Group Synthesis for Parametric Temporal-Epistemic Logic

Andrew V. Jones
Department of Computing
Imperial College London
andrew.jones@ic.ac.uk

Michał Knapik
ICS PAS
mknapik@ipipian.waw.pl

Alessio Lomuscio
Department of Computing
Imperial College London
a.lomuscio@ic.ac.uk

Wojciech Penczek
ICS PAS and
UPH Siedlce
penczek@ipipan.waw.pl

ABSTRACT

We investigate parameter synthesis in the context of temporal-epistemic logic. We introduce CTLPK, a parametric extension to the branching time temporal-epistemic logic CTLK with free variables representing groups of agents. We give algorithms for automatically synthesising the groups of agents that make a given parametric formula satisfied. We discuss an implementation of the technique on top of the open-source model checker MCMAS and demonstrate its attractiveness by reporting the experimental results obtained.

Categories and Subject Descriptors

D.2.4 [**Software/Program Verification**]: Model Checking

General Terms

Verification

Keywords

Temporal-Epistemic Logic, Model Checking

1. INTRODUCTION

Multi-agent systems (MAS) are distributed systems in which components, or agents, interact with one another trying to reach private or common goals. One of the recent topics of interest in this area is the issue of verification and validation of MAS, i.e., how to ascertain whether a given MAS satisfies certain specifications of interest. In this context a number of model checkers [5, 7, 9] have been developed to verify logics for MAS, including epistemic, deontic, and strategic logics.

Of particular interest to the community is work on automated model checking tailored to temporal-epistemic specifications. In this line specifications of MAS are defined on temporal languages augmented with modalities to reason about the knowledge of the agents in the system. As an example of this, most coordination protocols require common knowledge to be obtained within the group of agents before the protocol can be executed by the MAS [4]. Common knowledge (and other group modalities such as distributed knowledge [4]) are expressed in temporal-epistemic logic by using indices representing the groups they refer to. Model checkers such as MCMAS [9] and MCK [5] already support specifications with group operators including common knowledge and can be used to verify such properties in a given system.

There are scenarios, however, when checking knowledge for a specific group of agents is not sufficient. For example, in a MAS that implements distributed diagnosis we, as specifiers, would actually like to know *which groups* in the system obtain distributed knowledge of a particular fault in the system. Moreover, even if we have an intuition as to whether a particular group reaches common or distributed knowledge of a particular property, it is of interest to ascertain whether there is a maximal or minimal group that obtains this so that, for instance, we can minimise the number of agents involved in a coordination protocol.

If the set of agents is finite, we can solve this problem by repeatedly querying a model checker with all the possible instantiations of the specification for all possible valuations. However, if this set is large, its power set will not be of a trivial size, resulting in a large number of checks. If the specification of interest involves several, not necessarily equal groups, the number of instantiations grows exponentially. In symbolic model checking the bottleneck is normally the computation of the set of reachable states, but if the number of formulae to be checked is sufficiently high, the verification for the formulae can become the most time consuming operation.

The aim of this paper is to explore an alternative, potentially more efficient technique to identify (or *synthesise*) the groups of agents for which a given temporal-epistemic specification holds. We call this *parametric model checking for temporal-epistemic logic* due to its clear correspondence to parametric model checking for temporal specifications [8, 13], where temporal intervals are synthesised. Concisely, in the approach presented the groups under synthesis are treated as variables ranging over subsets of the set of all agents. The model checking algorithm we put forward returns only the subsets validating a given formula.

The rest of the paper is as follows. The syntax and semantics of CTLPK is introduced in the next section. Section 3 presents the synthesis algorithms for the verification of the parametric formulae. In Section 4 we use an experimental implementation to demonstrate results for parameter synthesis for the dining cryptographers protocol and diagnosability properties for a commonly used network protocol. Finally, in Section 5, we conclude.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

2. THE LOGIC CTLPK

In this section we introduce Parametric Computation Tree Logic with Knowledge (CTLPK, for short), a branching time temporal-epistemic logic that includes free variables as parameters for the group modalities. Intuitively, any CTLPK formula ϕ represents the set of formulae that can be constructed from ϕ by instantiating the parameters in ϕ with any non-empty set of agents in the group considered.

Definition 1 CTLPK syntax. Let $\mathcal{PV}$ be a set of *propositions*, *Groups* be a finite set of *group variables*, and $Agents = \{1, \ldots, n\}$ be a finite set of *agents*. The grammar of CTLPK in BNF is given below.

$$\begin{aligned}\phi \quad ::= \quad & p \mid \neg\phi \mid \phi \vee \phi \mid EX\phi \mid EG\phi \mid E\phi U\psi \mid \\ & K_i\phi \mid E_\Gamma\phi \mid D_\Gamma\phi \mid C_\Gamma\phi \mid K_Y\phi \mid E_Y\phi \mid D_Y\phi\end{aligned}$$

where $p \in \mathcal{PV}$, $i \in Agents$, $\Gamma \subseteq Agents$, $\Gamma \neq \emptyset$ and $Y \in Groups$.

A formula containing at least one group variable is said to be *parametric* while a formula with no group variable is called *ground*. Notice that the set of the ground formulae defines CTLK [11].

Recall that EX, EG, EU are temporal modalities, where E stands for "there exists a path", and X, G and U respectively mean "at the next state", "for all successor states" and "until". The epistemic modalities K_i, E_Γ, D_Γ, C_Γ are interpreted as follows: $K_i\phi$ stands for "agent i knows ϕ"; E_Γ is read as "everyone in group Γ knows ϕ"; and $D_\Gamma\phi$ ($C_\Gamma\phi$) stands for "the group Γ has distributed (common, respectively) knowledge of ϕ". Let $AF\phi \overset{def}{=} \neg EG\neg\phi$ be a derived temporal modality. The formula $AF\phi$ expresses: "for all paths eventually ϕ holds".

As an example, consider the formula $\phi = AF(D_Y fault)$, where Y is a group variable, and $Agents = \{a, b\}$. The formula ϕ represents the set of ground formulae $\{AF(D_{\{a\}}fault)$, $AF(D_{\{b\}}fault)$, $AF(D_{\{a,b\}}fault)\}$. The parametric formula above states that "for all paths eventually the agents in Y will have distributed knowledge of *fault*". Since ϕ contains a free variable, similar to first-order logic, an interpretation for ϕ in a model is an assignment υ from the variable Y to concrete instances in $2^{\{a,b\}} \setminus \{\emptyset\}$.

