Finite Abstractions for the Verification of Epistemic Properties in Open Multi-Agent Systems

Francesco Belardinelli
Laboratoire IBISC
Université d'Evry, France
belardinelli@ibisc.fr

Davide Grossi
Department of Computing
Liverpool University, UK
D.Grossi@liverpool.ac.uk

Alessio Lomuscio
Department of Computing
Imperial College London, UK
a.lomuscio@imperial.ac.uk

Abstract

We develop a methodology to model and verify open multi-agent systems (OMAS), where agents may join in or leave at run time. Further, we specify properties of interest on OMAS in a variant of first-order temporal-epistemic logic, whose characterising features include epistemic modalities indexed to individual terms, interpreted on agents appearing at a given state. This formalism notably allows to express group knowledge dynamically. We study the verification problem of these systems and show that, under specific conditions, finite bisimilar abstractions can be obtained.

1 Introduction

Modal temporal-epistemic logic has long been adopted as a formalism for reasoning about multi-agent systems (MAS). In its basic setting it consists of either the linear or the branching version of discrete-time temporal logic, augmented with knowledge modalities for the agents in the system. Several properties of MAS (e.g., perfect recall, no learning, synchronicity) have been axiomatised on the widely adopted semantics of Interpreted Systems [Fagin *et al.*, 1995]. In the past decade several model checking methodologies and toolkits that use temporal-epistemic specification languages have been developed [Penczek and Lomuscio, 2003; Gammie and van der Meyden, 2004; Lomuscio *et al.*, 2009].

Two key assumptions are made in the basic setting of the formalism above. Firstly, facts are expressed in propositional terms. Secondly, the number of agents is finite and given at design time. As a consequence, the indexes of individual knowledge operators are constants in a finite set Ag of agents, while the indexes for group knowledge operators are finite subsets of Ag. Proposals have been made to overcome the first limitation by introducing first-order versions of temporal-epistemic logic both on quantified versions of Interpreted Systems [Belardinelli and Lomuscio, 2012] and on Artifact-centric Multi-agent Systems [Belardinelli *et al.*, 2014]. These approaches surmount the limits of a purely propositional language by extending the syntax to fully-fledged first-order formulas. In some cases completeness can be retained [Belardinelli and Lomuscio, 2012] and verification can be performed on finite abstractions [Belardinelli and Lomuscio, 2013; Belardinelli *et al.*, 2014].

Regarding the second limitation, proposals have been put forward to consider a set of objects that vary at design time; the set of agents is normally considered to be finite in each system run. This is a sensible assumption in many scenarios, but there are applications of MAS (e.g., e-commerce, smart grids) where an unbounded number of agents may freely enter and leave the system at run time. There is, therefore, a need to account for the unbounded and possibly infinite agents joining in or leaving an open MAS. In this setting it is still of interest to reason about their evolution and what they know individually and collectively. For example, in an auction setting, such as fishmarkets [Rodríguez-Aguilar *et al.*, 1998], different agents attend different auctions at run time. Nonetheless, all of them, however many they may be, will eventually know what the reserve price for a particular good is. Formally, such a temporal-epistemic specification can be expressed in the proposed formalism as:

$$AG\, \forall x(\mathit{Good}(x) \rightarrow \exists y\ AF\ \forall z\ K_z \mathit{Price}(y, x)) \qquad (1)$$

(1) intuitively expresses that any good x has always a price y that will eventually be learnt by all agents z currently attending the auction. A key feature of this specification is that agents appear as quantifiable terms in the logical language and appear as such in the indexes of the epistemic operators. In particular, compare the subformula $\forall z K_z \mathit{Price}(y, x))$ of (1), where the quantification domain of $\forall z$ changes depending on the state, with the standard temporal-epistemic formula $\bigwedge_{a \in Ag} K_a \mathit{Price}(y, x)$, which assumes Ag fixed. In this paper we propose a formalism accounting for (1); we show that, while the verification problem is undecidable in general, bounded MAS admit finite abstractions.

Related Work. A quantified doxastic logic, with modalities indexed by variables, was introduced in [Lomuscio and Colombetti, 1996]. However, this focused on a 3-valued semantics in view of providing sound axiomatisations. In contrast, here we consider a 2-valued semantics and the model checking problem. Various classes of models and types of quantifications have been developed recently to account for Artifact-centric Systems [Belardinelli *et al.*, 2012; 2014; Hariri *et al.*, 2013]. While our work is influenced by the techniques introduced therein, none of these contributions deals with open MAS with a possibly infinite number of agents. In [Montali *et al.*, 2014] agents appear in the relational structure, but there is no explicit quantification on them. This feature extends the expressiveness of the framework we propose. There are also similarities with recent work on parametrised verification of MAS [Kouvaros and Lomuscio, 2013b; 2013a]. However, parametrised verification aims at establishing whether properties hold irrespective of the finite but unbounded number of agents in the system; here we deal with an infinite set of agents which we only bound at a state and not in the whole model. Closer to our approach is [Belardinelli and Grossi, 2015] which introduces a semantics for dynamic agent networks; however, epistemic operators are not discussed there.

Scheme of the paper. In Section 2 we formalise open multi-agent systems in our setting and introduce a novel first-order temporal-epistemic logic CTLK$_x$. We state the model checking problem for this setting (Section 2.2) and illustrate the formal machinery with a use case (Section 2.3). Section 3 contains the main theoretical results on the existence of finite, bisimilar abstractions. We conclude and point to future work in Section 4.

2 Open Multi-agent Systems

In this section we present a formalism to reason about open multi-agent systems (OMAS). A key feature of OMAS is that agents may join and leave the system at run time. We then put forward a first-order version of the temporal-epistemic logic CTLK to reason about OMAS, that allows us to index knowledge operators with variables. We conclude by formulating the model checking problem for OMAS. Since we wish to account for possibly infinite domains of objects and agents we import some basic terminology from related literature [Abiteboul *et al.*, 1995; Belardinelli *et al.*, 2014].

Definition 1 (Database schema and instance) *A* database schema *is a finite set* $\mathcal{D} = \{P_1/q_1, \ldots, P_n/q_n\}$ *of* predicate symbols P *with arity* $q \in \mathbb{N}$.

