A symmetry reduction technique for model checking temporal-epistemic logic

Mika Cohen[1] Mads Dam[2] Alessio Lomuscio[1] Hongyang Qu[1]

1. Department of Computing, Imperial College London, UK

2. School of Computer Science and Communication, KTH, Sweden

Abstract

We introduce a symmetry reduction technique for model checking temporal-epistemic properties of multi-agent systems defined in the mainstream interpreted systems framework. The technique, based on counterpart semantics, aims to reduce the set of initial states that need to be considered in a model. We present theoretical results establishing that there are neither false positives nor false negatives in the reduced model. We evaluate the technique by presenting the results of an implementation tested against two well known applications of epistemic logic, the muddy children and the dining cryptographers. The experimental results obtained confirm that the reduction in model checking time can be dramatic, thereby allowing for the verification of hitherto intractable systems.

1 Introduction

Model checking [3] is a well-established automated technique for verifying reactive systems against design requirements expressed in temporal logics. More recently, model checking has been extended to multi-agent systems (MAS) and design requirements specified in temporal-epistemic logic (cf. [1, 13, 19, 20, 22]).

However, a major obstacle to model checking is the so called *state explosion problem*: Model checking becomes intractable for systems with many agents, as the state space that needs to be constructed grows exponentially in the number variables used by agents in the system.

Even worse, for many popular applications of epistemic logic, the state space explosion is exasperated by an explosion of initial states; each model includes one initial state for every possible initial configuration of local states. For example, in voting protocols, card games, the muddy children scenario and the dining cryptographers protocol, we need to consider a different initial state for every distribution of votes/cards/mud/coin tosses and choice of payer. The explosion of initial states leads to an explosion of *reachable* states.

However, in these applications several agents can be seen as simple variations of a common generic agent. As a result, execution traces starting from different initial states might be the same up to a rearrangement of local states; the traces are symmetric. Even so, removing some initial states makes properties of the initial configuration

common knowledge: If proposition ϕ holds in all initial states that remain, it is common knowledge among agents that initially ϕ holds. So, by removing initial states, we obtain unwanted epistemic validities: A more sophisticated treatment is necessary.

In this paper, we present a technique for exploiting symmetry when model checking temporal-epistemic properties of multi-agent systems defined in the mainstream *interpreted systems* framework [12]. The key idea is to abstract the standard Kripke semantics into a counterpart semantics [15], by permuting agent names along the indistinguishability relation. This allows the initial states to be reduced by keeping a single representative for a group of initial states that are equivalent up to the order of local states. The abstraction is accurate in that the same temporal-epistemic specifications hold for both the original and the reduced models; there are neither false positives nor false negatives in the reduced model.

Experiments on the muddy children and the dining cryptographers − standard showcases for epistemic logic − confirm that the reduction in model checking time can be dramatic. The time is reduced by one to two orders of magnitude for moderate numbers of agents. Thus, the reduction technique permits the verification of hitherto intractable systems made up of many identical agents. Such systems are a characteristic peculiarity of several prototypical applications of epistemic logic, including communication protocols, cache coherence protocols, network protocols, voting protocols, authentication protocols and games.

Related work *Abstraction* is a family of techniques for simplifying large models by removing details inessential to the property to be verified [5]. While abstraction of reactive systems for temporal properties is an active research area, abstraction for epistemic properties has received little attention. Recently, [8, 11] abstract Kripke models for epistemic logic by approximating the epistemic possibility relation. However, the models are not computationally grounded, which hampers concrete applications [21]. In [7], interpreted systems are abstracted by collapsing local states and actions of each agent.

Closer to our contribution, *component symmetry reduction* is an abstraction technique for temporal logic aiming to reduce the state space by collapsing system states that are equivalent up to a reordering of local states into one representative state [4, 10]. There has so far been no attempt in the literature at extending symmetry reduction to epistemic logic. Most closely related to our work are the automata-based symmetry reduction techniques that annotate the reduced transition relation with process permutations [9]. The annotation allows processes to be tracked along the transitions, accommodating atomic propositions that specify properties of individual processes (and so break symmetry).

Counterpart semantics has been used before in a computationally grounded setting for epistemic logic [6], but without any relation to state space reduction.

Overview of paper The rest of the paper is organized as follows. In Section 2, we recall definitions for interpreted systems and the temporal-epistemic specification logic CTLK. In Section 3, we instantiate (component) symmetry to interpreted systems and define formula symmetry for CTLK. In Section 4, we present the counterpart semantics

used on the reduced system, and prove that CTLK validity is invariant between the original and reduced systems. In Section 5, we show how to detect symmetries. In Section 6, we report on experimental results. Finally, Section 7 concludes.

