ obtained from s by renaming each $u \in Dom(f)$ as $f(u)$. If f is also injective on $adom_h(s)$ (thus invertible) and the identity on each $\mathcal{C}_h$, then $f(s) \simeq s$.

Example 1 As an example of isomorphic db instances, consider the auctioneer A in Section 3 with local db schema $\mathcal{D}_A = \{Base/2, Bid_1/2, Bid_2/2, Status/2, </2\}$ and the interpretation domain $Items = \{000, 001, 010, 011, 100, 101, 110, 111\}$. Also, fix the sets $\mathcal{C}_{Items} = \{101\}$ and $\{\mathsf{active}, \mathsf{term}\}$ of constants. Further, let D_A and D_A' be the $\mathcal{D}_A$-instances as illustrated in Fig. 1. It can be readily seen that D_A and D_A' are isomorphic with witness ι such that $\iota(101) = 101$, $\iota(110) = 011$, $\iota(3.23) = 5.78$, $\iota(4.25) = 8.74$, $\iota(3.50) = 6.13$, and $\iota(6.73) = 10.09$. ■

$D_A(Base)$	
101	3.23
110	4.25

$D_A(Bid_1)$	
101	3.50

$D_A(Bid_2)$	
101	6.73

$D_A(Status)$	
101	active
110	term

$D_A'(Base)$	
101	5.78
011	8.74

$D_A'(Bid_1)$	
101	6.13

$D_A'(Bid_2)$	
101	10.09

$D_A'(Status)$	
101	active
011	term

Figure 1. Example of isomorphic db instances.

Observe that isomorphisms are such w.r.t. specific sets $\mathcal{C}_h$ of constants. Hereafter we assume that the various $\mathcal{C}_h$ are always fixed in advance. While isomorphic states share a common relational structure, they do not necessarily satisfy the same first-order formulas, as satisfaction depends also on values assigned to free variables. To account for this, we have to recast the notion of *equivalent assignments* in [6] to deal with types and symbols $Q \in \mathcal{D}$.

Definition 13 (Equivalent assignments) *Given isomorphic states s, s' and sets of variables $V_h \subseteq Var_h$ for each type T_h, the assignments $\sigma : Var_h \mapsto U_h$ and $\sigma' : Var_h \mapsto U'_h$ are* equivalent for all V_h w.r.t. s and s' *iff there exists a bijection $\gamma : adom_h(s) \cup \mathcal{C}_h \cup \sigma(V_h) \mapsto adom_h(s') \cup \mathcal{C}_h \cup \sigma'(V_h)$ such that*

(i) *the restriction $\gamma|_{adom_h(s)\cup\mathcal{C}_h}$ is a witness for $s \simeq s'$;*
(ii) $\sigma'|_{V_h} = \gamma \cdot \sigma|_{V_h}$;
(iii) *for every $\vec{u} \in (Dom(\gamma))^{k_b}$ and $A_i \in Ag$, $\vec{u} \in D_i(Q)$ iff $\gamma(\vec{u}) \in D'_i(Q)$.*

Intuitively, equivalent assignments preserve the (in)equalities of the variables in each V_h as well as the interpretation of symbols $Q \in \mathcal{D}$. Two assignments are said to be *equivalent for a tFO-CTLK formula* φ, also omitting the states s and s', if they are equivalent for all $fr_h(\varphi)$. We can now show that isomorphic states preserve the interpretation of typed first-order formulas.

Lemma 2 *Given isomorphic states s and s', a typed FO-formula φ with $con_h(\varphi) \subseteq \mathcal{C}_h$ for every type T_h, and assignments σ and σ' equivalent for φ, we have that*

$$(\mathcal{P}, s, \sigma) \models \varphi \quad \textit{iff} \quad (\mathcal{P}', s', \sigma') \models \varphi$$

Proof (sketch). By induction on the structure of φ. Consider the base case for an atomic formula $\varphi \equiv Q(t_1, \ldots, t_k)$. Then $(D_s, \sigma) \models \varphi$ iff $\langle \sigma(t_1), \ldots, \sigma(t_k) \rangle \in D_s(Q)$. Since σ and σ' are equivalent for φ, and $s \simeq s'$, this is the case iff $\langle \gamma(\sigma(t_1)), \ldots, \gamma(\sigma(t_k)) \rangle \in D_{s'}(Q)$, that is, $\langle \sigma'(t_1), \ldots, \sigma'(t_k) \rangle \in D_{s'}(Q)$. Hence, $(D_{s'}, \sigma') \models \varphi$. The base cases for $\varphi \equiv t = t'$ and $P(t_1, \ldots, t_k)$ are proved similarly, The inductive steps for propositional connectives and quantifiers are straightforward. □

An immediate consequence of this result is that isomorphic states satisfy the same first-order sentences. We aim at extending this preservation result to the full tFO-CTLK. In particular, plain bisimulations are known to preserve satisfaction in a propositional modal setting [10]. We now investigate the conditions under which this applies to AC-MAS as well, and begin by considering a notion of simulation. Throughout the rest of the paper we assume w.l.o.g. that $con_h(\varphi) \subseteq \mathcal{C}_h$ for every type T_h.

Definition 14 (Simulation) *A relation S on $\mathcal{S} \times \mathcal{S}'$ is a* simulation *if $\langle s, s' \rangle \in S$ implies:*

1. *$s \simeq s'$;*
2. *for $t \in \mathcal{S}$, if $s \to t$ then there is $t' \in \mathcal{S}'$ s.t. $s' \to t'$, $s \oplus t \simeq s' \oplus t'$ and $\langle t, t' \rangle \in S$;*
3. *for $A_i \in Ag$, $t \in \mathcal{S}$, if $s \sim_i t$ then there is $t' \in \mathcal{S}'$ s.t. $t \sim_i t'$, $s \oplus t \simeq s' \oplus t'$ and $\langle t, t' \rangle \in S$.*

Two states s and s' are *similar* iff $\langle s, s' \rangle \in S$ for some simulation S. Note that similar states are isomorphic by condition (1) above. Moreover, we impose that the disjoint union $s \oplus t$ is isomorphic to $s' \oplus t'$. Simulations can naturally be extended to bisimulations.

Definition 15 (Bisimulation) *A relation B on $\mathcal{S} \times \mathcal{S}'$ is a* bisimulation *iff both B and $B^{-1} = \{\langle s', s \rangle \mid \langle s, s' \rangle \in B\}$ are simulations.*

Two states s and s' are *bisimilar* iff $\langle s, s' \rangle \in B$ for some bisimulation B. Also, $\mathcal{P}$ and $\mathcal{P}'$ are *bisimilar*, or $\mathcal{P} \approx \mathcal{P}'$, iff so are their initial states s_0 and s'_0. By Lemma 2 it follows that bisimilar, hence isomorphic, states preserve typed FO-formulas. However, this is no longer the case when we consider the full tFO-CTLK language. We refer to [4] for an example of this fact. To overcome this difficulty we make more assumptions on our structures and introduce a novel notion of *uniformity*.