Given a (possibly infinite) interpretation domain X*, a* $\mathcal{D}$-instance *over* X *is a mapping* D *associating each predicate symbol* P *to a* finite *q-ary relation on X, i.e.,* $D(P) \subseteq X^q$.

For a database schema $\mathcal{D}$, $\mathcal{D}(X)$ is the set of all $\mathcal{D}$-instances on X; while the *active domain* $adom(D)$ is the *finite* set $\bigcup_{P\in\mathcal{D}}\{u_1, \ldots, u_q \in X \mid \langle u_1, \ldots, u_q\rangle \in D(P)\}$ of all individuals occurring in some predicate interpretation $D(P)$. Further, the *primed version* of a database schema $\mathcal{D}$ as above is the schema $\mathcal{D}' = \{P'_1/q_1, \ldots, P'_n/q_n\}$. Then, the *disjoint union* $D \oplus D'$ of $\mathcal{D}$-instances D and D' is the $(\mathcal{D} \cup \mathcal{D}')$-instance s.t. *(i)* $D \oplus D'(P) = D(P)$, and *(ii)* $D \oplus D'(P') = D'(P)$. Hereafter, primed versions and disjoint unions are used to account for the temporal evolution of a database from the previous state D to the next state D'.

2.1 Agents in OMAS

To introduce OMAS, we import some preliminary notions from [Belardinelli and Grossi, 2015]. Hereafter we assume a finite number of *agent types* $T_0, \ldots, T_k$. Each agent type T comprises *(i)* a *local database schema* $\mathcal{D}_T$, and *(ii)* a finite set Act_T of *parametric actions* $\alpha(\vec{x})$. Hence, agents of the same type share the database schema and available actions. For every agent type T, Ag_T, Ag'_T, ... are (possibly infinite) sets of agent names. In the rest of the paper, the interpretation domain X contains a set Ag_T of agent names for each type T, i.e., $X = Ag \cup U$ for $Ag = \bigcup_{type\ T} Ag_T$ and some other (possibly empty) set U of elements. We will also consider a set $Con \subseteq X$ of constants, including names for agents. To describe the temporal evolution of OMAS, we define protocols for agent types. To do so, we first introduce isomorphisms on database instances.

Definition 2 (Instance Isomorphism) *Instances* $D \in \mathcal{D}(X)$ *and* $D' \in \mathcal{D}(X')$ *are* isomorphic, *or* $D \simeq D'$*, iff for some bijection* $\iota : adom(D) \cup Con \mapsto adom(D') \cup Con$*,* (i) *$\iota$ is the identity on Con;* (ii) *ι is type-preserving, i.e., for every type T, ι is a bijection from* $(adom(D) \cup Con) \cap Ag_T$ *into* $(adom(D') \cup Con) \cap Ag'_T$*; and* (iii) *for every* $P \in \mathcal{D}$, $\vec{u} \in X^q$, $\vec{u} \in D(P)$ *iff* $\iota(\vec{u}) \in D'(P)$.

Whenever the above holds, we say that ι is a *witness* for $D \simeq D'$ and write $D \overset{\iota}{\simeq} D'$ to state this explicitly. While isomorphisms depend on the set *Con* of constants, in what follows we consider *Con* fixed and omit it.

We now introduce the *local protocol* Pr_T for a type T.

Definition 3 (Protocols) *Given domain* X*,* Pr_T *is a function from* $\mathcal{D}_T(X)$ *to* $2^{Act_T(X)}$*, where* $Act_T(X)$ *is the set of* ground actions $\alpha(\vec{u})$*, for* $\alpha(\vec{x}) \in Act_T$ *and* $\vec{u} \in X^{|\vec{x}|}$.

By Def. 3 the protocol Pr_T returns a ground action in $Act_T(X)$ for every $\mathcal{D}_T$-instance. In the rest of the paper we assume the following requirement on protocols:

for all instances $D, D' \in \mathcal{D}_T(X)$, if $D \overset{\iota}{\simeq} D'$ then
$\alpha(\vec{u}) \in Pr_T(D)$ iff $\alpha(\iota(\vec{u})) \in Pr_T(D')$ $\quad (*)$

So, by requirement $(*)$ isomorphic states allow "isomorphic" ground actions. Most OMAS of interest satisfy $(*)$. For example, in an English auction an agent may make a valid bid as long as the bid is above the current best price.

We now introduce the notion of *agent*.

Definition 4 (Agents) *Given an agent name* $a \in Ag_T$ *of type* T*, an* agent *is a tuple* $a = \langle \mathcal{D}_T, Act_T, Pr_T\rangle$ *where* $\mathcal{D}_T$*,* Act_T*, and* Pr_T *are defined as above.*

We assume a finite number of agent types, but we do not assume a bound on the number of agents of each type in any concrete instantiation of the system. This is common place in OMAS, such as in services, auctions, etc., whereby engineers have prior knowledge of the behaviour of the agent types without knowing how many instances of each type will be executed at runtime. We provide an example of this in Section 2.3. Agents as in Def. 4 are related to the notion of *agent templates* introduced in [Kouvaros and Lomuscio, 2013b; 2013a]. However, while the latter assumes that any concrete run admits a finite number of agents built on these types, we do not make this assumption here.

In the following an agent is often identified with her name; therefore we write $a = \langle \mathcal{D}_a, Act_a, Pr_a\rangle$ and omit the type. By Def. 4 a local state $l \in \mathcal{D}_a(U \cup Ag)$ encodes the knowledge of agent a about the elements in U as well as fellow agents in Ag. Thus, a fundamental difference with the standard approach to multi-agent systems [Parikh and Ramanujam, 1985;

Fagin *et al.*, 1995; Wooldridge, 2001] is that the agent's information is structured as a relational database.

We can now introduce OMAS to represent the interactions amongst agents, beginning with the notion of global state.