2 Interpreted systems and CTLK

We model multi-agent systems in the *interpreted systems* framework [12] and express requirements in the temporal-epistemic logic CTLK [18]; this section summarizes the basic definitions.

Consider a set $Ag = \{1...n\}$ of agents. For each agent i, assume a non-empty set L_i of local states that agent i can be in, and a non-empty set ACT_i of actions that agent i can perform. Assume also a non-empty set L_E of states for the environment and a non-empty set ACT_E of actions. Let $S = L_1 \times \cdots \times L_n \times L_E$ be the set of possible global states and $ACT = ACT_1 \times \cdots \times ACT_n \times ACT_E$ the set of joint actions. For each agent i, assume a local protocol $P_i : L_i \longrightarrow 2^{ACT_i}$ selecting actions depending on the current local state of i, and a local evolution function $t_i : ACT \times L_i \longrightarrow L_i$ specifying how agent i evolves from one local state to another depending on its action, the actions of the other agents, and the action of the environment. Analogously, assume an environment protocol $P_E : L_E \longrightarrow 2^{ACT_E}$, and assume an environment evolution function $t_E : ACT \times L_E \longrightarrow L_E$. Let $P = \langle P_1, \cdots, P_n, P_E \rangle$ be the joint protocol and $t = \langle t_1, \cdots, t_n, t_E \rangle$ be the joint evolution function. Finally, assume a non-empty set $I_0 \subseteq S$ of initial states, and an evaluation function $V : S \longrightarrow 2^A$, for some non-empty set A of propositional atoms.

Definition 2.1. *An interpreted system is a tuple $\mathcal{I} = \langle S, ACT, P, t, I_0, V \rangle$ with a global state space S, a joint actions space ACT, a joint protocol P, a joint evolution function t, a set I_0 of initial states and an evaluation function V.*

For any global state $g = \langle l_1, \ldots, l_n, l_E \rangle \in S$, we write $l_i(g)$ for the the local state l_i of agent i in g, and $l_E(g)$ for the environment state l_E in g.

The local protocols and the local evolution functions together determine how the system of agents proceeds from one global state to the next: The global transition relation in $\mathcal{I}$ is the relation $R \subseteq S \times S$ such that $\langle g, g' \rangle \in R$ if and only if, there exists $\overline{a} = \langle a_1, \ldots, a_n, a_E \rangle \in ACT$ such that $t_E(\overline{a}, l_E(g)) = l_E(g')$ and $a_E \in P_E(l_E(g))$ and for all agent i, $t_i(\overline{a}, l_i(g)) = l_i(g')$ and $a_i \in P_i(l_i(g))$. We assume through out the paper that the global transition relation R is serial, i.e., for every $g \in S$, there is $g' \in S$ such that gRg'.

A path in $\mathcal{I}$ is an infinite sequence $g^0, g^1, \ldots$ of global states in S such that every pair of adjacent states forms a transition, i.e., $g^j R g^{j+1}$ for all j. The set G of reachable states in $\mathcal{I}$ contains all global states $g \in S$ for which there is a path $g^0, g^1, \ldots, g, \ldots$ starting from an initial state $g^0 \in I_0$.

Intuitively, the local state $l_i(g)$ contains all the information available to agent i. We say that $g' \in G$ is epistemically possible for agent i at $g \in G$, written $g \sim_i g'$, if the local state of i in g is the local state of i in g', i.e., $l_i(g) = l_i(g')$. We refer to [12] for more details.

We consider specifications expressed in the temporal-epistemic logic CTLK, which extends Computation Tree Logic (CTL) with epistemic modalities.

Definition 2.2. *Assume a set $Ag = \{1..n\}$ of agents and a non-empty set A of propositional atoms p. $CTLK$ formulae are defined by the following BNF expresion:*

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid K_i\phi \mid EX\phi \mid EG\phi \mid E(\phi U \phi)$$

where $i \in Ag$.

The knowledge modality K_i is read "Agent i knows that", the quantifier E is read "For some computation path" and the temporal operators X, G and U are read "In the next state", "Always" and "Until" respectively. We assume customary abbreviations: $\overline{K}_i$ encodes the diamond modality $\neg K_i \neg$; $AG\,\phi$ represents $\neg E(true\,U\,\neg\phi)$ ("For all paths, always ϕ"); $AF\,\phi$ abbreviates $\neg EG\,\neg\phi$ ("For all paths, eventually ϕ").