Given that variables in CTLPK represent groups, a natural question to ask is *"which groups satisfy the temporal-epistemic specification ϕ on a given model M?"*. In the following we provide an efficient method for calculating all interpretations υ, such that all ground formulae constructed from ϕ by replacing the variable Y with $\Gamma \in \upsilon(Y)$ are satisfied in M.

Before we do so, we provide the semantics for CTLPK in terms of the interpreted systems formalism [4], a standard semantics for epistemic logic. We assume that each agent i in *Agents* is defined by means of a set of local states L_i, actions Act_i and a protocol $P_i : L_i \to 2^{Act_i}$. The environment is analogously defined by L_e, Act_e, and P_e. The set of global states $G \subseteq L_1 \times \cdots \times L_n \times L_e$ is a subset of the Cartesian product of the local states for the agents and the environment. The transitions are defined locally from local states on joint actions by considering evolution functions $\tau_i : L_i \times Act_1 \times \cdots \times Act_n \times Act_e \to L_i$. We refer to [4] for more details.

Definition 2 Interpreted Systems. Given a set of agents *Agents*, an *interpreted system* (or *model*) M is a tuple $M = (G, g^0, T, \sim_1, \ldots, \sim_n)$ such that:

- $G \subseteq L_1 \times \cdots \times L_n \times L_e$ is the set of *reachable* global states for the system, where $g^0 \in G$.
- The transition relation $T \subseteq G \times G$ is defined by $(g, g') \in T$ if there exists $(act_1, \ldots, act_n, act_e) \in Act_i \times \cdots \times Act_n \times Act_e$ such that $\tau_i(l_i(g), act_1, \ldots, act_n, act_e) = l_i(g')$ for all $i \in Agents$ and $\tau_e(l_e(g), act_1, \ldots, act_n, act_e) = l_e(g')$, where $l_i(g)$ returns the local state of agent i in the global state g. The actions $act_1, \ldots, act_n, act_e$ are all consistent with their respective protocols, i.e., $act_i \in P_i(l_i(g))$ and analogously for the environment.
- For any $i \in Agents$ the relation $\sim_i \subseteq G \times G$ is an epistemic accessibility relation such that $g \sim_i g'$ iff $l_i(g) = l_i(g')$, where $l_i(g)$ is as above.
- The function $\mathcal{L} : G \to 2^{\mathcal{PV}}$ is an interpretation for a set of the propositions $\mathcal{PV}$.

Given a concrete group Γ of agents, we introduce three group accessibility relations as follows:

$$\sim_\Gamma^E = \bigcup_{i\in\Gamma} \sim_i, \quad \sim_\Gamma^D = \bigcap_{i\in\Gamma} \sim_i, \quad \sim_\Gamma^C = \left(\sim_\Gamma^E \right)^+,$$

where $^+$ denotes the transitive closure. These relations are used to interpret the ground group modalities.

We now introduce the notion of a *path* as a sequence $\pi = (g_0, g_1, \ldots)$ such that $g_i \in G$ and $(g_i, g_{i+1}) \in T$ for all $i \geq 0$. We denote $\pi(j) = g_j$ for all $j \geq 0$.

In the definition of the semantics of the CTLPK formulae, we use a valuation of the group variables $\upsilon : Groups \to 2^{Agents} \setminus \{\emptyset\}$. The set of all the valuations of the group variables is denoted by *GroupVals*. Formally

$$GroupVals = \left(2^{Agents} \setminus \{\emptyset\}\right)^{Groups}.$$

Now we are in the position to introduce the semantics of CTLPK. If ϕ is a formula of CTLPK, then by $M, g \models_\upsilon \phi$ we denote that ϕ holds in the state g of the model M given the valuation υ. We omit the model symbol M, where this does not lead to ambiguity.

Definition 3 CTLPK semantics. Let M be a model and υ be a valuation of the group variables. The relation $\models_\upsilon$ is defined recursively as follows:

- $g \models_\upsilon p$ iff $p \in \mathcal{L}(g)$ for $p \in \mathcal{PV}$,
- $g \models_\upsilon \neg\phi$ iff $g \not\models_\upsilon \phi$,
- $g \models_\upsilon \phi \vee \psi$ iff $g \models_\upsilon \phi$ or $g \models_\upsilon \psi$,
- $g \models_\upsilon EX\phi$ iff there exists a path π starting at g, such that $\pi(1) \models_\upsilon \phi$,
- $g \models_\upsilon EG\phi$ iff there exists a path π starting at g, such that $\pi(i) \models_\upsilon \phi$ for all $i \geq 0$,
- $g \models_\upsilon E\psi U\phi$ iff there exists a path π starting at g, such that $\pi(j) \models_\upsilon \phi$ for some $j \geq 0$, and $\pi(i) \models_\upsilon \psi$ for all $0 \leq i < j$,
- $g \models_\upsilon K_i\phi$ iff for all $g' \in G$ if $g \sim_i g'$, then $g' \models_\upsilon \phi$,
- $g \models_\upsilon Z_\Gamma\phi$ iff for all $g' \in G$ if $g \sim_\Gamma^Z g'$, then $g' \models_\upsilon \phi$, where $Z \in \{E, D, C\}$,
- $g \models_\upsilon K_Y\phi$ iff $g \models_\upsilon K_i\phi$, where $\{i\} = \upsilon(Y)$,
- $g \models_\upsilon Z_Y\phi$ iff $g \models_\upsilon Z_{\upsilon(Y)}\phi$, where $Z \in \{E, D, C\}$.

We say that ϕ holds in a model M under valuation υ (denoted $M \models_\upsilon \phi$) if $M, g^0 \models_\upsilon \phi$.

Note that if ϕ is a ground formula (i.e., a CTLK formula), then $M, g \models_\upsilon \phi$ does not depend on the choice of υ.