Definition 5 (Global States) *Given a* finite *subset $A \subseteq Ag$ of agents $a_i = \langle \mathcal{D}_i, Act_i, Pr_i \rangle$ defined on domain $X = U \cup Ag$, for $i \leq n$, a* global state *is a tuple $s = \langle l_0, \ldots, l_n \rangle$ of instances $l_i \in \mathcal{D}_i(X)$ s.t. $\bigcup_{i \leq n} adom(l_i) \cap Ag \subseteq A$.*

Note that, while we admit an infinite number of agents in existence, only a finite number of them can be active at any given time, and different agents can be active at different times, thus accounting for the openness of the system. Also by Def. 5, a global state s comprises at least all agents appearing in its active domain $adom(s) = \bigcup_{i \leq n} adom(l_i)$. For instance, if agent a appears in the local state $l_b \in \mathcal{D}_b(X)$ of agent $b \in A$, and thus $a \in adom(s)$, then a also belongs to A. By assuming a fixed enumeration of agents, we will identify global states containing the same local states for the same agents, possibly in a different order. Further, let ag be the function that for any global state $s = \langle l_0, \ldots, l_n \rangle$ returns the set $ag(s) = \{a_0, \ldots, a_n\}$ of agents s.t. $l_i \in \mathcal{D}_{a_i}(X)$ for $i \leq n$. By the requirement above on global states, for every state s, $adom(s) \cap Ag \subseteq ag(s)$. We let $\mathcal{G}$ be the set $\bigcup_{n \in \mathbb{N}}(\prod_{i \leq n} \mathcal{D}_{a_i}(X))$ of all global states. As a consequence, $\mathcal{G}$ is infinite whenever X is.

To account for the knowledge of agents, we say that states $s = \langle l_0, \ldots, l_n \rangle$ and $s' = \langle l'_0, \ldots, l'_m \rangle$, of possibly different lengths, are *epistemically indistinguishable* for agent a_i, or $s \sim_i s'$, iff $a_i \in ag(s)$, $a_i \in ag(s')$, and $l_i = l'_i$. Since s and s' can be tuples of different length, an agent does not generally know the exact number of active agents at each moment, nor their identity. Observe that if $a \notin ag(s)$, then the set $\{s' \in \mathcal{G} \mid s' \sim_a s\}$ is empty. That is, if agent a is not active in state s, then no state is indistinguishable for her. We elaborate more on this point in Section 2.2.

Finally, we introduce open multi-agent systems.

Definition 6 (OMAS) *Given a (possibly infinite) domain $X = Ag \cup U$ containing a (possibly infinite) set $Ag = \{a_0, a_1, \ldots\}$ of agents $a_i = \langle \mathcal{D}_i, Act_i, Pr_i \rangle$, an* open multi-agent system *is a tuple $\mathcal{P} = \langle Ag, U, I, \tau \rangle$ where*

- *I is the set of* initial states *s_0 for some finite $ag(s_0) \subseteq Ag$;*
- *$\tau : \mathcal{G} \times Act(X) \mapsto 2^{\mathcal{G}}$ is the* global transition function*, where Act is the set of joint (parametric) actions, and $\tau(\langle l_0, \ldots, l_n \rangle, \langle \alpha_0(\vec{u}_0), \ldots, \alpha_n(\vec{u}_n) \rangle)$ is defined iff $\alpha_i(\vec{u}_i) \in Pr_i(l_i)$ for every $i \leq n$.*

An OMAS describes all system's executions from an initial state $s_0 \in I$, according to the global transition function τ, which returns the successor states $\tau(s, \alpha(\vec{u})) \subseteq \mathcal{G}$ given the current state s and joint ground action $\alpha(\vec{u})$ by all agents in s. Since the domain X is typically infinite, OMAS are infinite-state systems in general. Specifically, OMAS are *open* and *dynamic* as global states may be tuples of different length, comprising different agents. Differently from most literature on MAS [Parikh and Ramanujam, 1985; Fagin *et al.*, 1995; Wooldridge, 2001], which assumes that the set of agents is finite and fully specified at design time, here the successor states returned by the transition function may contain fewer or more agents w.r.t. the current state.

We now state a requirement on joint actions in OMAS. To introduce it, we first extend isomorphisms to global states.

Definition 7 (State Isomorphism) *The global states $s \in \mathcal{G}$ and $s' \in \mathcal{G}'$ are* isomorphic*, or $s \simeq s'$, iff for some bijection $\iota : adom(s) \cup Con \cup ag(s) \mapsto adom(s') \cup Con \cup ag(s')$, for every $a_j \in ag(s)$, ι is a witness for $l_{a_j} \simeq l'_{\iota(a_j)}$.*

Any function ι as above is a *witness* for $s \simeq s'$, also indicated as $s \overset{\iota}{\simeq} s'$. As for instance isomorphisms, $\simeq$ is an equivalence relation, and by Def. 7 isomorphic states are tuples of the same length. In the rest of the paper we impose the following requirement on the transition functions in OMAS:

for all states $s, s' \in \mathcal{G}$, $s \overset{\iota}{\simeq} s'$ implies that $t \in \tau(s, \alpha(\vec{u}))$ iff $\iota(t) \in \tau(s', \alpha(\iota(\vec{u})))$ $\quad (+)$

Similarly to protocols, requirement $(+)$ guarantees that actions performed with "isomorphic" values in isomorphic states, also return isomorphic states. In Section 2.3 we will discuss an OMAS satisfying $(+)$; but similar assumptions are common place in database theory and the theory of programming languages [Hariri *et al.*, 2013; Deutsch *et al.*, 2007].