Definition 16 (Uniformity) *An AC-MAS $\mathcal{P}$ is* uniform *iff for every $s, t, s' \in \mathcal{S}$, $t' \in \mathcal{D}(\vec{U})$,*

1. *if $s \xrightarrow{\alpha(\vec{u})} t$ and $s \oplus t \simeq s' \oplus t'$ for some witness ι, then for every type-consistent constant-preserving extension ι' of ι to $\vec{u}$, we have that $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$;*
2. *if $s \sim_i t$ and $s \oplus t \simeq s' \oplus t'$, then $s' \sim_i t'$.*

Further, if $\mathcal{P}$ is rigid, then (i) *the set $\mathcal{D}(\vec{U})$ above is restricted to db instances t' agreeing on the interpretation $\mathcal{P}(Q)$ of symbols $Q \in \mathcal{D}$;* (ii) *for all $\vec{u}$, there exist $\vec{v}$, $\vec{v}'$ s.t. $(\vec{v}, \vec{u}) \in \mathcal{P}(Q)$ and $(\vec{u}, \vec{v}') \in \mathcal{P}(Q)$; and* (iii) *for all $\vec{u} \in \mathcal{P}(Q)$, for all $i < b-1$, there exist v s.t. $(u_0, \ldots, u_i, v, u_{i+1}, \ldots, u_{b-2}) \in \mathcal{P}(Q)$ and $(u_1, \ldots, u_i, v, u_{i+1}, \ldots, u_{b-1}) \in \mathcal{P}(Q)$ (with an abuse of notation we assume that for $u_{i+1} = u_{b-1}$ or $u_i = u_0$ the sequence ends or begins with v).*

Intuitively, conditions (1) and (2) in Def. 16 say that if state t is reached by executing the ground action $\alpha(\vec{u})$ in s, and v is uniformly replaced with v' in s, $\vec{u}$ and t, thus obtaining, say, s', $\vec{u}'$ and t', then t' can be reached by executing $\alpha(\vec{u}')$ in s'. This condition is akin to the notion of *genericity* in database theory [1]. Further, the condition on rigid AC-MAS is aimed at obtaining the same uniform transitions while keeping fixed the interpretation of symbols $Q \in \mathcal{D}$. In particular, we have the following result.

Proposition 17 *The auction AC-MAS $\mathcal{A}$ is indeed uniform.*

Proof (sketch). This follows by definition of the transition function τ and by the fact that the only relation interpreted rigidly in $\mathcal{A}$ is $<$ on $\mathbb{Q}$. In particular, condition *(ii)* on rigid and uniform AC-MAS is satisfied as the order $<$ on $\mathbb{Q}$ has no endpoints; while condition *(iii)* follows by the density of $<$ on $\mathbb{Q}$. □

As a consequence, the auction AC-MAS $\mathcal{A}$ is a rigid and uniform AC-MAS.

We now state the main contribution of this section, which lifts the result in [6] to AC-MAS with types and predicates with an infinite interpretation. Hereafter $\sup_{s\in\mathcal{S}}\{|adom_h(s)|\} = \infty$ whenever an AC-MAS $\mathcal{P}$ is unbounded, i.e., there is no $b \in \mathbb{N}$ s.t. $|adom_h(s)| \leq b$ for all $s \in S$.

Theorem 3 *Consider bisimilar and uniform AC-MAS $\mathcal{P}$ and $\mathcal{P}'$, and a tFO-CTLK formula φ. If for every T_h,*

1. *$|U'_h| \geq 2 \sup_{s\in\mathcal{S}}\{|adom_h(s)|\} + |\mathcal{C}_h| + |var_h(\varphi)|$*
2. *$|U_h| \geq 2 \sup_{s'\in\mathcal{S}'}\{|adom_h(s')|\} + |\mathcal{C}_h| + |var_h(\varphi)|$*

then

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

A proof of Theorem 3 is given in Appendix A. By this result if each $\{|adom_h(s)| \mid s \in \mathcal{S}\}$ is bounded, and therefore all $\sup_{s\in\mathcal{S}}\{|adom_h(s)|\}$ are finite, then an infinite and uniform AC-MAS $\mathcal{P}$ can in principle be verified by model checking a finite bisimilar system $\mathcal{P}'$, whose interpretation domains satisfy condition (1) in Theorem 3. In the next section we introduce a class of infinite and uniform AC-MAS that admits finite abstractions.

5 Finite Abstraction

In this section we state sufficient conditions to reduce the model checking problem for an infinite AC-MAS to the verification of a

finite system. The main result is given as Theorem 7, which guarantees that for bounded and rigid AC-MAS uniformity is sufficient to obtain bisimilar finite abstractions that preserve tFO-CTLK formulas. In the following we assume for technical reasons and w.l.o.g. that any AC-MAS $\mathcal{P}$ is such that $adom_h(s_0) \subseteq \mathcal{C}_h$ (as each $adom_h(s_0)$ is finite). Also, N_h is the sum of the maximum numbers of parameters of type T_h contained in the action types of each agent, i.e., $N_h = \sum_{A_i \in Ag} \max_{\{\alpha(\vec{x}) \in Act_i, \vec{x} \in Var_h\}} \{|\vec{x}|\}$.

Definition 18 (Bounded AC-MAS) *An AC-MAS $\mathcal{P}$ is b_h-*bounded*, for $b_h \in \mathbb{N}$, iff for all $s \in \mathcal{S}$, $|adom_h(s)| \leq b_h$.*

Thus, an AC-MAS is b_h-bounded if no active domain of its reachable state space contains more than b_h distinct elements of type T_h. An AC-MAS $\mathcal{P}$ is *bounded* if for every type T_h, $\mathcal{P}$ is b_h-bounded for some $b_h \in \mathbb{N}$. Observe that bounded AC-MAS may still contain infinitely many states, all bounded by some b_h. So, bounded AC-MAS are infinite-state systems in general, with a non-trivial model checking problem.

We now introduce abstractions in a modular manner by first defining abstract agents.

Definition 19 (Abstract agent) *Let $A = \langle \mathcal{D}, Act, Pr \rangle$ be an agent defined on a countable interpretation domain U_h for each type T_h. Given a countable set U'_h of individuals for each T_h, the* abstract agent *A' is a tuple $\langle \mathcal{D}', Act', Pr' \rangle$ on $U'_1, \ldots, U'_k$ s.t.* (i) *$\mathcal{D}' = \mathcal{D}$;* (ii) *$Act' = Act$; and* (iii) *$Pr'$ is the smallest function defined as follows:*

- *if $\alpha(\vec{u}) \in Pr(D)$, $D' \in \mathcal{D}'(\vec{U}')$ and $D' \simeq D$ for some witness ι, then $\alpha(\vec{u}') \in Pr'(D')$, where $\vec{u}' = \iota'(\vec{u})$ for some type-consistent constant-preserving bijection ι' extending ι to $\vec{u}$.*

Given a set Ag of agents, let Ag' be the set of the corresponding abstract agents.

We remark that A', as defined in Def. 19, is indeed an agent according to Def. 3. In particular, the protocol function Pr' is well-defined provided Pr is. We now present the notion of abstraction.