3. GROUP SYNTHESIS FOR CTLPK

Given the semantics of CTLPK, a question that arises is, given a formula ϕ, how to determine all valuations υ for the group variables in ϕ such that all and only the ground instances of ϕ, obtained by substituting the group variables with concrete group values, are satisfied on the given model.

Formally, given a formula ϕ and a model M, we define a function $f_\phi : G \to 2^{GroupVals}$, returning at every global state the set of valuations for all the group variables to concrete group values such that $M, g \models_\upsilon \phi$ iff $\upsilon \in f_\phi(g)$.

In the rest of this section we present the algorithm $Synth_\phi$, a provably sound and complete procedure for constructing f_ϕ. The procedure is applied recursively bottom-up from the atoms to the out-most operators and is defined inductively. The correctness of the overall result follows from the correctness of the procedure for each individual modality.

In the following examples, we only consider formulae that contain a single group variable. We write $\{Y \rightsquigarrow \{A_1, \ldots, A_m\}\}$ to represent the set of valuations $\{\upsilon_1, \ldots, \upsilon_m\}$ s.t. $\upsilon_i(Y) = A_i$, for all $1 \le i \le m$.

Throughout this section we assume that the model M is fixed.

3.1 Boolean Operations and Non-parametric Modalities

We begin by defining the function f_ϕ for CTLPK modalities which do not contain group variables (i.e., the operators from CTLK).

Atomic Propositions.

For an atomic proposition $p \in \mathcal{PV}$, the function f_p is defined as follows:

$$f_p(g) = \begin{cases} GroupVals & \text{if } p \in \mathcal{L}(g), \\ \emptyset & \text{otherwise.} \end{cases}$$

For an atomic proposition, $M, g \models_\upsilon p$ does not depend upon υ; therefore for a state g such that $p \in \mathcal{L}(g)$, $f_p(g)$ is simply the set of all possible group valuations.

Example 1 (Atomic Propositions). Consider the simple three state, two-agent interpreted system model as illustrated in Figure 1. In this model we have $\mathcal{PV} = \{p, q\}$ where $\mathcal{L}(w_0) = \mathcal{L}(w_1) = \{p\}$ and $\mathcal{L}(w_2) = \{q\}$. Dashed lines labelled with an agent index represent the indistinguishability relation for that agent; solid lines labelled with 't' represent temporal transitions.

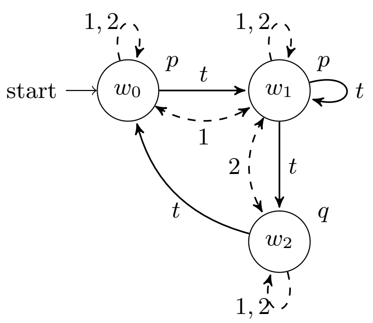

Figure 1: The interpreted system of Examples 1–5 and 7.

In this two-agent example with only one group variable we have that $GroupVals = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1, 2\}\}\}$. It is straightforward to observe that:

$$f_p(g) = \begin{cases} \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1, 2\}\}\} & \text{for } g \in \{w_0, w_1\}, \\ \emptyset & \text{if } g = w_2. \end{cases}$$

Negation.

Given f_ϕ, to construct the function $f_{\neg\phi}$ we complement the groups defined in f_ϕ:

$$f_{\neg\phi}(g) = GroupVals \setminus f_\phi(g)$$

As per the definition of f_ϕ, we have that for each state g and valuation of group variables $\upsilon \in GroupVals$ we have $M, g \models_\upsilon \phi$ iff $\upsilon \in f_\phi(g)$. This is equivalent to the fact that for each state g and valuation $\upsilon \in GroupVals$ we have that $M, g \not\models_\upsilon \phi$ iff $\upsilon \notin f_\phi(g)$, which in turn is equivalent to $\upsilon \in GroupVals \setminus f_\phi(g)$, i.e., $\upsilon \in f_{\neg\phi}(g)$.

In presented algorithms we assume the existence of a subroutine, denoted *Complement*, realising the above operation, i.e., $Complement(f_\phi) = f_{\neg\phi}$.

Example 2 (Negation). Consider the model from Example 1. After computing the complement of f_p we obtain:

$$f_{\neg p}(g) = \begin{cases} \emptyset & \text{for } g \in \{w_0, w_1\}, \\ \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1, 2\}\}\} & \text{if } g = w_2. \end{cases}$$

Disjunction.

For each $\phi, \psi \in$ CTLPK, we have that for each $g \in G$, $f_{\phi\vee\psi}(g) = f_\phi(g) \cup f_\psi(g)$. It can easily be seen that $s \models_\upsilon \phi \vee \psi$ iff $s \models_\upsilon \phi$ or $s \models_\upsilon \psi$. This is equivalent to $\upsilon \in f_\phi(s)$ or $\upsilon \in f_\psi(s)$.

Temporal Operators.

Intuitively, a temporal transition does not alter the group assignments in f_ϕ that hold at a given successor. Therefore, for $EX\phi$ we take the union of all assignments for ϕ from each next state:

$$f_{EX\phi}(g) = \bigcup_{\{g' \in G \mid (g, g') \in T\}} f_\phi(g').$$

We now consider the case of $EG\phi$. Note that $f_{EG\phi}(g) = f_\phi(g) \cap f_{EXEG\phi}(g)$, for each $g \in G$. Therefore, in a similar way to the non-parametric case [2, 6], $f_{EG\phi}$ can be obtained through a fixed-point calculation.

Similarly, for $E\phi U\phi$ we have that $f_{E\phi U\psi}(g) = f_\psi(g) \cup (f_\phi(g) \cap f_{EXE\phi U\psi}(g))$. Again, such a calculation can be performed as a fixed-point.

Example 3 (Temporal Operators). Let us consider the formula $EXE_Y p$ that is to be evaluated at the initial state w_0 of model presented in Figure 1. Let us also assume that the function $f_{E_Y p}$ has been correctly calculated and is as follows (we illustrate its construction in Example 4):

- $f_{E_Y p}(w_0) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1, 2\}\}\}$
- $f_{E_Y p}(w_1) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{E_Y p}(w_2) = \emptyset$

From the construction of $EX\phi$ (where $\phi = E_Y p$) presented previously we have that:

- $f_{EXE_Yp}(w_0) = f_{E_Yp}(w_1) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{EXE_Yp}(w_1) = f_{E_Yp}(w_1) \cup f_{E_Yp}(w_2) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{EXE_Yp}(w_2) = f_{E_Yp}(w_0) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}$

Finally, we evaluate f_{EXE_Yp} at the state w_0 and obtain the single satisfiable valuation of $\upsilon = \{Y \rightsquigarrow \{\{1\}\}\}$. Such an assignment would correspond to the concrete formula $EXE_\Gamma p$, where $\Gamma = \{1\}$. This formula can easily be seen to hold at w_0.