We now introduce some useful notation. We define the *transition relation* $s \to s'$ on global states iff $s \xrightarrow{\alpha(\vec{u})} s'$ for some joint ground action $\alpha(\vec{u})$, i.e., $s' \in \tau(s, \alpha(\vec{u}))$. An s-*run* r is an infinite sequence $s^0 \to s^1 \to \cdots$, with $s^0 = s$. For $n \in \mathbb{N}$, we set $r(n) = s^n$. A state s' is *reachable* from s iff $s' = r(i)$ for some s-run r and $i \geq 0$. Hereafter we enforce seriality on the transition relation $\to$ by assuming skip actions. Further, we introduce $\mathcal{S}$ as the set of states reachable from some initial state $s_0 \in I$. Since the domain X may be infinite, the set $\mathcal{S}$ of reachable states is also infinite in principle. Indeed, OMAS are infinite-state systems in general. Finally, we will refer to the *global* database schema $\mathcal{D}_s = \mathcal{D}_0 \cup \cdots \cup \mathcal{D}_n$ of a state $s = \langle l_0, \ldots, l_n \rangle$, and the corresponding $\mathcal{D}_s$-instance D_s s.t. $D_s(P) = \bigcup_{i \leq n} l_i(P)$, for $P \in \mathcal{D}_s$. Therefore, we suppose that each agent has a truthful, yet limited, view of the global database $\mathcal{D}_s$. Also, the *disjoint union* $s \oplus s'$ is defined as state $s'' = \langle l''_0, \ldots, l''_m \rangle$ on $ag(s) \cup ag(s')$ s.t. *(i)* if $a_i \in ag(s) \cap ag(s')$ then $l''_i = l_i \oplus l'_i$; *(ii)* if $a_i \in ag(s) \setminus ag(s')$ then $l''_i = l_i$; and *(iii)* if $a_i \in ag(s') \setminus ag(s)$ then $l''_i = l'_i$.

2.2 The Specification Language FO-CTLK$_x$

We now introduce FO-CTLK$_x$, a first-order extension of the temporal epistemic logic CTLK, as a specification language for OMAS. Differently from other quantified temporal-epistemic logics [Belardinelli *et al.*, 2014], FO-CTLK$_x$ features an expressive formulation of the epistemic operators that can be indexed by individual terms. Below we consider a set *Var* of *individual variables* containing a set Var_{Ag} of variables for agents, as well as the database schema $\mathcal{D} = \bigcup_{type\ T} \mathcal{D}_T$. Terms $t, t', \ldots$ are either variables or constants in *Con*.

Definition 8 (FO-CTLK$_x$) *The FO-CTLK$_x$ formulas are defined in BNF as follows:*

$$\varphi ::= P(\vec{t}) \mid t = t' \mid \neg\varphi \mid \varphi \to \varphi \mid \forall x \varphi \mid AX\varphi \mid A\varphi U \varphi \mid E\varphi U \varphi \mid K_a \varphi \mid K_z \varphi$$

where t, t' are terms, $P \in \mathcal{D}$, $a \in Con \cap Ag$, $z \in Var_{Ag}$, and $\vec{t}$ is a q-tuple of terms.

The temporal formulas $AX\varphi$ and $A\varphi U\varphi'$ (resp. $E\varphi U\varphi'$) are read as "for all runs, next φ" and "for every (resp. some) run, φ until φ'". The epistemic formula $K_t\varphi$ means that "the agent denoted by t knows φ". The fact that epistemic modalities are indexed to terms represents a significant difference w.r.t. standard approaches. Free and bound variables are defined as standard, as well as sets $var(\phi)$ (resp. $fr(\phi), con(\phi)$) of all variables (resp. free variables, constants) in ϕ. Notice that $z \in fr(K_z\phi)$ and $a \in con(K_a\phi)$. The same symbols are sometimes used to refer to individual variables and action parameters, the context will disambiguate.

To define the satisfaction of an FO-CTLK$_x$ formula on an OMAS, we introduce the notion of an *assignment* $\sigma : Var \mapsto X$ s.t. for every $z \in Var_{Ag}$, $\sigma(z) \in Ag$. We denote by σ^x_u the assignment s.t. *(i)* $\sigma^x_u(x) = u$; and *(ii)* $\sigma^x_u(x') = \sigma(x')$ for every x' different from x. Also, $\sigma(c) = c$ for all $c \in Con$.

Definition 9 (Semantics of FO-CTLK$_x$) *We define whether an OMAS $\mathcal{P}$* satisfies *a formula φ in a state s according to assignment σ, or $(\mathcal{P}, s, \sigma) \models \varphi$, as follows (clauses for propositional connectives are omitted as straightforward):*

$(\mathcal{P}, s, \sigma) \models P(\vec{t})$ *iff* $\langle \sigma(t_1), \ldots, \sigma(t_q)\rangle \in D_s(P)$
$(\mathcal{P}, s, \sigma) \models t = t'$ *iff* $\sigma(t) = \sigma(t')$
$(\mathcal{P}, s, \sigma) \models \forall x\varphi$ *iff for all* $u \in adom(s) \cup ag(s)$, $(\mathcal{P}, s, \sigma^x_u) \models \varphi$
$(\mathcal{P}, s, \sigma) \models AX\varphi$ *iff for all s-runs* r, $(\mathcal{P}, r(1), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models A\varphi\mathcal{U}\varphi'$ *iff for all s-runs* r, $(\mathcal{P}, r(k), \sigma) \models \varphi'$ *for some* $k \geq 0$, *and for all* j, $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models E\varphi\mathcal{U}\varphi'$ *iff for some s-run* r, *for some* $k \geq 0$, $(\mathcal{P}, r(k), \sigma) \models \varphi'$, *and for all* j, $0 \leq j < k$ *implies* $(\mathcal{P}, r(j), \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models K_a\varphi$ *iff for all* $s' \in \mathcal{S}$, $s \sim_a s' \Rightarrow (\mathcal{P}, s', \sigma) \models \varphi$
$(\mathcal{P}, s, \sigma) \models K_z\varphi$ *iff for all* $s' \in \mathcal{S}$, $s \sim_{\sigma(z)} s' \Rightarrow (\mathcal{P}, s', \sigma) \models \varphi$

A formula φ is *true* at s, or $(\mathcal{P}, s) \models \varphi$, if $(\mathcal{P}, s, \sigma) \models \varphi$ for all assignments σ; φ is *true* in $\mathcal{P}$, or $\mathcal{P} \models \varphi$, if $(\mathcal{P}, s_0) \models \varphi$ for all $s_0 \in I$. We remark that Def. 9 adopts an *active domain* semantics, where quantifiers range over the set $adom(s) \cup ag(s)$ of active individuals and agents. This is an extension to agents of the standard assumption in database theory, also used in data-aware systems [Belardinelli *et al.*, 2014; Hariri *et al.*, 2013]. Also, notice that the active domain may vary at each state. Furthermore, by definition of epistemic indistinguishability, if $a \notin ag(s)$ then $(\mathcal{P}, s, \sigma) \models K_a\varphi$, for all formulas φ, as for no $s' \in \mathcal{S}$, $s' \sim_a s$. In other words, epistemic formulas are vacuously true for agents not in the active domain of the state considered. So, for an epistemic formula not to be satisfied, it is required that an agent in the active domain does not know the fact in question.