Given an interpreted system $\mathcal{I}$, the modality K_i is interpreted by the epistemic possibility relation $\sim_i$ as follows:

- $(\mathcal{I}, g) \models K_i\phi$ iff $(\mathcal{I}, g') \models \phi$ for all g' such that $g \sim_i g'$

The CTL modalities are interpreted by means of the global transition relation R in the usual way: $(\mathcal{I}, g) \models EX\phi$ iff for some path $g^0, g^1, \ldots$ in $\mathcal{I}$ such that $g = g^0$, we have $(\mathcal{I}, g^1) \models \phi$; $(\mathcal{I}, g) \models EG\phi$ iff for some path $g^0, g^1, \ldots$ in $\mathcal{I}$ such that $g = g^0$, we have $(\mathcal{I}, g^i) \models \phi$ for all $i \geq 0$; $(\mathcal{I}, g) \models E(\phi U \phi')$ iff for some path $g^0, g^1, \ldots$ in $\mathcal{I}$ such that $g = g^0$, there is a natural number i such that $(\mathcal{I}, g^i) \models \phi'$ and $(\mathcal{I}, g^j) \models \phi$ for all $0 \leq j < i$.

We write $|\phi|$ for the extension of formula ϕ in $\mathcal{I}$, i.e., the set of reachable states $g \in G$ such that $(\mathcal{I}, g) \models \phi$. We say that formula ϕ is true in system $\mathcal{I}$, written $\mathcal{I} \models \phi$, iff $I_0 \subseteq |\phi|$.

Example 2.3 (Muddy children)**.** *[12] A group of n children have been out playing in the mud, and k of them now have mud on their forehead. They sit in a circle and see only the foreheads of other children. Their father announces "At least one of you is muddy!", and then asks: "Does any one of you know whether you are muddy?". The father repeats the question over and over. After each question, the children answer simultaneously; all children answer "I do not know" until round k when the muddy children announce that they know they are muddy.*

The above scenario can be modeled as an interpreted system $\mathcal{I}$ (see [12] for details). We would like to verify − by means of a model checker − that the announcement of each child is truthful:

$$\bigwedge_i AG(saysknow(i) \leftrightarrow (K_i mud(i) \vee K_i \neg mud(i))) \tag{1}$$

Example 2.4 (Dining cryptographers)**.** *[2] A group of n cryptographers share a meal around a circular table. Either one of them paid for the meal or their employer did. They would like to make it known if one of them paid without revealing the identity of the payer (if one of them did pay). To this end, every cryptographer tosses a coin and shows the outcome to his right-hand neigbour. Comparing his own coin to the*

coin shown to him, each cryptographer announces if the two coins agree or not; if a cryptographer paid for the meal, he announces the opposite of what he sees.

The scenario can be modeled as an interpreted system (see [17] for details). We would like to verify that no agent i can ever come to know that a particular other agent j is the payer:

$$\bigwedge_{i \neq j} AG\, \neg K_i\, paid(j) \tag{2}$$

and each agent eventually comes to know whether or not the employer paid:

$$\bigwedge_{i} AF\, (K_i\, epaid \vee K_i\, \neg epaid) \tag{3}$$

3 Symmetry

In this section, we instantiate *agent (component) symmetry* [4, 10] to interpreted systems, and generalize *formula symmetry* [9] from CTL to CTLK.

Informally, an agent symmetry is an interchange of local states preserving the behaviors of the system. Formally, a *symmetry*, or *automorphism*, of a set $X \subseteq S$ of global states, or of a relation $X \subseteq S \times S$ between global states, is a bijection $\pi : S \longrightarrow S$ that leaves X unchanged: $\pi(X) = X$, where $\pi(X) = \{\pi(g) : g \in X\}$ if X is a set, and $\pi(X) = \{\langle \pi(g), \pi(g') \rangle : \langle g, g' \rangle \in X\}$ if X is a relation. An automorphism of the system $\mathcal{I}$ is an automorphism of both the global transition relation R and the set I_0 of initial states.

In general, one can allow permutations to affect data as well as control. In this paper, however, we consider only *agent symmetries*: automorphisms induced by a reordering of local states in global states. We identify a reordering of local states with an agent permutation, i.e., a bijection $\pi : Ag \longrightarrow Ag$. This agent permutation π extends naturally to global states by moving each local state $l_i(g)$ of agent i to that of agent $\pi(i)$: $\pi(g) = \langle l_{\pi^{-1}(1)}(g), \ldots, l_{\pi^{-1}(n)}(g), l_E(g) \rangle$. For example, $p(\langle A, A, B, B, l_E(g) \rangle) = \langle B, A, A, B, l_E(g) \rangle$ if π is the right-shift permutation defined by $\pi(i) = i + 1$ modulo 4. The set $Aut(X)$ of agent permutations that are automorphisms of $X \in \{R, I_0, \mathcal{I}\}$ is a group with respect to function composition, function inverse and identity. We write $Sym(Ag)$ for the group of all agent permutations.