Definition 20 (Abstraction) *Let $\mathcal{P} = \langle Ag, s_0, \tau \rangle$ be an AC-MAS, and Ag' the set of abstract agents as in Def. 19. The AC-MAS $\mathcal{P}' = \langle Ag', s'_0, \tau' \rangle$ is an* abstraction *of $\mathcal{P}$ iff* (i) *$s'_0 \simeq s_0$, and* (ii) *τ' is the smallest function defined as follows*

- *if $s \xrightarrow{\alpha(\vec{u})} t$, $s', t' \in \mathcal{D}'(\vec{U})$ and $s \oplus t \simeq s' \oplus t'$ for some witness ι, then $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$ for some type-consistent constant-preserving bijection ι' extending ι to $\vec{u}$.*

Notice that $\mathcal{P}'$ is indeed an AC-MAS as it satisfies the relevant conditions on protocols and transitions in Def. 4. Also, by varying each U'_h we can obtain different abstractions. Moreover, the abstraction of a rigid AC-MAS is not itself rigid in general. The last point is key in the definition of finite abstractions.

We start with a lemma that extends Prop. 3.7 in [6], that is, req. 2 in Def. 16 can be derived from req. 1 whenever $adom_h(s_0) \subseteq \mathcal{C}_h$. Notice that, differently from [6], here we have to assume that the AC-MAS $\mathcal{P}$ is rigid.

Lemma 4 *If a rigid AC-MAS $\mathcal{P}$ satisfies req. 1 in Def. 16, and $adom_h(s_0) \subseteq \mathcal{C}_h$ for every type T_h, then req. 2 is also satisfied.*

Proof (sketch). If $s \oplus t \simeq s' \oplus t'$, then then there is a witness $\iota : adom_h(s) \cup adom_h(t) \cup \mathcal{C}_h \mapsto adom_h(s') \cup adom_h(t') \cup \mathcal{C}_h$ that is the identity on $\mathcal{C}_h$ (and hence on $adom_h(s_0)$). Suppose that $s \sim_i t$, i.e., $D_i(s) = D_i(t)$. In particular, $D_i(s') = \iota(D_i(s)) = \iota(D_i(t)) = D_i(t')$ by rigidity. However, it is not guaranteed that $s' \sim_i t'$, as we need to prove that $t' \in \mathcal{S}$. This can be done by showing that t' is reachable from s_0. Since t is reachable from s_0, there exists a run $s_0 \to s_1 \to \ldots \to s_k$ s.t. $s_k = t$. We now extend ι to a total and injective function $\iota'_h : adom_h(s_0) \cup \cdots \cup adom_h(s_k) \cup \mathcal{C}_h \mapsto U_h$. This can always be done because $|U_h| \geq |adom_h(s_0) \cup \cdots \cup adom_h(s_k) \cup \mathcal{C}_h|$. Moreover, if $\mathcal{P}$ is rigid, by the condition on rigid AC-MAS ι' can be found so that all D_{s_m} and $D_{\iota'(s_m)}$ agree on the interpretation of symbols $Q \in \mathcal{D}$. Indeed, consider $\vec{u} \in adom(s_0) \cup \cdots \cup adom(s_k) \cup \mathcal{C}$ s.t. $\vec{u} \in \mathcal{P}(Q)$. It might be the case that some $v \in \vec{u}$ belongs to $adom(s) \cup adom(t) \cup \mathcal{C}$. Hence, $\iota'(v) = \iota(v)$ is fixed. Let $\iota'(\vec{v})$ be the tuple of all such elements in the order they appear in $\vec{u}$. By the assumption of uniformity, we can find a tuple $\vec{w}$ s.t. either $\langle \vec{w}, \iota'(\vec{v}) \rangle$ or $\langle \iota'(\vec{v}), \vec{w} \rangle$ belongs to $\mathcal{P}(Q)$. Further, we can insert elements in each tuple so as to obtain a tuple $\vec{u}' \in \mathcal{P}(Q)$ and $\vec{u}' = \iota'(\vec{u})$. Finally, consider the sequence $\iota'(s_0), \iota'(s_1), \ldots, \iota'(s_k)$. Since $adom_h(s_0) \subseteq \mathcal{C}_h$, $\iota(s_0) = s_0$ and, since ι' extends ι, $\iota'(s_0) = \iota(s_0) = s_0$. Further, $\iota'(s_k) = \iota(t) = t'$. By repeated applications of req. 1 we can show that $\iota'(s_m) \xrightarrow{\alpha(\iota'(\vec{u}))} \iota'(s_{m+1})$ whenever $s_m \xrightarrow{\alpha(\vec{u})} s_{m+1}$, for $m < k$. Hence, the sequence is actually a run from s_0 to t'. Therefore, $t' \in \mathcal{S}$, and $s' \sim_i t'$. □

Next, we investigate the relationship between an AC-MAS and its abstractions.

Lemma 5 *Every abstraction $\mathcal{P}'$ of a rigid AC-MAS $\mathcal{P}$ is uniform.*

Proof (sketch). In the light of Lemma 4 and the assumption that $adom_h(s_0) \subseteq \mathcal{C}_h$, it is sufficient to prove condition 2 in Def. 16. Hence, suppose that $s, t, s' \in \mathcal{S}'$, $t' \in \mathcal{D}(U')$, $\alpha(\vec{u}) \in Act'(\vec{U}')$ s.t. $s \xrightarrow{\alpha(\vec{u})} t$, and $s \oplus t \simeq s' \oplus t'$ for some witness ζ. We need to show that $\mathcal{P}'$ admits a transition from s' to t'. Since $\mathcal{P}'$ is an abstraction of $\mathcal{P}$, given the definition of τ', there exist $s'', t'' \in \mathcal{P}$ and $\alpha(\vec{u}'') \in Act(\vec{U})$ s.t. $s'' \xrightarrow{\alpha(\vec{u}'')} t''$, $s'' \oplus t'' \simeq s \oplus t$ for some witness ι, and $\vec{u} = \iota'(\vec{u}'')$ for some type-consistent constant-preserving bijection ι' extending ι to $\vec{u}''$. Consider now $\vec{u}' \in \vec{U}'^{|\vec{u}|}$ s.t. $\vec{u}' = \zeta'(\vec{u})$ for some type-consistent constant-preserving bijection ζ' extending ζ to $\vec{u}$. The composition $\zeta' \cdot \iota'$ is a type-consistent constant-preserving bijection s.t. $\vec{u}' = \zeta'(\iota'(\vec{u}''))$. Moreover, it is a witness for $s'' \oplus t'' \simeq s' \oplus t'$. Since $\mathcal{P}'$ is an abstraction of $\mathcal{P}$, this implies that $s' \xrightarrow{\alpha(\vec{u}')} t'$. Thus, $\mathcal{P}'$ is uniform. □

The following result guarantees that for every uniform and rigid AC-MAS there exists a bisimilar abstraction, provided that the latter is built over sufficiently large interpretation domains.