Epistemic Operators.

The set of assignments that make $K_i\phi$ hold at g is equal to the intersection of all sets of assignments that make ϕ hold at any state i-distinguishable from g. Therefore, given the formula $K_i\phi \in$ CTLPK and a global state $g \in G$ we have:

$$f_{K_i\phi}(g) = \bigcap_{\{g' \in G \mid g \sim_i g'\}} f_\phi(g').$$

For the operators E_Γ, D_Γ and C_Γ we follow the same construction, but use the relations $\sim_\Gamma^E$, $\sim_\Gamma^D$, $\sim_\Gamma^C$ respectively.

3.2 Group Synthesis for Parametric Epistemic Operators

We now present a methodology for synthesising group assignments for the parametric epistemic modalities. As is common in techniques for calculating the satisfaction of epistemic formulae (see [12]), the following algorithms rely on the use of the dual for the associated modality being evaluated.

We begin by presenting the synthesis technique for the parametric "everybody knows" modality ($E_Y\phi$). As individual knowledge is a special case of everybody knows (i.e., where the set Γ for E_Γ consists of a single agent i), we delay this presentation until after $E_Y\phi$.

Synthesis for Everybody Knows.

To calculate the parametric assignments for $E_Y\phi$, we need to define the *existential group pre-image* between two global states g and g'. We denote this as $Link_Y^\exists(g, g')$ and it consists of all group valuations υ such that there exists an $i \in \upsilon(Y)$ and $g \sim_i g'$. Formally:

$$Link_Y^\exists(g, g') = \{\upsilon \in GroupVals \mid g \sim_i g' \text{ for some } i \in \upsilon(Y)\}.$$

It is important to note that $\sim_i$ is an equivalence relation, thus from its reflexivity we have that $Link_Y^\exists(g, g) = GroupVals$ and from its symmetry $Link_Y^\exists(g, g') = Link_Y^\exists(g', g)$ follows.

Algorithm 1 $Synth_E\,(f_\phi, Y)$

```
Input: $f_\phi \in \left(2^{GroupVals}\right)^G$
Output: $f_{E_Y\phi} \in \left(2^{GroupVals}\right)^G$
1: $f := Complement\,(f_\phi)$
2: $h := \emptyset$
3: for all $g \in G$ do
4:     $h(g) := \bigcup_{g' \in G} \left(Link_Y^\exists\,(g, g') \cap f\,(g')\right)$
5: end for
6: return  $Complement\,(h)$
```

Lemma 4 *(Correctness of $Synth_E$). Let $M = (G, g^0, T, \sim_1, \dots, \sim_n)$ be an interpreted system, $g \in G$, and $\phi \in$ CTLPK. Then, $M, g \models_\upsilon E_Y\phi$ iff $\upsilon \in Synth_E(f_\phi, Y)(g)$.*

Proof. Using the definition of $Link_Y^\exists$ and the inductive assumption on f_ϕ, we prove correctness of $Synth_E\,(f_\phi, Y)$. At the end of the loop between Lines 3 and 5 in Algorithm 1, for each global state $g \in G$, the set $h(g)$ consists of all the assignments υ such that there exists $i \in \upsilon(Y)$ and there exists $g' \in G$ where $g \sim_i g'$ and $M, g' \models_\upsilon \neg\phi$.

As such, $\upsilon \in h(g)$ iff $M, g \models_\upsilon \overline{E}_Y\neg\phi$, where $\overline{E}_Y\phi$ is defined as $\neg E_Y\neg\phi$. By taking the complement of h, we obtain the set of all assignments υ, which map each global state g to a set of valuations of group variables, such that:

$$\begin{aligned} & M, g \not\models_\upsilon \overline{E}_Y\neg\phi \text{ (def. of complement)} \\ \Leftrightarrow\ & M, g \models_\upsilon \neg\overline{E}_Y\neg\phi \text{ (def. of } \not\models) \\ \Leftrightarrow\ & M, g \models_\upsilon E_Y\phi \text{ (def. of } \overline{E}_Y). \end{aligned}$$

□

Example 4 (Everybody Knows). We now show that f_{E_Yp} from Example 3 is correctly synthesised. At the first line of Algorithm 1, we take the complement of f_p (i.e., $f_{\neg p}$) and store it in the variable f. As such, the function held in the variable f contains the assignments $f(w_0) = f(w_1) = \emptyset$ and $f(w_2) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}$ (see Example 2).

We now show the construction of $Link_Y^\exists(g, g')$ for all pairs of states. These are shown below (we omit the symmetric cases):

$$\begin{aligned} Link_Y^\exists(w_i, w_i) &= \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}, \text{ for } i \in \{0,1,2\} \\ Link_Y^\exists(w_0, w_1) &= \{Y \rightsquigarrow \{\{1\}, \{1,2\}\}\} \\ Link_Y^\exists(w_1, w_2) &= \{Y \rightsquigarrow \{\{2\}, \{1,2\}\}\} \\ Link_Y^\exists(w_0, w_2) &= \emptyset \end{aligned}$$

At Line 4, we use $Link_Y^\exists(g, g')$ and $f(g')$ to construct $h(g)$. Therefore we need to compute $Link_Y^\exists(g, g') \cap f(g')$ for all pairs of states $g, g' \in \{w_0, w_1, w_2\}$. Notice that if $g' \in \{w_0, w_1\}$ then $f(g') = \emptyset$. Similarly, if $g = w_0$ and $g' = w_2$ then $Link_Y^\exists(g, g') = \emptyset$. This leaves only two remaining, non-empty cases:

$$\begin{aligned} Link_Y^\exists(w_1, w_2) \cap f(w_2) &= \{Y \rightsquigarrow \{\{2\}, \{1,2\}\}\} \\ Link_Y^\exists(w_2, w_2) \cap f(w_2) &= \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\} \end{aligned}$$