In some applications, if local states are structured terms in which variables appear, we might want the agent permutations to act on variables inside the local states being reshuffled: $\pi(g) = \langle \pi(l_{\pi^{-1}(1)}(g)), \ldots, \pi(l_{\pi^{-1}(n)}(g)), \pi(l_E(g)) \rangle$. The state reduction technique presented in this paper does generalize to symmetries obtained under this finer definition of $\pi(g)$, but we only discuss the former definition for ease of presentation, given that the formal results are equivalent.

We assume that propositional atoms have the form $p(\bar{i})$, where p is an *agent predicate*, and $\bar{i} \in Ag^\star$ respects the arity of p. Intuitively, $p(\bar{i})$ states that property p holds of agents $\bar{i}$. To reflect this, we require of any evaluation function V that:

$$p(\bar{i}) \in V(g) \quad \Leftrightarrow \quad p(\pi(\bar{i})) \in V(\pi(g)) \tag{4}$$

for any agent permutation π and any states $g, \pi(g) \in S$. For example, the interpretation of the unary agent predicate mud must satisfy: $g \models mud(i)$ if and only if $\pi(g) \models mud(\pi(i))$.

We apply the agent permutation π on formula ϕ by substituting $\pi(i)$ for agent name i inside the formula.

Definition 3.1. *$\pi(\phi)$ is defined inductively over ϕ as follows: $\pi(p(\bar{i})) = p(\pi(\bar{i}))$, $\pi(\phi \wedge \phi') = \pi(\phi) \wedge \pi(\phi')$, $\pi(\neg\phi) = \neg\pi(\phi)$, $\pi(K_i\,\phi) = K_{\pi(i)}\pi(\phi)$, $\pi(EX\,\phi) = EX\,\pi(\phi)$, etc.*

Applying an agent permutation to a CTL formula modifies only atomic sub-formulae. Here, by contrast, an agent permutation modifies epistemic modalities as well. For example, if π transposes 1 and 2, i.e., $\pi(1) = 2$ and $\pi(2) = 1$, then, $\pi(K_1\,mud(2)) = K_2\,mud(1)$.

An automorphism of a formula ϕ is an agent permutation π such that ϕ is semantically equivalent to $\pi(\phi)$, i.e., $|\phi| = |\pi(\phi)|$ in all systems $\mathcal{I}$. For example, an agent permutation that transposes 1 and 2 is an automorphism of $K_1\,mud(2) \wedge K_2\,mud(1)$, and every agent permutation is an automorphism of specifications (1), (2) and (3) (cf. lemma 5.3). As before, the set $Aut(\phi)$ of automorphisms of ϕ is a group.

4 Agent symmetry reduction

As highlighted in the introduction, multi-agent systems in key prototypical applications of epistemic logic exhibit considerable agent symmetry. In this section, we present a technique that exploits agent symmetries to reduce the initial states in the system.

Assume a group $\Gamma \subseteq Aut(\mathcal{I})$ of symmetries of the system $\mathcal{I}$. (See Section 5 for how to compute Γ). Let $\mathcal{I}/\Gamma = \langle S, ACT, P, t, I'_0, V\rangle$ be the result of replacing the set I_0 of initial states with a minimal $I'_0 \subseteq I_0$ such that $I_0 = \Gamma(I'_0) = \{\pi(g) \mid \pi \in \Gamma, g \in I'_0\}$, i.e., the result of leaving a single representative initial state g for each equivalence class $\{\pi(g) \mid \pi \in \Gamma\}$ of initial states.

Example 4.1. *For the system $\mathcal{I}$ of muddy children from Example 2.3, we can choose Γ such that $\mathcal{I}/Sym(Ag)$ contains a single initial state where 1 child is muddy, a single initial state where 2 children are muddy, etc. While $\mathcal{I}$ has 2^n initial states, $\mathcal{I}/Sym(Ag)$ has only $n+1$ initial states.*

We show that the original system $\mathcal{I}$ and the reduced system $\mathcal{I}/\Gamma$ can be seen to validate the same CTLK formulae. To achieve this invariance, we use an abstract semantics for the epistemic modalities in $\mathcal{I}/\Gamma$, a semantics that abstracts the Kripke semantics of Section 2 into a counterpart semantics.

4.1 Abstract semantics

In counterpart semantics [15], the modal accessibility relation is indexed by a correlation between individuals: If a state g' is accessible from a state g under a correlation C, then C relates an individual i at state g to its "counterpart" $C(i)$ at state g'. In our abstract semantics, individuals are agents and correlations are agent permutations.