Finally, we state the model checking problem for OMAS with respect to the specification language FO-CTLK$_x$.

Definition 10 (Model Checking Problem) *Given an OMAS $\mathcal{P}$ and an FO-CTLK$_x$ formula φ, determine whether for every initial state $s_0 \in I$, $(\mathcal{P}, s_0, \sigma_0) \models \varphi$ for some assignment σ_0.*

Def. 10 assumes that the transition function τ is given as a computable function, and that we have finitary descriptions for the set I of initial states and the domain X. These requirements are normally fulfilled in cases of interest (see Section 2.3). Moreover, the specification φ is typically an FO-CTLK$_x$ sentence, with no free variables. Hence, the model checking problem reduces to determine whether $\mathcal{P} \models \varphi$. Model checking general data-aware systems is known to be undecidable [Deutsch *et al.*, 2007]. In [Belardinelli *et al.*, 2012; 2014] this problem is proved decidable for *bounded* and *uniform* systems. However, all these contributions assume that the set of agents is fixed at design time. In [Belardinelli and Grossi, 2015] preliminary results on the verification of a particular class of OMAS are presented, but without considering the epistemic dimension.

2.3 Use Case: Knowledge in open MAS networks

We now illustrate the formalism introduced by means of an example on agent networks. In [Belardinelli and Grossi, 2015] it is shown how a non-probabilistic variant of the SIR network diffusion model (see [Jackson, 2008, Ch. 7]) can be formally verified against first-order, purely temporal specifications. In the SIR model a group of agents connected in a network structure goes through three different stages during an 'epidemic' involving the spread of diseases, ideas, information, or similar social phenomena. First, each agent is *susceptible* to be infected; she may actually get *infected* at a certain point depending on whether any of her neighbors in the network are also infected; then an agent will eventually *recover*. OMAS can be used to encode open and dynamic SIR models, also incorporating the epistemic aspects of diffusion. The specification language FO-CTLK$_x$ allows us to express properties of SIR models concerning: *(i)* how knowledge influences diffusion through the network; and *(ii)* how knowledge itself spreads within the system.

Let a binary predicate N denote the network structure, so $N(x, y)$ means that agents x and y are connected; while the unary predicates *Sus*, *Inf* and *Rec* denote the properties of being susceptible, infected, and recovered respectively. As examples of the first group of properties consider the following formulas:

$$AG\,\forall x, y(K_x(Inf(y) \wedge N(x, y)) \rightarrow AF\neg N(x, y)) \quad (2)$$
$$AG\,\forall x(K_x Sus(x) \rightarrow AFAG\forall y(N(x, y) \rightarrow Rec(y))) \quad (3)$$

Formula (2) states that it is always the case that if an agent x knows that she is connected to an infected agent y, then she will part at some point in the future. Formula (3) states that it is always the case that if an agent x knows she is susceptible, then eventually she will always be connected only to recovered agents.

We stress the fundamental difference between a quantified formula $\forall x K_x\phi$, which express *dynamically* the joint knowledge of ϕ for all active agents in a given state s, and the standard, *static* epistemic formula $E\phi = \bigwedge_{a \in Ag} K_a\phi$. Actually, for $E\phi$ to be a formula, the set Ag of agents has to be finite and specified at design time. Moreover, a formula such as $AG\,\forall x K_x\phi$ refers to the knowledge of a possibly different group of active agents at each time.

As examples of the second group of properties above consider the following formulas:

$$AG\, \forall x(Rec(x) \rightarrow AF\, \exists y K_y Rec(x)) \quad (4)$$

$$AG\, \forall y(Inf(y) \rightarrow (AF\, \forall x(N(x,y) \rightarrow K_x Inf(y)))) \quad (5)$$

Formula (4) states that it is always the case that if an agent is recovered, then this fact won't be ignored, i.e., someone will know it. Formula (5) states that it is always the case that if some agent y is infected, then all agents that are connected to y will eventually know this fact. We stress once more that to express epistemic properties, such as (2)-(5) above, in open MAS we do need epistemic modalities indexed by terms and quantification, as the set Ag of agents is infinite in general.

In the next section we develop techniques to model check OMAS against such first-order temporal-epistemic specifications.

3 Bisimulation

In Section 2 we stated that model checking OMAS against FO-CTLK$_x$ specifications is undecidable in general. To single out semantical fragments with a decidable model checking problem, we first introduce a notion of bisimulation and show that bisimilar OMAS satisfy the same FO-CTLK$_x$ formulas. The results presented in this section build upon previous work in the literature [Belardinelli *et al.*, 2012; 2014; Belardinelli and Grossi, 2015]. However, the present setting differs, as we consider open MAS, where agents can join and leave at run time, and our specification language contains term-indexed epistemic modalities.

In the rest of the paper we let $\mathcal{P} = \langle Ag, U, I, \tau \rangle$ and $\mathcal{P}' = \langle Ag', U', I', \tau' \rangle$ be OMAS and assume that $s = \langle l_0, \ldots, l_n \rangle \in \mathcal{S}$ and $s' = \langle l'_0, \ldots, l'_n \rangle \in \mathcal{S}'$. According to Def. 7 isomorphic states have the same relational structure, but to account also for values assigned to free variables we introduce the following notion.

Definition 11 (Equivalent assignments) *Given states $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$, and a formula ϕ, assignments $\sigma : Var \mapsto X$ and $\sigma' : Var \mapsto X'$ are* equivalent for ϕ (w.r.t. s and s') *iff for some bijection $\gamma : adom(s) \cup ag(s) \cup Con \cup \sigma(fr(\varphi)) \mapsto adom(s') \cup ag(s') \cup Con \cup \sigma'(fr(\varphi))$,* (i) *the restriction $\gamma|_{adom(s) \cup ag(s) \cup Con}$ is a witness for $s \simeq s'$; and* (ii) $\sigma'|_{fr(\varphi)} = \gamma \circ \sigma|_{fr(\varphi)}$.