Next, at Line 4, we calculate the function h by assigning to each $h(g)$ the union of $Link_Y^\exists(g, g') \cap f(g')$ for all $g' \in \{w_0, w_1, w_2\}$. The function h is therefore as follows:

- $h(w_0) = \emptyset$
- $h(w_1) = \{Y \rightsquigarrow \{\{2\}, \{1,2\}\}\}$
- $h(w_2) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}$

Finally, we take the complement of h to calculate the function f_{E_Yp}. The result can be seen below:

- $f_{E_Yp}(w_0) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}$
- $f_{E_Yp}(w_1) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{E_Yp}(w_2) = \emptyset$

These assignments mean that E_Yp can be satisfied at the state w_0 with any assignment to the variable Y and at the state w_1 with the assignment $Y = \{1\}$. However, there is no substitution that makes E_Yp hold at w_2.

Synthesis of Individual Knowledge.

As previously stated, individual knowledge is a special case of everybody knows, where each group assignment consists of only a single agent. We adapt the set $Link^{\exists}_Y$ as below:

$$Link^{\text{indv.}}_Y(g,g') = \left\{ v \in GroupVals \mid v(Y) = \{i\} \text{ and } g \sim_i g' \right\}.$$

The set $Link^{\text{indv.}}_Y$ contains only those group valuations that assign to Y a single agent. To compute $f_{K_Y\phi}$, we use Algorithm $Synth_K$; this algorithm can be simply constructed by substituting $Link^{\exists}_Y$ with $Link^{\text{indv.}}_Y$ on Line 4 of Algorithm 1.

Corollary 5 ***(Correctness of*** $Synth_K$***).***
Let $M = (G, g^0, T, \sim_1, \ldots, \sim_n)$ be an interpreted system, $g \in G$, and $\phi \in CTLPK$. Then, $M, g \models_v K_Y\phi$ iff $v \in Synth_K(f_\phi, Y)(g)$.

The proof is straightforward and can easily be obtained by replacing $\overline{E}_Y$ with $\overline{K}_Y$ in the proof of Lemma 4.

Example 5 (Individual Knowledge). We demonstrate how to construct $f_{K_Y p}$ for the model in Example 3. The only difference between $Synth_K$ and Algorithm 1 is that we substitute $Link^{\exists}_Y(g,g')$ for $Link^{\text{indv.}}_Y(g,g')$. The latter consists of all group valuations $v \in Link^{\exists}_Y(g,g')$ such that $v(Y)$ is a single-element group.

Using the values for $Link^{\exists}_Y(g,g')$ from Example 4, and selecting only those group assignments consisting of a single agent, we obtain:

$$Link^{\text{indv.}}_Y(w_1,w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{2\}\}\}$$
$$Link^{\text{indv.}}_Y(w_2,w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{1\},\{2\}\}\}$$

where $Link^{\text{indv.}}_Y(g,g') = \emptyset$ for the remaining cases.

Taking the union of $Link^{\text{indv.}}_Y(g,g') \cap f(g')$ for all $g, g' \in G$, and then complementing, we obtain the function $f_{K_Y p}$ as:

- $f_{K_Y p}(w_0) = \{Y \rightsquigarrow \{\{1\},\{2\}\}\}$
- $f_{K_Y p}(w_1) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{K_Y p}(w_2) = \emptyset$

It can easily be seen that for all states $g \in G$ the set $f_{K_Y p}(g)$ consists of exactly those valuations from $f_{E_Y p}(g)$ that assign to Y groups consisting of one agent only.

Synthesis of Distributed Knowledge.

Given two global states $g, g' \in G$ and a group $Y \in Groups$, we define:

$$Link^{\forall}_Y(g,g') = \{v \in GroupVals \mid g \sim_i g' \text{ for all } i \in v(Y)\}.$$

In contrast to $Link^{\exists}_Y(g,g')$, the set $Link^{\forall}_Y(g,g')$ consists of all assignments v, such that for all agents $i \in v(Y)$, i considers g and g' as indistinguishable (i.e., the group Y considers g and g' indistinguishable).

Algorithm 2 $Synth_D\ (f_\phi, Y)$

```
Input: $f_\phi \in \left(2^{GroupVals}\right)^G$
Output: $f_{D_Y\phi} \in \left(2^{GroupVals}\right)^G$
1: $f := Complement\,(f_\phi)$
2: $h := \emptyset$
3: for all $g \in G$ do
4:     $h(g) := \bigcup_{g' \in G} \left(Link^{\forall}_Y\,(g,g') \cap f\,(g')\right)$
5: end for
6: return  $Complement\,(h)$
```

Lemma 6 ***(Correctness of*** $Synth_D$***).***
Let $M = (G, g^0, T, \sim_1, \ldots, \sim_n)$ be an interpreted system, $g \in G$, and $\phi \in CTLPK$. Then, $M, g \models_v D_Y\phi$ iff $v \in Synth_D(f_\phi, Y)(g)$.

PROOF. We define the modality $\overline{D}_Y\phi$ as $\neg D_Y\neg\phi$; a global state g satisfies $\overline{D}_Y\phi$ under group valuation v if there exists a state g' such that $g \sim^{v(Y)}_D g'$ and $M, g \models_v \phi$.

At the end of Line 5 of Algorithm 2 we have that, for each global state $g \in G$, the set $h(g)$ consists of all assignments v such that there exists a global state $g' \in G$, where for all $i \in v(Y)$, $g \sim_i g'$ and $M, g' \models_v \neg\phi$. So we have $M, g \models_v \overline{D}_Y\neg\phi$.

By taking the complement of h at the end of Algorithm 2, we obtain the set of all assignments v, which map each global state g to a set of valuations of group variables, such that:

$$\begin{aligned} & M, g \not\models_v \overline{D}_Y\neg\phi \text{ (def. of complement)} \\ \Leftrightarrow\ & M, g \models_v \neg\overline{D}_Y\neg\phi \text{ (def. of } \not\models) \\ \Leftrightarrow\ & M, g \models_v D_Y\phi \text{ (def. of } \overline{D}_Y). \end{aligned}$$

Therefore, we have that for all $g \in G$, for all valuations $v \in f_{D_Y\phi}(g)$ iff $M, g \models_v D_Y\phi$.