Lemma 6 *Consider a uniform and rigid AC-MAS $\mathcal{P}$ over infinite interpretation domains U_h, and interpretation domains U'_h s.t. $\mathcal{C}_h \subseteq U'_h$. If for every type T_h, $|U'_h| \geq 2 \sup_{s \in \mathcal{S}} \{|adom_h(s)|\} + |\mathcal{C}_h| + N_h$, then there exists an abstraction $\mathcal{P}'$ of $\mathcal{P}$ over $U'_1, \ldots, U'_k$ that is bisimilar to $\mathcal{P}$.*

Proof (sketch). First consider the abstraction $\mathcal{P}'$ of $\mathcal{P}$ defined on $U'_1, \ldots, U'_k$ as specified in Def. 20. Then, we show that $B = \{\langle s, s' \rangle \in \mathcal{P} \times \mathcal{P}' \mid s \simeq s'\}$ is a bisimulation s.t. $\langle s_0, s'_0 \rangle \in B$. We start by proving that B is a simulation relation. To this end, observe that $\langle s_0, s'_0 \rangle \in B$ as $s_0 \simeq s'_0$. Next, consider $\langle s, s' \rangle \in B$, thus $s \simeq s'$. Suppose that $s \to t$, for some $t \in \mathcal{P}$. Then, there exists $\alpha(\vec{u}) \in$

$Act(\vec{U})$ s.t. $s \xrightarrow{\alpha(\vec{u})} t$. Moreover, since $|U'_h| \geq 2b_h + |\mathcal{C}_h| + N_h$, $\sum_{A_i \in Ag, \vec{u_i} \in U_h} |\vec{u}_i| \leq N_h$, and $|adom_h(s) \cup adom_h(t)| \leq 2b_h$, the witness ι for $s \simeq s'$ can be extended to a bijection ι' on $\bigcup_{A_i \in Ag} \vec{u}_i$. Now let $t' = \iota'(t)$. By the way ι' has been defined, it can be seen that $s \oplus t \simeq s' \oplus t'$. Further, since $\mathcal{P}'$ is an abstraction of $\mathcal{P}$, we have that $s' \xrightarrow{\alpha(\vec{u}')} t'$ for $\vec{u}' = \iota'(\vec{u})$, that is, $s' \to t'$ in $\mathcal{P}'$. As a result, there exists $t' \in \mathcal{P}'$ s.t. $s' \to t'$, $s \oplus t \simeq s' \oplus t'$, and $\langle t, t' \rangle \in B$.

As regards the epistemic relation, suppose that $s \sim_i t$ for some $A_i \in Ag$ and $t \in \mathcal{P}$. By definition of $\sim_i$ and rigidity, we have that $D_i(s) = D_i(t)$. Since $|U'_h| \geq 2b_h + |\mathcal{C}_h|$, any witness ι for $s \simeq s'$ can be extended to a witness ι' for $s \oplus t \simeq s' \oplus t'$, where $t' = \iota'(t) \cup \bigcup_{Q \in \mathcal{D}} D_i(s')(Q)$. Notice that t' is well-defined as $\bigcup_{Q \in \mathcal{D}} D_i(s')(Q)$ and $\iota'(t)$ coincides on the interpretation of symbols $Q \in \mathcal{D}$ in $adom(s')$. Hence, we have $D_i(s') = D_i(t')$. Thus, the only thing left to show that $s' \sim_i t'$ is that $t' \in \mathcal{S}'$, i.e., t' is reachable in $\mathcal{P}'$ from s'_0. To this end, observe that since $t \in \mathcal{P}$, there exists a run r s.t. $r(0) = s_0$ and $r(k) = t$, for some $k \geq 0$. Thus, there exist $\alpha^1(\vec{u}^1) \ldots, \alpha^k(\vec{u}^k)$ s.t. $r(j) \xrightarrow{\alpha^{j+1}(\vec{u}^{j+1})} r(j+1)$ for $0 \leq j < k$. Since $|U'_h| \geq 2b_h + |\mathcal{C}_h|$, we can define, for $0 \leq j < k$, a function ι_j that is a witness for $r(j) \oplus r(j+1) \simeq \iota_j(r(j)) \oplus \iota_j(r(j+1))$. In particular, this can be done starting from $j = k-1$, defining ι_{k-1} so that $\iota_{k-1}(r(k)) = \iota_{k-1}(t) = t'$, and proceeding backwards to $j = 0$, guaranteeing that, for $0 \leq j < k$, $\iota_j(r(j+1)) = \iota_{j+1}(r(j+1))$. Observe that since $adom(s_0) \subseteq \mathcal{C}$, necessarily $i_0(adom(r(0))) = i_0(adom(s_0)) = adom(s_0) = adom(s'_0)$. Moreover, as $|U'_h| \geq 2b_h + |\mathcal{C}_h| + N_h$, each ι_j can be extended to a bijection ι'_j, to the elements occurring in $\vec{u}^{j+1}$. Thus, given that $\mathcal{P}'$ is an abstraction of $\mathcal{P}$, for $0 \leq j < k$, we have that $\iota'_j(r(j)) \xrightarrow{\alpha^{j+1}(\iota'_j(\vec{u}^{j+1}))} \iota'_j(r(j+1))$. Hence, the sequence $\iota'_0(r(0)) \to \cdots \to \iota'_{k-1}(r(k))$ is a run from s'_0 in $\mathcal{P}'$, and since $t' = \iota'_{k-1}(r(k))$, t' is reachable in $\mathcal{P}'$. Therefore $s' \sim_i t'$. Further, since $t \simeq t'$, it is the case that $\langle t, t' \rangle \in B$, hence B is a simulation.

To prove that B^{-1} is a simulation, given $\langle s, s' \rangle \in B$ (thus $s \simeq s'$), suppose that $s' \to t'$ for some $t' \in \mathcal{P}'$. Then, there exists $\alpha(\vec{u}') \in Act(\vec{U}')$ s.t. $s' \xrightarrow{\alpha(\vec{u}')} t'$. Because $\mathcal{P}'$ is an abstraction of $\mathcal{P}$, there are $s'', t'' \in \mathcal{P}$ and $\alpha(\vec{u}'') \in Act(U)$ s.t. $s'' \oplus t'' \simeq s' \oplus t'$, for some witness ι, and $s'' \xrightarrow{\alpha(\vec{u}'')} t''$, with $\vec{u}'' = \iota'(\vec{u}')$ for some bijection ι' extending ι to $\vec{u}'$. Observe that $s' \simeq s''$, thus, by transitivity of $\simeq$, we have $s \simeq s''$. The fact that there exists $t \in \mathcal{P}$ s.t. $s \to t$ follows from the uniformity of $\mathcal{P}$. Thus, since $t' \simeq t$, we have $\langle t, t' \rangle \in B$.

For the epistemic relation, suppose that $s' \sim_i t'$, for some $t' \in \mathcal{P}'$ and $A_i \in Ag$. Let ι be a witness for $s' \simeq s$, and let ι' be an extension of ι that is a witness for $s' \oplus t' \simeq s \oplus t$. In particular, by reasoning as in Lemma 4 in Appendix A, we can define ι' so as to preserve the interpretation in $\mathcal{P}$ of symbols $Q \in \mathcal{D}$. Indeed, we have to compose the finite interpretations $D_{s'}(Q)$ and $D_{t'}(Q)$, for a symbol $Q \in \mathcal{D}$, into a unique interpretation $\mathcal{P}(Q)$. Suppose that $\vec{u}' \in D_{t'}(Q)$ and let $\vec{v}'$ be the elements of $\vec{u}'$ appearing in $adom(s')$. Then, $\iota(\vec{v}') \in adom(s)$ and by uniformity we can find elements to extend $\iota(\vec{v}')$ to a tuple $\vec{u}$ s.t. $\vec{u} \in \mathcal{P}(Q)$ and $\vec{u} = \iota'(\vec{u}')$ for such extension ι' of ι. Then, for $t = \iota'(t')$, it can be seen that $D_i(s) = D_i(t)$. Using an argument analogous to the one above, but exploiting the fact that $\mathcal{P}$ is uniform, $\mathcal{P}'$ is bounded, and $|U_h| > 2b_h + |\mathcal{C}_h| + N_h$ as U_h is infinite, we show that t is reachable by constructing a run r of $\mathcal{P}$ s.t. $r(k) = t$, for some $k \geq 0$. As a result, $s \sim_i t$. Further, since $t' \simeq t$, we have $\langle t, t' \rangle \in B^{-1}$. Therefore, B^{-1} is also a simulation, and $\mathcal{P}$ and $\mathcal{P}'$ are bisimilar. □

Finally, we observe that for bounded AC-MAS the value of $\sup_{s \in \mathcal{S}}\{|adom_h(s)|\}$ is always finite and equal to b_h. Thus, by combining Lemma 6 and Theorem 3 we can prove the main technical result of the paper.