Equivalent assignments preserve agent types, the (in)equalities in φ, as well as the active elements in s and s', *modulo* renaming.

Bisimulations are known to preserve the satisfaction of modal formulas in a propositional setting [Blackburn *et al.*, 2001, Ch. 2]. We now investigate under which conditions this is true of OMAS as well.

Definition 12 (Simulation) *A relation $R \subseteq \mathcal{S} \times \mathcal{S}'$ is a* simulation *iff $R(s, s')$ implies* (i) *$s \simeq s'$;* (ii) *for every $t \in \mathcal{S}$, if $s \rightarrow t$ then for some $t' \in \mathcal{S}'$, $s' \rightarrow t'$, $s \oplus t \simeq s' \oplus t'$, and $R(t, t')$; and* (iii) *for every $t \in \mathcal{S}$, $a \in ag(s)$, if $s \sim_a t$ then for some $t' \in \mathcal{S}'$, $s' \sim_a t'$, $s \oplus t \simeq s' \oplus t'$, and $R(t, t')$.*

A state s' *simulates* s iff $R(s, s')$ holds for some simulation R. In particular, similar states are isomorphic by condition 12.*(i)* above. Simulations can then be extended to bisimulations.

Definition 13 (Bisimulation) *A relation $B \subseteq \mathcal{S} \times \mathcal{S}'$ 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*, or $s \approx s'$, iff $B(s, s')$ holds for some bisimulation B. Notice that $\approx$ is the largest bisimulation and an equivalence relation on $\mathcal{S} \cup \mathcal{S}'$. Finally, the OMAS $\mathcal{P}$ and $\mathcal{P}'$ are *bisimilar*, or $\mathcal{P} \approx \mathcal{P}'$, iff *(i)* for every $s_0 \in I$, $s_0 \approx s'_0$ for some $s'_0 \in I'$, and *(ii)* for every $s'_0 \in I'$, $s_0 \approx s'_0$ for some $s_0 \in I$.

In [Belardinelli *et al.*, 2014] it is shown that, differently from the propositional modal case, in data-aware systems bisimilarity does not preserve first-order temporal-epistemic formulas. Nonetheless, we prove that *uniform* OMAS admit FO-CTLK$_x$-preserving bisimulations.

Definition 14 (Uniformity) *An OMAS $\mathcal{P}$ is* uniform *iff for every $s, t, s' \in \mathcal{S}$, $t' \in \mathcal{G}$,* (i) *if $s \rightarrow t$ and $s \oplus t \simeq s' \oplus t'$ then $s' \rightarrow t'$; and* (ii) *for every $a \in ag(s)$, if $s \sim_a t$ and $s \oplus t \overset{\iota}{\simeq} s' \oplus t'$ then $s' \sim_{\iota(a)} t'$.*

Intuitively, uniformity expresses a fullness condition on OMAS: a uniform OMAS allows all "isomorphic" transitions. We discuss uniformity in more depth in Section 3.

We finally prove that FO-CTLK$_x$ formulas cannot distinguish between bisimilar and uniform OMAS, as longs as specific cardinality constraints on the interpretation domains are satisfied.

Theorem 1 *Consider bisimilar and uniform OMAS $\mathcal{P}$ and $\mathcal{P}'$, bisimilar states $s \in \mathcal{S}$ and $s' \in \mathcal{S}'$, an FO-CTLK$_x$ formula φ, and assignments σ and σ' equivalent for φ w.r.t. s and s'. If*

1. *for every s-run r, for every $k \geq 0$,* (i) *$|X'| \geq |adom(r(k)) \cup ag(r(k)) \cup adom(r(k+1)) \cup ag(r(k+1)) \cup Con \cup \sigma(fr(\varphi))| + |var(\varphi) \setminus fr(\varphi)|$, and* (ii) *$|Ag'_T| \geq |ag_T(r(k)) \cup ag_T(r(k+1)) \cup Con \cup \sigma(fr(\varphi))| + |var(\varphi) \setminus fr(\varphi)|$ for every type T;*
2. *for every s'-run r', for every $k \geq 0$,* (i) *$|X| \geq |adom(r'(k)) \cup ag(r'(k)) \cup adom(r'(k+1)) \cup ag(r'(k+1)) \cup Con \cup \sigma'(fr(\varphi))| + |var(\varphi) \setminus fr(\varphi)|$, and* (ii) *$|Ag_T| \geq |ag_T(r'(k)) \cup ag_T(r'(k+1)) \cup Con \cup \sigma'(fr(\varphi))| + |var(\varphi) \setminus fr(\varphi)|$ for every type T;*

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

As a consequence of Theorem 1, bisimilar states satisfy the same FO-CTLK$_x$ formulas for equivalent assignments, whenever cardinality constraints (1) and (2) are satisfied.

We now apply Theorem 1 to the model checking problem for OMAS. First of all, we introduce bounded OMAS.

Definition 15 (Bounded OMAS) *An OMAS $\mathcal{P}$ is b-*bounded, *for $b \in \mathbb{N}$, iff for all $s \in \mathcal{S}$, $|adom(s) \cup ag(s)| \leq b$.*

An OMAS $\mathcal{P}$ is *bounded* iff it is b-bounded for some $b \in \mathbb{N}$. We remark that bounded OMAS are still infinite-state systems in general. Hereafter let $\sup_{s \in \mathcal{S}}\{|adom(s) \cup ag(s)|\}$ be equal to ∞ whenever the OMAS $\mathcal{P}$ is unbounded. Similarly for $\sup_{s \in \mathcal{S}}\{|ag_T(s)|\}$.