□

Example 6 (Distributed Knowledge). We now adapt the interpreted system from Figure 1 by making all the states indistinguishable for Agent 2 (see Figure 2). As before, we have $\mathcal{PV} = \{p, q\}$ and the states are labelled such that $\mathcal{L}(w_0) = \mathcal{L}(w_1) = \{p\}$ and $\mathcal{L}(w_2) = \{q\}$.

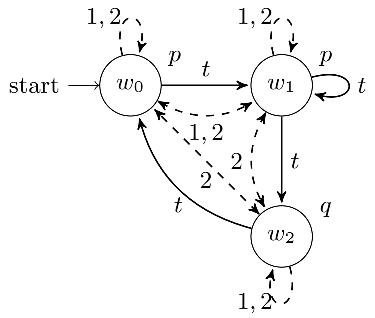

Figure 2: The interpreted system of Example 6.

The introduction of new epistemic links in the model does not change the values of either f_p or $f_{\neg p}$. Therefore, the variable f (i.e., $f_{\neg p}$) at Line 1 of Algorithm 2 is such that $f(w_0) = f(w_1) = \emptyset$ and $f(w_2) = \{Y \rightsquigarrow \{\{1\},\{2\},\{1,2\}\}\}$.

Recall that $Link^{\forall}_Y(g,g')$ consists of all group valuations assigning to variable Y groups consisting solely of agents that cannot distinguish between g and g'. We now build $Link^{\forall}_Y(g,g')$ for all the pairs of states. Again we omit the symmetric cases:

$$Link^{\forall}_Y(w_i,w_i) = \{Y \rightsquigarrow \{\{1\},\{2\},\{1,2\}\}\} \text{ for all } i \in \{0,1,2\}$$
$$Link^{\forall}_Y(w_0,w_1) = \{Y \rightsquigarrow \{\{1\},\{2\},\{1,2\}\}\}$$
$$Link^{\forall}_Y(w_1,w_2) = Link^{\forall}_Y(w_0,w_2) = \{Y \rightsquigarrow \{\{2\}\}\}$$

In Line 4 of Algorithm 2 for each state g we compute $h(g)$ as the union of sets $Link^{\forall}_Y(g,g') \cap f(g')$ over all $g' \in \{w_0, w_1, w_2\}$. Again, notice that if $g' \in \{w_0, w_1\}$ then $Link^{\forall}_Y(g,g') \cap f(g') = \emptyset$, thus we need to consider only the following cases:

$$Link^{\forall}_Y(w_0,w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{2\}\}\}$$
$$Link^{\forall}_Y(w_1,w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{2\}\}\}$$
$$Link^{\forall}_Y(w_2,w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{1\},\{2\},\{1,2\}\}\}$$

The function evaluated in the loop between Lines 3–5 and held in variable h is equal to $f_{\overline{D}_Y \neg p}$; thus after complementing in the return statement we obtain:

- $f_{D_Y p}(w_0) = \{Y \rightsquigarrow \{\{1\}, \{1,2\}\}\}$
- $f_{D_Y p}(w_1) = \{Y \rightsquigarrow \{\{1\}, \{1,2\}\}\}$
- $f_{D_Y p}(w_2) = \emptyset$

Synthesis of Common Knowledge.
Recall from [4] that $C_\Gamma \phi \Leftrightarrow E_\Gamma(\phi \wedge C_\Gamma \phi)$, for any $\Gamma \subseteq$ *Agents*. We can use this equivalence to compute the set of states satisfying $C_\Gamma \phi$ by reasoning through existential pre-images of the epistemic relations for E_Γ. We use the following algorithm, an extension of the non-parametric version presented in [12].

The synthesis of parametric common knowledge employs a similar observation, i.e., that C_Y is the fixed-point of E_Y.

Algorithm 3 $Synth_C\ (f_\phi, Y)$

```
Input: $f_\phi \in \left(2^{GroupVals}\right)^G$
Output: $f_{C_Y \phi} \in \left(2^{GroupVals}\right)^G$
1: $f := \emptyset$
2: $h := Complement\,(f_\phi)$
3: while $f \neq h$ do
4:     $f := h$
5:     for all $g \in G$ do
6:         $h(g) := \bigcup_{g' \in G} \left(Link^{\exists}_Y\,(g, g') \cap f\,(g')\right)$
7:     end for
8: end while
9: return  $Complement\,(h)$
```

Lemma 7 ***(Correctness of $Synth_C$).***
Let $M = (G, g^0, T, \sim_1, \ldots, \sim_n)$ be an interpreted system, $g \in G$, and $\phi \in CTLPK$. Then, $M, g \models_\upsilon C_Y \phi$ iff $\upsilon \in Synth_C(f_\phi, Y)(g)$.

PROOF. As usual, let $\overline{E}^0_Y \phi = \phi$, $\overline{E}^{i+1}_Y \phi = \overline{E}_Y(\overline{E}^i_Y \phi)$ for all $i \geq 0$ and $\overline{C}_Y \phi = \neg C_Y \neg \phi$. Since $C_Y \phi \Leftrightarrow E_Y\,(\phi \wedge C_Y \phi)$, we have that for each state g and each group valuation $\upsilon \in GroupVals$:

$$M, g \models_\upsilon \overline{C}_Y \phi \text{ iff } M, g \models_\upsilon \bigvee_{i=0}^{j} \overline{E}^i_Y \phi \text{ for some } j \geq 0. \qquad (\star)$$

Observe now that in Algorithm 3, prior to the execution of the loop between Lines 3–8, the function h is equivalent to $f_{\neg\phi}$. Given Lemma 4 (i.e., the correctness of $Synth_E$), after the first iteration of this inner loop h evaluates to $f_{\neg\phi \vee \overline{E}_Y \neg\phi}$. After the j-th iteration of the main body (Lines 3–8), the function h evaluates to

$$h = f_{\bigvee_{i=0}^{j} \overline{E}^i_Y \neg\phi}.$$

Given that we work on finite models, h eventually reaches a fixed-point. At that point $h = f_{\bigvee_{j=0}^{k} \overline{E}^j_Y \neg\phi}$, where k is the smallest value of j in $\star$. Therefore we have that $h = f_{\overline{C} \neg\phi}$. After taking the complement we have:

$$\begin{aligned} & M, g \not\models_\upsilon \overline{C}_Y \neg\phi \text{ (def. of complement)} \\ \Leftrightarrow\ & M, g \models_\upsilon \neg\overline{C}_Y \neg\phi \text{ (def. of } \not\models) \\ \Leftrightarrow\ & M, g \models_\upsilon C_Y \phi \text{ (def. of } \overline{C}_Y). \end{aligned}$$

This concludes the proof for $C_Y \phi$.