Theorem 7 *Consider a bounded, uniform and rigid AC-MAS $\mathcal{P}$ over infinite interpretation domains U_h, a tFO-CTLK formula φ, and interpretation domains U'_h s.t. $\mathcal{C}_h \subseteq U'_h$. If for every type T_h, $|U'_h| \geq 2b_h + |\mathcal{C}_h| + \max\{|var_h(\varphi)|, N_h\}$, then there exists an abstraction $\mathcal{P}'$ of $\mathcal{P}$ over $U'_1, \ldots, U'_k$ such that*

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

Proof (sketch). By the hypothesis on the cardinalities of the various U_h and U'_h, Lemma 6 applies, so there exists an abstraction $\mathcal{P}'$ bisimilar to $\mathcal{P}$. Further, by Lemma 5 $\mathcal{P}'$ is uniform. Obviously, also $\mathcal{P}'$ is bounded. Thus, since $\mathcal{P}$ and $\mathcal{P}'$ are bounded, and by the cardinality hypothesis on U_h and U'_h, Theorem 3 applies. In particular, $|U'_h| \geq 2\sup_{s \in \mathcal{S}}\{|adom_h(s)|\} + |\mathcal{C}_h| + |var_h(\varphi)|$ as for all $s \in \mathcal{S}$, $|adom_h(s)| \leq b_h$ by boundedness. Therefore, $\mathcal{P} \models \varphi$ iff $\mathcal{P}' \models \varphi$. □

We remark that the U'_h in Theorem 7 might as well be finite. So, by using a sufficient number of abstract values in U'_h, we can in principle reduce the model checking problem for infinite-state AC-MAS to the verification of a finite abstraction. Specifically, we obtain the following result.

Corollary 8 *Given a bounded, uniform and rigid AC-MAS $\mathcal{P}$ over infinite interpretation domains $\vec{U}$, and a tFO-CTLK formula φ, there exists an abstract AC-MAS $\mathcal{P}'$ over finite interpretation domains $\vec{U'}$ s.t. φ is satisfied by $\mathcal{P}$ iff $\mathcal{P}'$ satisfies φ.*

To conclude this section we briefly outline how to derive a finite abstraction of the auction AC-MAS $\mathcal{A}$ in Section 3.

5.1 Abstract Auction

We observe that the auction AC-MAS $\mathcal{A}$ is indeed bounded, uniform and rigid. We showed above that $\mathcal{A}$ is uniform and rigid. As to boundedness, notice that the only infinite interpretation domain in $\mathcal{A}$ is the set $\mathbb{Q}$ of rational numbers. By definition of $\mathcal{A}$, for each global state s, there can be at most $|Items|(2|Ag|-1)$ distinct rational numbers in the active domain of s: $|Items|$ elements to represent base prices, $|Items|(|Ag|-1)$ elements to represent true values, and $|Items|(|Ag|-1)$ elements for bids. Further, consider the specifications appearing in Section 3 to be verified. No constant appears in these formulas and the active domain of the initial state s_0 is empty, therefore so is the set $\mathcal{C}_\mathbb{Q}$ of constants for rational numbers. Finally, 19 variables of type $\mathbb{Q}$ appear in our specifications, and this number exceeds $N_\mathbb{Q}$. As a consequence, we consider a finite abstract domain $U'_\mathbb{Q}$ of cardinality greater of equal to $2|Items|(2|Ag|-1)+19$, as required in Theorem 7.

We now describe briefly the abstract agents A' and $B'_1, \ldots, B'_\ell$ for the concrete auctioneer A and bidders $B_1, \ldots, B_\ell$. By Def. 19 the abstract bd schema $\mathcal{D}'$ and action types in Act' are the same as $\mathcal{D}$ and Act. As to the protocol functions, now these take values not in $\mathbb{Q}$ but $U'_\mathbb{Q}$. As an example, consider the clause for action $bid_i(it, bd)$ in Def. 10: $bid_i(it, bd) \in Pr_i(D)$ whenever the item it appears in $D(TValue_i)$, the highest bid bd_j in some Bid_j ($j \neq i$) for item it is strictly less than the true value tv for bidder B_i, $bd_j < bd \leq tv$, and $(it, \mathsf{active}) \in D(Status)$. Now, the condition on protocols in Def. 19 requires that for $D' \in \mathcal{D}'(\vec{U})$, $bid_i(it, bd') \in Pr'_i(D'_i)$ whenever $D' \simeq D$ for some witness ι. In particular, this means that

$bd' \in U'_{\mathbb{Q}}$ is an abstract value that has not yet been used to represent any bid in D'. By assumption $|U'_{\mathbb{Q}}| \geq 2|Items|(2|Ag| - 1) + 19$ on the cardinality of $U'_{\mathbb{Q}}$ in Theorem 7 it is always possible to find such an element.

Finally, given the set $Ag' = \{A', B'_1, \ldots, B'_\ell\}$ of abstract agents on $Items$, $\{\mathsf{active}, \mathsf{term}\}$ and $U'_{\mathbb{Q}}$, we briefly illustrate the abstract auction AC-MAS $\mathcal{A}' = \langle Ag', s'_0, \tau' \rangle$ where

- $s'_0 = s_0|_{adom(s_0)}$;
- τ' is the global transition function that mimicks τ. For instance, if $\alpha_i = bid_i(it, bd')$, then $s' \xrightarrow{\alpha(\vec{u})} t'$ whenever t' is the db instance that modifies s' by replacing any pair (it, bd) in $D'_j(Bid_i)$ with (it, bd'), where the value $bd' \in U'_{\mathbb{Q}}$ has been found as detailed above.

Moreover, by Def. 20 and the definition of isomorphism, we have that bd' is strictly greater than the highest bid bd_j in some Bid_j in s' for item it, but less than the true value tv for bidder B_i. This information defines the interpretation $D_{t'}(<)$ of symbol $<$ in t'. Indeed, the interpretation of $<$ in $\mathcal{A}'$ is not rigid, thus allowing for the reuse of values. In our example, the old value bd would no longer appear in the active domain of t'. Hence it might be used again in the next transition if needed.