Corollary 2 *Consider bisimilar and uniform OMAS $\mathcal{P}$ and $\mathcal{P}'$, and an FO-CTLK$_x$ formula φ. If*

1. $|X'| \geq 2\sup_{s\in\mathcal{S}}\{|adom(s) \cup ag(s)|\} + |con(\phi)| + |var(\varphi)|$ *and* $|Ag'_T| \geq 2\sup_{s\in\mathcal{S}}\{|ag_T(s)|\}+|con(\phi)|+|var(\varphi)|$ *for every type* T*;*
2. $|X| \geq 2\sup_{s'\in\mathcal{S}'}\{|adom(s') \cup ag(s')|\} + |con(\phi)| + |var(\varphi)|$ *and* $|Ag_T| \geq 2\sup_{s'\in\mathcal{S}'}\{|ag_T(s')|\} + |con(\phi)| + |var(\varphi)|$ *for every type* T*;*

then $\mathcal{P} \models \varphi$ *iff* $\mathcal{P}' \models \varphi$.

Corollary 2 shows that an infinite-state OMAS $\mathcal{P}$ can in principle be verified by model checking a bisimilar system $\mathcal{P}'$, as long as X' is sufficiently large for $\mathcal{P}'$ to bisimulate $\mathcal{P}$.

In the next section we show that finite abstractions can indeed be defined for bounded OMAS, thus allowing for the verification of properties, such as (2)-(5) in Section 2.3, but first we briefly discuss uniform OMAS.

Discussion: Uniformity The notion of uniformity was put forward in [Belardinelli *et al.*, 2012] to prove the decidability of model checking data-aware systems. In [Belardinelli *et al.*, 2014] uniformity is related to *genericity* in databases [Abiteboul *et al.*, 1995]. Intuitively, in uniform systems trasitions depend on the logical form of data, rather than on the actual data content. It has been argued that the class of uniform systems covers most cases of interest [Hariri *et al.*, 2013; Deutsch *et al.*, 2007].

We now analyse uniformity in the context of OMAS through the following result.

Lemma 3 *Suppose that an OMAS* $\mathcal{P}$ *satisfies the following:*

for $s_0 \in \mathcal{S}$, $s'_0 \in \mathcal{S}'$, $s_0 \simeq s'_0$ *implies* $s_0 \in I$ *iff* $s'_0 \in I$ $\quad(\star)$

Then, $\mathcal{P}$ *is* uniform.

The proof of Lemma 3 makes essential use of conditions $(*)$ and $(+)$ on protocols and transition functions respectively. Thus, requirements *(i)* and *(ii)* in Def. 14 can be substituted by closure $(\star)$ of initial states under isomorphism. The latter condition is quite natural, as we are mainly interested in the relational structure of data, not the actual data content. In the final section we assume that all OMAS satisfy $(\star)$ and are therefore uniform.

3.1 Finite Abstraction

We now show that an infinite OMAS can in principle be verified by checking a finite abstraction. The main result is Theorem 5, which ensures that boundedness and closure under uniform initial states $(\star)$ are sufficient to obtain finite bisimilar abstractions, thus preserving FO-CTLK$_x$ formulas.

We first define the notion of abstract agents.

Definition 16 (Abstract agents) *Let* $a = \langle \mathcal{D}, Act, Pr\rangle \in Ag_T$ *be an agent of type* T *defined on a domain* $X = U \cup Ag$. *Given a set* $X' = U' \cup Ag'$ *of elements, the* abstract agent $a' \in Ag'_T$ *is the tuple* $\langle \mathcal{D}, Act, Pr'\rangle$ *on* X' *s.t.* Pr' *is the smallest function defined as*

- *if* $\alpha(\vec{u}) \in Pr(l)$, $l' \in \mathcal{D}'(X')$ *and* $l' \overset{\iota}{\simeq} l$, *then* $\alpha(\iota(\vec{u})) \in Pr'(l')$.

Given a set Ag_T of agents, let Ag'_T be the set of the corresponding abstract agents. Notice that Ag and Ag' are used to denote both the set of agent names and of agents; the context will disambiguate. The abstract agent a' in Def. 16 is indeed an agent of type T, as defined in Def. 4, since a and a' share the same database schema and actions. Moreover, protocol Pr' is well-defined whenever Pr is, and it satisfies condition $(*)$ on protocols by definition. We now present abstractions.

Definition 17 (Abstractions) *Let* $\mathcal{P} = \langle Ag, U, I, \tau\rangle$ *be an OMAS, and* Ag' *the set of abstract agents defined on* X' *as in Def. 16. The OMAS* $\mathcal{P}' = \langle Ag', U', I', \tau'\rangle$ *is an* abstraction *of* $\mathcal{P}$ *iff* (i) $I' = \{s'_0 \in \mathcal{G}' \mid s'_0 \simeq s_0$ *for some* $s_0 \in I\}$, *and* (ii) τ' *is the smallest function defined as follows*

- *if* $s \xrightarrow{\alpha(\vec{u})} t$ *in* $\mathcal{P}$, $s', t' \in \mathcal{G}'$, *and* $s \oplus t \simeq s' \oplus t'$ *for some witness* ι, *then* $s' \xrightarrow{\alpha(\iota(\vec{u}))} t'$.

The abstraction $\mathcal{P}'$ in Def. 17 is an OMAS as it complies with Def. 6. Moreover, condition $(+)$ on transition functions is satisfied. Notice that, by varying X' we can obtain abstractions of different cardinalities, in particular finite abstractions.

Next, we explore the relationship between an OMAS and its abstractions. By the next result every abstraction is uniform, independently from the concrete OMAS.

Lemma 4 *Every abstraction* $\mathcal{P}'$ *of an OMAS* $\mathcal{P}$ *is uniform. Moreover, if* $\mathcal{P}$ *is uniform and* $X' = X$, *then* $\mathcal{P}' = \mathcal{P}$.

By the next result there exists a bisimilar abstraction for every bounded OMAS, provided that the former is built over a sufficiently large domain. Hereafter we suppose that, for a bound $b \in \mathbb{N}$, N_b is the maximum numbers of parameters contained in any parametric joint actions, i.e., $N_b = b \cdot \max_{\{\alpha(\vec{x}) \in Act_T, type\ T\}}\{|\vec{x}|\}$.