Definition 4.2. *Assume an interpreted system $\mathcal{I}$ and a group Γ of agent permutations. A state $g' \in G$ is (epistemically) accessible to agent i from state $g \in G$ under agent permutation $\pi \in \Gamma$, written $g \approx_i^\pi g'$, if and only if, $g \sim_i \pi^{-1}(g')$.*

Thus, $g \approx_i^\pi g'$ if the local state of i in g is the local state of i in g' after applying π^{-1}. Intuitively, if $g \approx_i^p g'$ then each agent j in g is the counterpart of agent $\pi(j)$ in the accessible state g'. At state g, therefore, agent i holds fact $\phi(j)$ as possible if the counterpart fact $\phi(\pi(j))$ holds at state g'.

Definition 4.3 (Abstract semantics)**.** *Truth of ϕ at g in interpreted system $\mathcal{I}$ under Γ, written $(\mathcal{I}, g) \models_\Gamma \phi$, is defined inductively by:*

- $(\mathcal{I}, g) \models_\Gamma K_i\phi$ *iff* $(\mathcal{I}, g') \models_\Gamma \pi(\phi)$ *for all* $g' \in G$ *and* $\pi \in \Gamma$ *such that* $g \approx_i^\pi g'$

All other cases are unchanged from Section 2.

Note that when Γ contains only the identity permutation, the abstract semantics coincides with the basic Kripke semantics of Section 2: $(\mathcal{I}, g) \models \phi$, if and only if, $(\mathcal{I}, g) \models_{\{id\}} \phi$.

Example 4.4. *Consider the system $\mathcal{I}$ of muddy children from Example 2.3. Let $\mathcal{I}'$ be the result of removing the initial state g^1 where only child 1 is muddy. Observe that under the standard semantics, at the initial state g^0 where no child is muddy, child 1 knows she is not muddy: $(\mathcal{I}', g^0) \models K_1 \neg mud(1)$. In fact, since we removed initial state g^1, the only state reachable in $\mathcal{I}'$ which is epistemically possible for child 1 at g^0 is g^0 itself. By contrast, under the abstract semantics, child 1 does not know she is not muddy: $(\mathcal{I}', g^0) \not\models_{Sym(Ag)} K_1 \neg mud(1)$. This is seen as follows. By assumption, $\mathcal{I}'$ contains some initial state g^2 where exactly one child, say child 2, is muddy: $g^2 \models mud(2)$. Choose an agent permutation π that transposes 1 and 2. Then, $\pi^{-1}(g^2) = g^1$, i.e., $g^0 \sim_1 \pi^{-1}(g^2)$, i.e., $g^0 \approx_1^\pi g^2$. The claim follows by definition 4.3, since $g^2 \models \pi(mud(1))$.*

4.2 Reduction theorem

We reach the reduction result by way of two lemmas. First, applying a system automorphism π on the possibility relation for agent i yields the possibility relation for agent $\pi(i)$.

Lemma 4.5. *If $\pi \in Aut(\mathcal{I})$, then $\pi(\sim_i) = \sim_{\pi(i)}$.*

Proof. (Sketch) $l_i(g) = l_i(g')$ iff $l_{\pi(i)}(\pi(g)) = l_{\pi(i)}(\pi(g'))$. But, $g, g' \in G$ iff $\pi(g), \pi(g') \in G$, since $\pi \in Aut(\mathcal{I})$. Thus, $g \sim_i g'$, if and only if, $\pi(g) \sim_{\pi(i)} \pi(g')$. □

Secondly, applying a system automorphism π to the extension of a formula ϕ produces the extension of $\pi(\phi)$.

Lemma 4.6. *If $\pi \in Aut(\mathcal{I})$, then $\pi(|\phi|) = |\pi(\phi)|$.*

Proof. (Sketch) We show that $(\mathcal{I}, g) \models \phi$, if and only if, $(\mathcal{I}, \pi(g)) \models \pi(\phi)$, by induction on ϕ. Base step: From requirement (4). Induction step, temporal modalities: Since $\pi \in Aut(\mathcal{I})$, gRg', if and only if, $\pi(g)R\pi(g')$. Induction step, epistemic modalities: From lemma 4.5. □

Theorem 4.7 (Reduction)**.** *Let Γ be a subgroup of both $Aut(\mathcal{I})$ and $Aut(\phi)$. Then, $\mathcal{I} \models \phi$ if and only if $\mathcal{I}/\Gamma \models_\Gamma \phi$.*