Reasoning as above we can generate the whole state space $\mathcal{S}'$ of the abstract auction AC-MAS $\mathcal{A}'$, as $Items$, $\{\mathsf{active}, \mathsf{term}\}$ and $U'_{\mathbb{Q}}$ are finite. Then, we can model check our tFO-CTLK formulas on this finite abstraction. Indeed, both the concrete AC-MAS $\mathcal{A}$ and the interpretation domain $U'_{\mathbb{Q}}$ satisfy the hypotheses of Theorem 7. Thus, by this result it is guarantee that the specifications are satisfied in the abstraction $\mathcal{A}'$ iff they are satisfied in the concrete AC-MAS $\mathcal{A}$.

6 Conclusions and Future Work

In this paper we advanced the state-of-the-art on the verification of auctions by model checking. First, we extended the framework of artifact-centric multi-agent systems [6, 7] to support both typed languages and relations whose interpretation graph may be infinite. We argued that both features are essential to formalise English (ascending bid) auctions on multiple sessions running in parallel as AC-MAS. Most importantly, we provided a novel abstraction technique within this enhanced setting. As a result, we are now able to model check a significant class of infinite-state AC-MAS, including parallel English auctions, against sophisticated specifications in tFO-CTLK, by verifying their finite bisimilar abstractions.

In future work we aim at applying the theoretical results above to concrete use cases. Indeed, one relevant issue concerns the boundedness check for AC-MAS. In this respect, it would be of interest to find sufficient conditions ensuring boundedness, similarly to those discussed in [22]. Further, general constructive techniques to build abstractions from concrete AC-MAS are also essential for deployment in business processes.

REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.

[2] T. Ågotnes, P. Harrenstein, W. v. d. Hoek, and M. Wooldridge, 'Verifiable equilibria in boolean games', In Rossi [28].

[3] A. Badica and C. Badica, 'Specification and verification of an agent-based auction service', in *Information Systems Development*, eds., George Angelos Papadopoulos, Wita Wojtkowski, Gregory Wojtkowski, Stanislaw Wrycza, and Joe Zupancic, 239–248, Springer US, (2010).

[4] F. Belardinelli and A. Lomuscio, 'Decidability of model checking non-uniform artifact-centric quantified interpreted systems', In Rossi [28].

[5] F. Belardinelli, A. Lomuscio, and F. Patrizi, 'Verification of Deployed Artifact Systems via Data Abstraction', in *Proc. of the 9th International Conference on Service-Oriented Computing (ICSOC'11)*, pp. 142–156, (2011).

[6] F. Belardinelli, A. Lomuscio, and F. Patrizi, 'An Abstraction Technique for the Verification of Artifact-Centric Systems', in *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pp. 319 – 328, (2012).

[7] F. Belardinelli, A. Lomuscio, and F. Patrizi, 'Verification of GSM-Based Artifact-Centric Systems through Finite Abstraction', in *Proc. of the 10th International Conference on Service-Oriented Computing (ICSOC'12)*, pp. 17–31, (2012).

[8] F. Belardinelli, A. Lomuscio, and F. Patrizi, 'Verification of agent-based artifact systems', *CoRR*, **abs/1301.2678**, (2013).

[9] K. Bhattacharya, C. E. Gerede, R. Hull, R. Liu, and J. Su, 'Towards Formal Analysis of Artifact-Centric Business Process Models', in *Proc. of the 5th International Conference on Business Process Management (BPM'07)*, pp. 288–304, (2007).

[10] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 2001.

[11] J. M. Corera, I. Laresgoiti, and N. Jennings, 'Using archon, part 2: Electricity transportation management.', *IEEE Expert*, **11**(6), 71–79, (1996).

[12] E. Damaggio, A. Deutsch, and V. Vianu, 'Artifact Systems with Data Dependencies and Arithmetic', *ACM Transactions on Database Systems*, **37**(3), 22:1–22:36, (2012).

[13] G. De Giacomo, Y. Lespérance, and F. Patrizi, 'Bounded Situation Calculus Action Theories and Decidable Verification', in *Proc. of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, pp. 467–477, (2012).

[14] A. Deutsch, R. Hull, F. Patrizi, and V. Vianu, 'Automatic Verification of Data-centric Business Processes', in *Proc. of the 12th International Conference on Database Theory (ICDT'09)*, pp. 252–267, (2009).

[15] A. Deutsch, L. Sui, and V. Vianu, 'Specification and Verification of Data-Driven Web Applications', *Journal of Computer and System Sciences*, **73**(3), 442–474, (2007).

[16] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, New York, NY, USA, 2010.

[17] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning About Knowledge*, The MIT Press, 1995.

[18] D. Fischer, E. Grädel, and L. Kaiser, 'Model Checking Games for the Quantitative mu-Calculus', *Theory Comput. Syst.*, **47**(3), 696–719, (2010).

[19] C. E. Gerede and J. Su, 'Specification and Verification of Artifact Behaviors in Business Process Models', in *Proc. of the 5th International Conference on Service-Oriented Computing (ICSOC'07)*, pp. 181–192, (2007).

[20] E. M. Tadjouddine F. Guerin and W. Vasconcelos, 'Abstractions for model-checking game-theoretic properties of auctions', in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, AAMAS '08, pp. 1613–1616, Richland, SC, (2008). International Foundation for Autonomous Agents and Multiagent Systems.

[21] N. Haque, N. R. Jennings, and L. Moreau, 'Resource allocation in communication networks using market-based agents', *Knowledge-Based Systems*, **18**(4-5), 163–170, (August 2005).

[22] B. Bagheri Hariri, D. Calvanese, G. De Giacomo, A. Deutsch, and M. Montali, 'Verification of relational data-centric dynamic systems with external services', in *PODS*, eds., R. Hull and W. Fan, pp. 163–174. ACM, (2013).

[23] R. Hull, 'Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges', in *Proc. (part II) of Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008 (On the Move to Meaningful Internet Systems: OTM'08)*, pp. 1152–1163, (2008).

[24] R. Hull, E. Damaggio, R. De Masellis, F. Fournier, M. Gupta, F. T. Heath, S. Hobson, M. Linehan, S. Maradugu A. Nigam, P. Sukaviriya, and R. Vaculin, 'Business Artifacts with Guard-Stage-Milestone Lifecycles: Managing Artifact Interactions with Conditions and Events', in *Proc. of the 5th ACM International Conference on Distributed Event-Based Systems (DEBS'11)*, pp. 51–62, (2011).

[25] M. Kacprzak, W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, M. Szreter, B. Wozna, and A. Zbrzezny, 'VerICS 2007 - a Model Checker for Knowledge and Real-Time', *Fundamenta Informaticae*, **85**(1-4), 313–328, (2008).
[26] A. Lomuscio, H. Qu, and F. Raimondi, 'MCMAS: A Model Checker for the Verification of Multi-Agent Systems', in *Proc. of the 21st International Conference on Computer Aided Verification (CAV'09)*, pp. 682–688, (2009).
[27] Daniel M. Reeves, Michael P. Wellman, Jeffrey K. MacKie-Mason, and Anna Osepayshvili, 'Exploring bidding strategies for market-based scheduling', *Decision Support Systems*, **39**(1), 67–85, (2005).
[28] F. Rossi, ed. *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013.
[29] Nicolas Troquard, Wiebe van der Hoek, and Michael Wooldridge, 'Model checking strategic equilibria', in *MoChArt*, eds., Doron Peled and Michael Wooldridge, volume 5348 of *Lecture Notes in Computer Science*, pp. 166–188. Springer, (2008).
[30] M. Webster, L. Dennis, and M. Fisher, 'Model-checking auctions, coalitions and trust', Technical report, University of Liverpool, (2009).
[31] H. Xu and Y. Cheng, 'Model checking bidding behaviors in internet concurrent auctions.', *Comput. Syst. Sci. Eng.*, **22**(4), (2007).
[32] Haiping Xu, Christopher K. Bates, and Sol M. Shatz, 'Real-time model checking for shill detection in live online auctions', in *Software Engineering Research and Practice*, eds., Hamid R. Arabnia and Hassan Reza, pp. 134–140. CSREA Press, (2009).