Theorem 5 *Consider a bounded OMAS* $\mathcal{P}$ *over an infinite domain* X, *an FO-CTL*$_x$ *formula* φ, *and a domain* $X' \supseteq con(\varphi)$. *If* (i) $|X'| \geq 2b + |con(\varphi)| + \max\{|var(\varphi)|, N_b\}$, *and* (ii) *for every type* T, $|Ag'_T| \geq 2b + |con(\varphi)| + \max\{|var(\varphi)|, N_b\}$, *then there exists a bisimilar abstraction* $\mathcal{P}'$ *of* $\mathcal{P}$ *over* X'. *In particular,* $\mathcal{P} \models \varphi$ *iff* $\mathcal{P}' \models \varphi$.

Notice that each Ag'_T and X' in Theorem 5 might as well be finite. So, by using a sufficient number of abstract agents and values, we can in principle reduce the model checking problem for infinite-state OMAS to the verification of a finite abstraction. Specifically, we obtain the following corollary to Theorem 5.

Corollary 6 *Given a bounded OMAS* $\mathcal{P}$ *over an infinite domain* X, *and an FO-CTL*$_x$ *formula* φ, *there exists an abstract OMAS* $\mathcal{P}'$ *over a* finite *domain* X' *s.t.* φ *holds in* $\mathcal{P}$ *iff it holds in* $\mathcal{P}'$.

As a consequence of Corollary 6, we can in principle verify an infinite-state, bounded OMAS, by model checking its finite, bisimilar abstraction.

4 Conclusions

In this paper we tackled an ongoing problem in the formal verification of multi-agent systems, namely the verification of open MAS where agents may enter and leave the system at run time. A notable feature of our proposal concerns the rich specification language FO-CTLK$_x$, which includes epistemic operators indexed by individual terms. As we discussed, the latter are key to express relevant properties of OMAS. Further, we analysed the model checking problem within this setting, showing that it can be addressed through finite bisimilar abstractions, under some natural conditions.

An open problem not tackled in the present contribution and left for future work is the development of methodologies for generating finite abstractions, so that effective model checking procedures can be provided. This is a major challenge for the verification of open MAS.

References

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

[Belardinelli and Grossi, 2015] Francesco Belardinelli and Davide Grossi. On the formal verification of diffusion phenomena in open dynamic agent networks. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS15)*, 2015.

[Belardinelli and Lomuscio, 2012] Francesco Belardinelli and Alessio Lomuscio. Interactions between Knowledge and Time in a First-Order Logic for Multi-Agent Systems: Completeness Results. *Journal of Artificial Intelligence Research*, 45:1–45, 2012.

[Belardinelli and Lomuscio, 2013] Francesco Belardinelli and Alessio Lomuscio. Decidability of model checking non-uniform artifact-centric quantified interpreted systems. In Rossi [2013].

[Belardinelli *et al.*, 2012] Francesco Belardinelli, Alessio Lomuscio, and Fabio 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)*, pages 319 – 328, 2012.

[Belardinelli *et al.*, 2014] Francesco Belardinelli, Fabio Patrizi, and Alessio Lomuscio. Verification of agent-based artifact systems. *Journal of Artificial Intelligence Research*, 51:333–77, 2014.

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

[Deutsch *et al.*, 2007] Alin Deutsch, Liying Sui, and Victor Vianu. Specification and Verification of Data-Driven Web Applications. *J. Comput. Syst. Sci.*, 73(3):442–474, 2007.

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

[Gammie and van der Meyden, 2004] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *Proceedings of 16th International Conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.

[Hariri *et al.*, 2013] B. Bagheri Hariri, D. Calvanese, G. De Giacomo, A. Deutsch, and M. Montali. Verification of relational data-centric dynamic systems with external services. In R. Hull and W. Fan, editors, *PODS*, pages 163–174. ACM, 2013.

[Jackson, 2008] M. O. Jackson. *Social and Economic Networks.* Princeton University Press, 2008.

[Kouvaros and Lomuscio, 2013a] Panagiotis Kouvaros and Alessio Lomuscio. Automatic verification of parameterised multi-agent systems. In Maria L. Gini, Onn Shehory, Takayuki Ito, and Catholijn M. Jonker, editors, *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 861–868. IFAAMAS, 2013.

[Kouvaros and Lomuscio, 2013b] Panagiotis Kouvaros and Alessio Lomuscio. A cutoff technique for the verification of parameterised interpreted systems with parameterised environments. In Rossi [2013].

[Lomuscio and Colombetti, 1996] Alessio Lomuscio and Marco Colombetti. QLB: A quantified logic for belief. In Jörg P. Müller, Michael Wooldridge, and Nicholas R. Jennings, editors, *Intelligent Agents III, Agent Theories, Architectures, and Languages, ECAI '96 Workshop (ATAL), Budapest, Hungary, August 12-13, 1996, Proceedings*, volume 1193 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 1996.

[Lomuscio *et al.*, 2009] A. Lomuscio, H. Qu, and F. Raimondi. Mcmas: A model checker for the verification of multi-agent systems. In A. Bouajjani and O. Maler, editors, *CAV*, volume 5643 of *Lecture Notes in Computer Science*, pages 682–688. Springer, 2009.

[Montali *et al.*, 2014] Marco Montali, Diego Calvanese, and Giuseppe De Giacomo. Verification of data-aware commitment-based multiagent system. In Ana L. C. Bazzan, Michael N. Huhns, Alessio Lomuscio, and Paul Scerri, editors, *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 157–164. IFAAMAS/ACM, 2014.

[Parikh and Ramanujam, 1985] R. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In *Logic of Programs*, pages 256–268, 1985.

[Penczek and Lomuscio, 2003] Wojciech Penczek and Alessio Lomuscio. Verifying Epistemic Properties of Multi-agent Systems via Bounded Model Checking. *Fundamenta Informaticae*, 55(2):167–185, 2003.

[Rodríguez-Aguilar *et al.*, 1998] Juan A. Rodríguez-Aguilar, Francisco J. Martín, Pablo Noriega, Pere Garcia, and Carles Sierra. Towards a test-bed for trading agents in electronic auction markets. *AI Communications*, 11(1):5–19, 1998.

[Rossi, 2013] Francesca Rossi, editor. *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013.

[Wooldridge, 2001] Michael Wooldridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., 2001.