Proof. (Sketch) Since $\Gamma \subseteq Aut(\mathcal{I})$, it follows that $G = \{\pi(g) \mid \pi \in \Gamma, g \in G'\}$, where G and G' are the sets of reachable states in $\mathcal{I}$ and $\mathcal{I}/\Gamma$ respectively. Therefore, we can evaluate the epistemic modality in $\mathcal{I}$ by scanning the reduced space G' and apply agent permutations "on the fly", expanding each state g' into its equivalence class $\{\pi^{-1}(g') \mid \pi \in \Gamma\}$. So, $(\mathcal{I}, g) \models K_i\phi$, if and only if, $\forall g' \in G' : \forall \pi \in \Gamma : g \sim_i \pi^{-1}(g') \Rightarrow (\mathcal{I}, \pi^{-1}(g')) \models \phi$. In fact, by lemma 4.6, we can replace the test of the property ϕ at $\pi^{-1}(g')$ with the test of the counterpart property $\pi(\phi)$ at g', and so obtain: $(\mathcal{I}, g) \models K_i\phi$, if and only if, $\forall g' \in G' : \forall \pi \in \Gamma : g \sim_i \pi^{-1}(g') \Rightarrow (\mathcal{I}, g') \models \pi(\phi)$. But $g \sim_i \pi^{-1}(g')$ means $g \approx_i^\pi g'$. By induction over ϕ, therefore, we obtain: $(\mathcal{I}, g) \models \phi$, if and only if, $(\mathcal{I}/\Gamma, g) \models_\Gamma \phi$, for all $g \in G'$. The theorem follows since $\Gamma \subseteq Aut(\phi)$. □

Following Reduction Theorem 4.7, we can reduce the initial states before feeding a system to a model checker, provided the model checker implements the abstract semantics.

Example 4.8. *Applying the Reduction Theorem 4.7 to the system $\mathcal{I}$ of muddy children from Example 2.3, $\mathcal{I} \models (1)$, if and only if, $\mathcal{I}/Sym(Ag) \models_{Sym(Ag)} (1)$.*

5 Symmetry detection

To apply Reduction Theorem 4.7, we need to know a group Γ of symmetries of the system $\mathcal{I}$ and of the specification ϕ. In this section, we show how to compute a group Γ directly from the description of $\mathcal{I}$ and the shape of ϕ.

Intuitively, system symmetries arise from similarities in the agents. To exploit this intuition, we extend the notion of symmetry to joint protocols and joint evolution functions.

Definition 5.1. *An automorphism of the joint protocol P is an agent permutation π such that $P_i = P_{\pi(i)}$. An automorphism of the joint evolution function t is an agent permutation π such that $t_i(l, \overline{a}) = t_{\pi(i)}(l, \pi(\overline{a}))$ and $t_E(l, \overline{a}) = t_E(l, \pi(\overline{a}))$, where $\pi(\overline{a}) = \langle a_{\pi^{-1}(1)}, \ldots, a_{\pi^{-1}(n)}, a_E \rangle$.*

It can be checked that the sets $Aut(P)$ and $Aut(t)$ of automorphisms of the joint protocol P and the joint evolution function t are groups.

We can now make precise the intuition that symmetries arise from similarities in the agents: Any symmetry of the joint protocol and the joint evolution function is a symmetry of the global transition relation R.

Lemma 5.2. $Aut(P) \cap Aut(t) \subseteq Aut(R)$.

Proof. Let $\langle g, g' \rangle \in R$ and $\pi \in Aut(P) \cap Aut(t)$. So there is $\overline{a}$ such that $t_i(\overline{a}, l_i(g)) = l_i(g')$ with $a_i \in P_i(l_i(g))$, and so $t_{\pi(i)}(\pi(\overline{a}), l_i(g)) = l_i(g')$ with $a_i \in P_{\pi(i)}(l_i(g))$. So, $t_{\pi(i)}(\pi(\overline{a}), l_{\pi(i)}(\pi(g))) = l_{\pi(i)}(\pi(g'))$ with $\pi(a)_{\pi(i)} \in P_{\pi(i)}(l_{\pi(i)}(\pi(g)))$, i.e., $t_i(\pi(\overline{a}), l_i(\pi(g))) = l_i(\pi(g'))$ and $\pi(a)_i \in P_i(l_i(\pi(g)))$, i.e., $\langle \pi(g), \pi(g') \rangle \in R$. □

Using lemma 5.2, we can detect automorphisms of the global transition relation R without explicitly constructing R.

Turning next to the detection of formula symmetries, we observe that the specifications we are interested are often *universal* formulae of the form $\bigwedge_{\overline{j}} \phi(\overline{j}/\overline{i})$, where $\overline{i}$ is the tuple of agent names appearing in ϕ, and $\overline{j}$ ranges over tuples of distinct agent names. In other words, universal formulae either have the form $\bigwedge_i \phi(i)$, or the form $\bigwedge_{i \neq j} \phi(i, j)$, etc. For example, (1), (2) and (3) are universal. According to our second lemma, universal formulae are fully symmetric.