A Proof of Theorem 3

In this section we give a proof of Theorem 3. Several partial results are required and presented below. A first, distinctive feature of uniform systems is that all isomorphic states are bisimilar.

Lemma 9 *If an AC-MAS $\mathcal{P}$ is uniform, then for every $s, s' \in \mathcal{P}$, $s \simeq s'$ implies $s \approx s'$.*

Proof (sketch). We prove that $B = \{\langle s, s'\rangle \in \mathcal{S} \times \mathcal{S} \mid s \simeq s'\}$ is a bisimulation. Observe that since $\simeq$ is an equivalence relation, so is B. Thus, B is symmetric and $B = B^{-1}$. Therefore, if B is a simulation then also B^{-1} is a simulation. Hence, it is sufficient to prove that B is a simulation. To this end, let $\langle s, s'\rangle \in B$, and assume $s \rightarrow t$ for some $t \in \mathcal{S}$. Then, $s \xrightarrow{\alpha(\vec{u})} t$ for some $\alpha(\vec{u}) \in Act(\vec{U})$. Consider a witness ι for $s \simeq s'$. By cardinality considerations ι can be extended to a total and injective function $\iota' : adom_h(s) \cup adom_h(t) \cup \{\vec{u}\} \cup \mathcal{C}_h \mapsto U_h$. Further, if $\mathcal{P}$ is rigid, by the condition on rigid AC-MAS ι' can be found so that D_s, D_t, $D_{s'}$ and $D_{\iota'(t)}$ agree on the interpretation of symbols $Q \in \mathcal{D}$. Indeed, consider $\vec{u} \in adom(s) \cup adom(t) \cup \mathcal{C}$ s.t. $\vec{u} \in \mathcal{P}(Q)$. It might be the case that some $v \in \vec{u}$ belongs to $adom(s) \cup \mathcal{C}$. Hence, $\iota'(v) = \iota(v)$ is fixed. Let $\iota'(\vec{v})$ be the tuple of all such elements in the order they appear in $\vec{u}$. By the assumption of uniformity, we can find a tuple $\vec{w}$ s.t. either $\langle \vec{w}, \iota'(\vec{v})\rangle$ or $\langle \iota'(\vec{v}), \vec{w}\rangle$ belongs to $\mathcal{P}(Q)$. Further, we can insert elements in each tuple so as to obtain a tuple $\vec{u}' \in \mathcal{P}(Q)$ and $\vec{u}' = \iota'(\vec{u})$. Now, consider $t' = \iota'(t)$; it follows that ι' is a witness for $s \oplus t \simeq s' \oplus t'$. Since $\mathcal{P}$ is uniform, $s' \xrightarrow{\alpha(\iota'(\vec{u}))} t'$, that is, $s' \rightarrow t'$. Moreover, ι' is a witness for $t \simeq t'$, thus $\langle t, t'\rangle \in B$.

Next assume that $\langle s, s'\rangle \in B$ and $s \sim_i t$, for some $t \in \mathcal{S}$. By reasoning as above we can find a witness ι for $s \simeq s'$ and an extension ι' of ι s.t. $t' = \iota'(t)$ and ι' is a witness for $s \oplus t \simeq s' \oplus t'$. Since $\mathcal{P}$ is uniform, $s' \sim_i t'$ and $\langle t, t'\rangle \in B$. □

Next we extend Lemma 4.6 in [8] to our setting to show that under appropriate cardinality constraints the bisimulation preserves the equivalence of assignments w.r.t. a given FO-CTLK formula.

Lemma 10 *Consider bisimilar and uniform AC-MAS $\mathcal{P}$ and $\mathcal{P}'$, bisimilar states $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$, and an tFO-CTLK formula φ. For every assignments σ and σ' equivalent for φ w.r.t. s and s', we have that*

1. *for every $t \in \mathcal{S}$, if* (i) $s \rightarrow t$, *and* (ii) *for every type T_h, $|U'_h| \geq |adom_h(s) \cup adom_h(t) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$, then there exists $t' \in \mathcal{S}'$ s.t. $s' \rightarrow t'$, $t \approx t'$, and σ and σ' are equivalent for φ w.r.t. t and t'.*
2. *for every $t \in \mathcal{S}$, if* (i) $s \sim_i t$, *and* (ii) *if for every type T_h, $|U'_h| \geq |adom_h(s) \cup adom_h(t) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$, then there exists $t' \in \mathcal{S}'$ s.t. $s' \sim_i t'$, $t \approx t'$, and σ and σ' are equivalent for φ w.r.t. t and t'.*

Proof (sketch). To prove (1), let γ be a bijection witnessing that σ and σ' equivalent for φ w.r.t. s and s'. Also, suppose that $s \rightarrow t$. Since $s \approx s'$, by definition of bisimulation there exists $t'' \in \mathcal{S}'$ s.t. $s' \rightarrow t''$, $s \oplus t \simeq s' \oplus t''$, and $t \approx t''$. Now, for every type T_h define $Dom_h(j) = adom_h(s) \cup adom_h(t) \cup \mathcal{C}_h$ and partition it into:

- $Dom(\gamma) = adom_h(s) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))$
- $Y = adom_h(t) \setminus Dom_h(\gamma)$

Observe that for each type T_h, $|Im_h(\gamma)| = |adom_h(s') \cup \mathcal{C}_h \cup \sigma'(fr_h(\varphi))| = |adom_h(s) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$, thus from the fact that $|U'_h| \geq |adom_h(s) \cup adom_h(t) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$ we have $|U'_h \setminus Im_h(\gamma)| \geq |Y|$. Since $|U'_h \setminus Im_h(\gamma)| \geq |Y|$, there exists a (invertible) total function $f : Y \mapsto U'_h \setminus Im_h(\gamma)$. We now define a function $j : Dom_h(j) \mapsto U'_h$ as follows:

$$j(u) = \begin{cases} \gamma(u), & \text{if } u \in Dom(\gamma) \\ f(u), & \text{if } u \in Y \end{cases}$$