□

Example 7 (Common Knowledge). We now present how to synthesise $f_{C_Y p}$ for the model presented in Figure 1. As before, let $\mathcal{PV} = \{p, q\}$ and the states be labelled such that $\mathcal{L}(w_0) = \mathcal{L}(w_1) = \{p\}$ and $\mathcal{L}(w_2) = \{q\}$.

Note that in the first run of the 3–8 loop of Algorithm 3 the result of the evaluation of the function held in h variable is equal to the result of the evaluation in 3–5 loop of Algorithm 1. We can therefore reuse the values for h and $Link^{\exists}_Y(g, g')$ from Example 4 to compute:

$$\begin{aligned} & Link^{\exists}_Y(w_0, w_1) \cap f(w_1) = \{Y \rightsquigarrow \{\{1,2\}\}\} \\ & Link^{\exists}_Y(w_0, w_2) \cap f(w_2) = \emptyset \\ & Link^{\exists}_Y(w_1, w_1) \cap f(w_1) = Link^{\exists}_Y(w_1, w_2) \cap f(w_2) \\ & = Link^{\exists}_Y(w_2, w_1) \cap f(w_1) = \{Y \rightsquigarrow \{\{2\}, \{1,2\}\}\} \\ & Link^{\exists}_Y(w_2, w_2) \cap f(w_2) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\} \end{aligned}$$

Note that f is equal to h due to the substitution in Line 4 and the remaining cases are equal to $Link^{\exists}_Y(g, w_0) \cap f(w_0)$, which yields the empty set.

In the second run of the while loop we again calculate $h(g)$ for each state g by computing the union of all $Link^{\exists}_Y(g, g') \cap f(g')$ for each $g' \in \{w_0, w_1, w_2\}$ (Line 6). The result is as follows:

- $h(w_0) = \{Y \rightsquigarrow \{\{1,2\}\}\}$
- $h(w_1) = \{Y \rightsquigarrow \{\{2\}, \{1,2\}\}\}$
- $h(w_2) = \{Y \rightsquigarrow \{\{1\}, \{2\}, \{1,2\}\}\}$

It can be easily seen that the next run of the loop does not change the value of h, so the fixed-point is reached and we exit the while loop. The function held in variable h is equal to $f_{\overline{C}_Y \neg p}$, therefore after the complement in the return statement we obtain:

- $f_{C_Y p}(w_0) = \{Y \rightsquigarrow \{\{1\}, \{2\}\}\}$
- $f_{C_Y p}(w_1) = \{Y \rightsquigarrow \{\{1\}\}\}$
- $f_{C_Y p}(w_2) = \emptyset$

which concludes our example.

The following algorithm is presented to provide the entry point for calculating the function f_ϕ for given $\phi \in$ CTLPK by recursively calling previously introduced subroutines.

Algorithm 4 $Synth_{\mathrm{CTLPK}}\,(\phi)$

```
Input: $\phi \in$ CTLPK
Output: $f_\phi \in \left(2^{GroupVals}\right)^G$
1: if $\phi = Z_Y \psi$ then
2:     return  $Synth_Z(Synth_{\mathrm{CTLPK}}(\psi), Y)$
3: else /* non-parametric mod. omitted for simplicity */
4:     return  $f_\phi$
5: end if
```

Theorem 8 (Group Synthesis for CTLPK).
For each model M, for all global states $g \in G$ and for all formulae ϕ in CTLPK, we have that $M, g \models_\upsilon \phi$ iff $\upsilon \in f_\phi(g)$.

PROOF. The validity of the theorem follows immediately from the previous treatment of propositions, Boolean and temporal operators, Lemmas 4, 6 and 7 and Corollary 5. □

4. EVALUATION

We have implemented the presented parametric approach as an experimental extension to the open-source model checker MCMAS [9]. A GNU GPL licenced release is available from `http://vas.doc.ic.ac.uk/tools/mcmas_parametric/`. As MCMAS is a symbolic model checker, the satisfiable group valuations are also stored symbolically. We compare this to a naïve approach that iteratively checks each possible group assignment for satisfiability. Observe that for n agents and m group variables, there are $(2^n - 1)^m$ unique group assignments.

We carry out this comparison using two benchmarks.

4.1 Dining Cryptographers

We first consider Chaum's Dining Cryptographers protocol [1]; this problem has been widely studied with respect to multi-agent systems and temporal-epistemic logic [10].

We consider the following two parametric specifications (we write $paid_i$ to represent the proposition "diner i paid"):

- $\varphi_{DC1} = AG\,(paid_1 \rightarrow (C_Y\,(paid_1)))$
- $\varphi_{DC2} = AG\,(paid_1 \rightarrow (C_Y\,(paid_1 \wedge \neg D_Z\,(\neg paid_2))))$

The first specification φ_{DC1} expresses "if diner one paid, then the group Y has common knowledge that diner one paid". This specification is satisfied only for the valuation $Y = \{Diner\_1\}$. The formula φ_{DC2} extends the first formula by additionally stating that "the group Y also has common knowledge that the group Z does not have distributed knowledge that diner two did not pay". Considering only the case in which diner one paid, this specification is satisfied when $Z = GroupVals \setminus \{Diner\_1, Diner\_2\}$ (Y is as for φ_{DC1}). The group Z cannot include $Diner\_2$, because if diner one paid, then diner two knows (individually) that he did not.