Lemma 5.3. *If ϕ is universal then $Aut(\phi) = Sym(Ag)$.*

Proof. Since ϕ is equivalent to $\bigwedge_\pi \pi(\phi')$, for some ϕ'. □

Proposition 5.4 (Detection)**.** *$Aut(P) \cap Aut(t) \cap Aut(I_0)$ is a subgroup of $Aut(\mathcal{I})$ and $Aut(\phi)$, if ϕ is universal.*

Proof. From lemmas 5.2 and 5.3. □

In other words, the group $\Gamma = Aut(P) \cap Aut(t) \cap Aut(I_0)$ can be used to form the reduced system $\mathcal{I}/\Gamma$ in Reduction Theorem 4.7 when verifying a universal specification ϕ.

6 Implementation and Experiments

To show the effectiveness of the proposed reduction technique, we implemented the abstract semantics from Section 4.1 on MCMAS, a publicly available model checker for multi-agent systems [16]. Using *symbolic model checking via OBDD's*, MCMAS can verify CTL, ATL, epistemic and deontic formulae with fairness constraints and produce counterexamples/witness executions. Its input language ISPL adopts the interpreted systems in Section 2 as the underlying semantics. From the public release source of MCMAS, we modified the interpretation of the epistemic modality as follows:

function computing $|\overline{K}_i \phi|$:
  $X = \emptyset$
  for π in Γ:
    $X = X \cup PreImage(\approx_i^\pi, |\pi(\phi)|)$
  return $X \cap G$

where G is the set of states reachable in the supplied system (i.e., the reduced system $\mathcal{I}/\Gamma$). Note that the loop-body computes the extension of $\pi(\phi)$ rather than ϕ. Of

course, when $\pi, \pi' \in \Gamma$ agree on the agent names in the formula $\overline{K}_i\phi$, the loop need not consider both of them. Since the algorithm may revisit the same sub-formula several times, we store constructed extensions in a BDD cache.

We tested our reduction technique on the dining cryptographers (Example 2.4) and the muddy children (Example 2.3) on a machine running Linux Fedora 10 x86_64 version with Intel Core 2 Duo E4500 2.2GHz and 4GB memory. In the ISPL program for the dining cryptographers, every rotation (sideways shift $\pi(i) = i + k$ modulo n, for some k) is an automorphism of the joint protocol, joint evolution function and the initial states. By Proposition 5.4 and Theorem 4.7, we can thus form the ISPL program for a reduced system by requiring (in the program section defining the initial states) that either agent 1 pays or no agent pays, thereby obtaining a portion $2/(n + 1)$ of initial states of the original ISPL program. Verification times (in seconds) for the universal anonymity specification (2) and different number of cryptographers are presented in Table 1, which shows we obtained a linear reduction. The running time for 11 cryptographers is less than for 10 because the BDDs generated in the former are more compact. This is a known feature of symbolic techniques for this example [14].

Table 1: Verification results for the dining cryptographers

Number of cryptographers	Without reduction		With reduction	
	States	Time	States	Time
9	15,360	2	3072	1
10	33,792	12	6,144	3
11	73,728	4	12,288	2
12	159,744	13	24,576	7
13	344,064	35	49,152	16
14	737,280	302	98,304	9

In the ISPL program for the muddy children, every agent permutation is an automorphism of the joint protocol, joint evolution function and the initial states. By Proposition 5.4 and Theorem 4.7, we can simplify the program section defining the initial states by keeping only one initial state from each group of initial states with the same number of muddy children. Table 2 shows the verification results for the universal specification (1); we obtain an exponential reduction in verification time. We compare the actual BDD memory usage (in MB) because (1) it is difficult to get the exact number of reachable states due to the special BDD encoding for integer (a state can have multiple copies in the BDD representing reachable states); (2) it is interesting to examine the memory consumption, since the reduction generates extra BDDs.

7 Conclusion

While model checking epistemic logic has received considerable attention in the AI community, there has been little work so far on techniques for tackling the state explosion problem. But, if model checking for MAS and AI logics is to succeed, it is

Table 2: Verification results for the muddy children

Number of Children	Without reduction		With reduction	
	BDD memory	Time	BDD memory	Time
7	14	2	11	1
8	20	7	13	1
9	22	18	14	3
10	46	50	16	4
11	47	112	27	6
12	56	360	32	9
13	52	305	25	11
14	59	595	29	16
15	87	2602	32	17
16	64	2082	38	37

essential that attention is dedicated to this.