Obviously, j is invertible. Moreover, if $\mathcal{P}'$ is rigid, by the condition on uniform and rigid AC-MAS, the function j can be defined so as to preserve the interpretation $\mathcal{P}'(Q)$ of symbols $Q \in \mathcal{D}$, similarly as in the proof of Lemma 9. Thus, j is a witness for $s \oplus t \simeq s' \oplus t'$, where $t' = j(t)$. In particular, we have that $s \oplus t \simeq s' \oplus t'$. Also, $s \oplus t \simeq s' \oplus t''$ implies $s' \oplus t' \simeq s' \oplus t''$ by transitivity of $\simeq$. Thus, $s' \rightarrow t'$, as $\mathcal{P}'$ is uniform. Moreover, σ and σ' are equivalent for φ w.r.t. t and t', by construction of t'. To check that $t \approx t'$, observe that, since $t' \simeq t''$ and $\mathcal{P}'$ is uniform, by Lemma 9 it follows that $t' \approx t''$. Thus, since $t \approx t''$ and $\approx$ is transitive, we obtain that $t \approx t'$. The proof for (2) has an analogous structure and is therefore omitted. □

For technical convenience we shall use also the concept of *temporal-epistemic run* (t.e. run for short). Formally a t.e. run r from a global state s is an infinite sequence $s^0 \rightsquigarrow s^1 \rightsquigarrow \ldots$ such that $s^0 = s$ and $s^i \rightarrow s^{i+1}$ or $s^i \sim_k s^{i+1}$, for some $k \in Ag$. A state s' is *t.e. reachable* from s if there exists a t.e. run r from the global state $r(0) = s$ s.t. $r(i) = s'$, for some $i \geq 0$. Obviously, temporal-epistemic runs include purely temporal runs as a special case. The following result shows that Lemma 10 generalizes to t.e. runs.

Lemma 11 *Consider bisimilar and uniform AC-MAS $\mathcal{P}$ and $\mathcal{P}'$, bisimilar states $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$, a tFO-CTLK formula φ, and assignments σ and σ' equivalent for φ w.r.t. s and s'. For every t.e. run r of $\mathcal{P}$, if* (i) $r(0) = s$, *and* (ii) *for all $i \geq 0$, for all types T_h, $|U'_h| \geq |adom_h(r(i)) \cup adom_h(r(i+1)) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$, then there exists a t.e. run r' of $\mathcal{P}'$ s.t. for all $i \geq 0$:*

(i) $r'(0) = s'$;
(ii) $r(i) \approx r'(i)$;
(iii) *σ and σ' are equivalent for φ w.r.t. $r(i)$ and $r'(i)$;*

(iv) *for every* $i \geq 0$*, if* $r(i) \rightarrow r(i+1)$ *then* $r'(i) \rightarrow r'(i+1)$*, and if* $r(i) \sim_j r(i+1)$*, for some* $A_j \in Ag$*, then* $r'(i) \sim_j r'(i+1)$.

Proof (sketch). The result follows from Lemma 10, the proof is similar to Lemma 3.10 in [6]. Specifically, let r be a t.e. run s.t. $|U'_h| \geq |adom_h(r(i)) \cup adom_h(r(i+1)) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$ for all types h and $i \geq 0$. We inductively build r' and show that the conditions above are satisfied. Suppose that $r(i) \approx r'(i)$ and σ and σ' are equivalent for φ w.r.t. $r(i)$ and $r'(i)$ (we notice that for $i = 0$ this is indeed the case as $r(i) = s \approx s' = r'(i)$). Since $r(i) \rightsquigarrow r(i+1)$ and $|U'_h| \geq |adom_h(r(i)) \cup adom_h(r(i+1)) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))|$, by Lemma 10 there exists $t' \in \mathcal{S}'$ s.t. $r'(i) \rightsquigarrow t'$, σ and σ' are equivalent for φ w.r.t. $r(i+1)$ and t', and $r(i+1) \approx t'$. Let $r'(i+1) = t'$. It is clear that r' is a t.e. run in $\mathcal{P}'$, and that, by Lemma 10, the transitions of r' can be chosen so as to fulfill requirement *(iv)*. □

We can now prove the following result, which states that FO-CTLK formulas cannot distinguish bisimilar and uniform AC-MAS.

Theorem 12 *Consider bisimilar and uniform AC-MAS* $\mathcal{P}$ *and* $\mathcal{P}'$*, bisimilar states* $s \in \mathcal{S}$ *and* $s' \in \mathcal{S}'$*, an FO-CTLK formula* φ*, and assignments* σ *and* σ' *equivalent for* φ *w.r.t.* s *and* s'*. If*

1. *for every t.e. run* r *s.t.* $r(0) = s$*, for all* $k \geq 0$ *we have* $|U'_h| \geq |adom_h(r(k)) \cup adom_h(r(k+1)) \cup \mathcal{C}_h \cup \sigma(fr_h(\varphi))| + |var_h(\varphi) \setminus fr_h(\varphi)|$*;*
2. *for every t.e. run* r' *s.t.* $r'(0) = s'$*, for all* $k \geq 0$ *we have* $|U_h| \geq |adom_h(r'(k)) \cup adom_h(r'(k+1)) \cup \mathcal{C}_h \cup \sigma'(fr_h(\varphi))| + |var_h(\varphi) \setminus fr_h(\varphi)|$*;*

then

$$(\mathcal{P}, s, \sigma) \models \varphi \quad \textit{iff} \quad (\mathcal{P}', s', \sigma') \models \varphi.$$

Proof (sketch). The proof is by induction on the structure of φ and makes use of Lemma 11 for the inductive cases concerning the modal operators. The inductive cases for propositional connectives are straightforward, while the base case for atomic formulas follows from Lemma 2. The proof details are similar to Theorem 3.11 in [6] □

We now state the main contribution of this section, which lifts the result in [6] to AC-MAS with types and predicates with an infinite interpretation. Hereafter $\sup_{s \in \mathcal{S}}\{|adom_h(s)|\} = \infty$ whenever an AC-MAS $\mathcal{P}$ is unbounded, i.e., there is no $b \in \mathbb{N}$ s.t. $|adom_h(s)| \leq b$ for all $s \in S$.

Theorem 3 *Consider bisimilar and uniform AC-MAS* $\mathcal{P}$ *and* $\mathcal{P}'$*, and a tFO-CTLK formula* φ*. If for every* T_h*,*

1. $|U'_h| \geq 2 \sup_{s \in \mathcal{S}}\{|adom_h(s)|\} + |\mathcal{C}_h| + |var_h(\varphi)|$
2. $|U_h| \geq 2 \sup_{s' \in \mathcal{S}'}\{|adom_h(s')|\} + |\mathcal{C}_h| + |var_h(\varphi)|$

then

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

Proof (sketch). Equivalently, we prove that if $(\mathcal{P}, s_0, \sigma) \not\models \varphi$ for some σ, then there exists σ' s.t. $(\mathcal{P}', s'_0, \sigma') \not\models \varphi$, and viceversa. To this end, notice that hypotheses 1 (resp. 2) of Theorem 3 implies hypotheses 1 (resp. 2) of Lemma 12. Further, we observe that, by cardinality considerations, given assignment $\sigma : Var_h \mapsto U_h$, there exists an assignment $\sigma' : Var_h \mapsto U'_h$ s.t. σ and σ' are equivalent for φ w.r.t. s_0 and s'_0. Thus, by an application of Theorem 12 if there exists an assignment σ s.t. $(\mathcal{P}, s_0, \sigma) \not\models \varphi$, then there exists an assignment σ' s.t. $(\mathcal{P}', s'_0, \sigma') \not\models \varphi$. The converse can be proved analogously, as the hypotheses are symmetric. □