In this paper, we have introduced an agent symmetry reduction technique for MAS specified in temporal-epistemic logic. By means of this technique, we abstract the standard interpreted systems semantics into a counterpart semantics, by permuting agent names along the epistemic accessibility relation. This allows the initial states to be reduced by keeping a single representative for a group of symmetric initial states. The abstraction technique is shown to be sound and complete, i.e., there are neither false positives nor false negatives in the reduced model.

Experiments with the muddy children and the dining cryptographers show substantial reductions in verification time under symbolic model checking: The time is reduced by one to two orders of magnitude for moderate numbers of agents. Since BDDs are a form of symbolic optimization, one may expect even bigger reductions for any implementation on top of an explicit model checker.

Looking ahead, we intend to transfer the reduction technique to data symmetry and to automate the symmetry detection method in Section 5 for ISPL programs. Automation of symmetry detection is a relatively recent area in mainstream verification with significant open problems. However, the very structured nature of ISPL might prove helpful.

References

[1] R. H. Bordini, M. Fisher, C. Pardavila, and M. Wooldridge. Model checking AgentSpeak. In J. S. Rosenschein, T. Sandholm, W. Michael, and M. Yokoo, editors, *AAMAS-03*, pages 409–416. ACM Press, 2003.

[2] D. Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[3] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs Workshop*, pages 52–71, London, UK, 1982. Springer-Verlag.

[4] E. M. Clarke, R. Enders, T. Filkorn, and S. Jha. Exploiting symmetry in temporal logic model checking. *Form. Methods Syst. Des.*, 9(1-2):77–104, 1996.

[5] E. M. Clarke, O. Grumberg, and D. E. Long. Model checking and abstraction. *ACM Trans. Program. Lang. Syst.*, 16(5):1512–1542, 1994.

[6] M. Cohen and M. Dam. A complete axiomatization of knowledge and cryptography. In *LICS'07*, pages 77–88. IEEE Computer Society, 2007.

[7] M. Cohen, M. Dam, A. Lomuscio, and F. Russo. Abstraction in model checking multi-agent systems. In *AAMAS-09*, 2009.

[8] F. Dechesne, S. Orzan, and Y. Wang. Refinement of kripke models for dynamics. In *ICTAC'08*, pages 111–125. Springer, 2008.

[9] E. A. Emerson and A. P. Sistla. Utilizing symmetry when model checking under fairness assumptions: An automata-theoretic approach. In *CAV'95*, pages 309–324, London, UK, 1995. Springer-Verlag.

[10] E. A. Emerson and A. P. Sistla. Symmetry and model checking. *Form. Methods Syst. Des.*, 9(1-2):105–131, 1996.

[11] C. Enea and C. Dima. Abstractions of multi-agent systems. In *CEEMAS'07*, pages 11–21, 2007.

[12] R. Fagin, J. Y. Halpern, M. Y. Vardi, and Y. Moses. *Reasoning about knowledge*. MIT Press, Cambridge, MA, USA, 1995.

[13] P. Gammie and R. van der Meyden. MCK: Model checking the logic of knowledge. In *CAV'04*, volume 3114 of *LNCS*, pages 479–483. Springer-Verlag, 2004.

[14] M. Kacprzak, A. Lomuscio, A. Niewiadomski, W. Penczek, F. Raimondi, and M. Szreter. Comparing bdd and sat based techniques for model checking chaum's dining cryptographers protocol. *Fundam. Inf.*, 72(1-3):215–234, 2006.

[15] David Lewis. Counterpart theory and quantified modal logic. *Journal of Philosophy*, 65:113–126, 1968.

[16] A. Lomuscio, F. Raimondi, and H. Qu. MCMAS: A model checker for multi-agent systems *http://dfn.dl.sourceforge.net/sourceforge/ist-contract/mcmas-0.9.6.2.tar.gz*.

[17] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. In *CSFW '04*, page 280, Washington, DC, USA, 2004. IEEE Computer Society.

[18] R. van der Meyden and K. S. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.

[19] W. Nabialek, A. Niewiadomski, W. Penczek, A. Pólrola, and M. Szreter. VerICS 2004: A model checker for real time and multi-agent systems. In *CS&P'04*, pages 88–99. Humboldt University, 2004.

[20] F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. *Journal of Applied Logic*, 5(2):235–251, 2007.

[21] M. Wooldridge. Computationally grounded theories of agency. In E. Durfee, editor, *ICMAS*, pages 13–22. IEEE Press, 2000.

[22] M. Wooldridge, M. Fisher, M. Huget, and S. Parsons. Model checking multiagent systems with mable. In *AAMAS-02*, pages 952–959, Bologna, Italy, 2002.