The experimental results for parameter synthesis for these formulae over models of varying size can be seen in Table 1. The values were collected over three runs, with a timeout of one hour per run. The machine employed for these benchmarks was an Intel Core 2 Duo processor 3.00 GHz, with a 6144 KiB cache, and ran 32-bit Fedora 14, kernel 2.6.35.14. These results show that the parametric approach can, memory permitting, perform synthesis faster than the naïve approach. This is exemplified for 18 diners and the formula φ_{DC2}, where parametric verification completed in under 11 minutes but naïve did not finish within the hour.

For a model containing 14 diners, the construction of the reachable state space, regardless of implementation or formula, exhibited an unusually high run-time. This is reflected in the TIMEOUT and MEMOUT results for this sized model. We believe that this was caused by BDD reordering within the CUDD library utilised by MCMAS.

4.2 IEEE Token Ring Network

We now compare the parametric and naïve approaches using the industry standard IEEE token ring bus network. In the comparison that follows, we automated the injection of faults into the model, following the approach of [3]; this allows for the automatic analysis of fault-diagnosability properties. We briefly summarise the scenario below; for a complete description we refer the reader to [3].

The IEEE token ring protocol connects n nodes in a ring topology; data moves between nodes on the network in a clockwise fashion. Access is granted to nodes on the network in the form of a token; this is passed from node to node. Tokens are issued onto the network from an "active monitor". To detect faults, tokens contain a "time to live" field, initialised to the maximum time that a token would take to circulate the whole network and counting down to zero. Should a token fail to circulate back to the active monitor within the given time-frame, it is deduced that a fault has occurred on the network.

We consider instantiations of the network where the first node wishes to transmit a data token to the final node. Consequently, data needs to pass through every single intermediate node on the system.

Using a modified version of the fault injector from [3], we inserted two types of non-deterministic faults: even nodes stop sending tokens and odd nodes stop receiving tokens. The properties verified are shown in Table 2. To exemplify: φ_{TR2} states that "there exists a future state where the group Y has common knowledge that the group Z does not possess common knowledge that faults have not been injected".

Table 3 shows the comparison between the parametric and naïve approach. These results demonstrate the benefits of parametric verification.

5. CONCLUSIONS

We have introduced a novel parametric temporal-epistemic logic, as well as presenting a sound and complete approach for parameter synthesis. Using two non-trivial examples, we have shown that a symbolic implementation of the parametric technique can be more efficient than a brute force approach.

The results corroborate the expected: the parametric technique sacrifices memory efficiency for tractability. The current experimental results seem favourable to the parametric approach. However, the results also demonstrate that there exist models where the naïve technique can complete synthesis, while the parametric approach runs out of memory.

We are interested in applying the parametric technique to a wider range of industrial scenarios, for example, those where the synthesised groups can be used to define cliques of agents during the design and implementation phase.

Acknowledgements.

The authors acknowledge support from: the EPSRC (grants EP/E02727X/1 and EP/I00520X/1); the Polish National Science Centre (grant No. 2011/01/B/ST6/01477); and the SRI PAS sponsored International Ph.D. Projects in Intelligent Computing.

6. REFERENCES

[1] D. Chaum. The Dining Cryptographers Problem. *J. Cryptol.*, 1:65–75, 1988.

[2] E. M. Clarke, Jr., O. Grumberg, and D. A. Peled. *Model Checking.* MIT Press, Cambridge, MA, USA, 1999.

[3] J. Ezekiel and A. Lomuscio. A Methodology for Automatic Diagnosability Analysis. In *Proc. of ICFEM'10*, pages 549–564, 2010.

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

Table 1: Comparison for the Dining Cryptographers

Model		Formula	Group Valuations		Time (s)		Memory (KiB)	
Diners	States		Possible	# SAT	PARAMETRIC	NAÏVE	PARAMETRIC	NAÏVE
6	1,344	φ_{DC1}	63	1	0.084	0.064	11,500	10,504
		φ_{DC2}	3,969	15	0.155	0.777	13,184	10,504
10	33,792	φ_{DC1}	1,023	1	1.054	3.380	30,372	46,164
		φ_{DC2}	1,046,529	255	1.959	1,286.33	53,296	45,800
14	737,280	φ_{DC1}	16,383	1	TIMEOUT	317.093	TIMEOUT	138,951
		$\varphi_{DC2}{}^*$	268,402,689	$-^*$	MEMOUT	TIMEOUT	MEMOUT	TIMEOUT
18	$1.5 \cdot 10^7$	φ_{DC1}	262,143	1	324.034	731.034	$2.096 \cdot 10^6$	$1.747 \cdot 10^6$
		φ_{DC2}	68,718,952,449	65535	651.206	TIMEOUT	$2.606 \cdot 10^6$	TIMEOUT

* MCMAS-PARAMETRIC used over 3 GiB of RAM and MCMAS-NAÏVE failed to finish within the one-hour timeout.

Table 2: Diagnosability Properties for the Token Ring Protocol

Formula	Specification
φ_{TR1}	$EF\ C_Y \neg (\bigvee_{fault \in Faults} fault\_injected)$
φ_{TR2}	$EF\ C_Y \neg C_Z \neg (\bigvee_{fault \in Faults} fault\_injected)$
φ_{TR3}	$E[(C_Y \neg(\bigvee_{fault \in Faults} fault\_injected))\ U\ (EF\ E_Z\ D_V \neg(\bigvee_{fault \in Faults} (fault\_injected \vee fault\_stopped)))]$

Table 3: Comparison for the Token Ring Network

Model		Formula	Group Valuations		Time (s)		Memory (KiB)	
Nodes	States		Possible	# SAT	PARAMETRIC	NAÏVE	PARAMETRIC	NAÏVE
4	1,116	φ_{TR1}	15	5	0.594	0.544	14,016	12,072
		φ_{TR2}	225	171	0.752	0.761	25,388	12,300
		φ_{TR3}	3,375	3,255	0.776	3.117	15,888	12,160
6	21,578	φ_{TR1}	63	7	1.434	1.209	23,684	23,680
		φ_{TR2}	3,969	3,571	2.321	7.285	55,588	23,680
		φ_{TR3}	250,047	232,407	11.368	408.478	40,752	23,680
8	336,632	φ_{TR1}	255	9	3.369	3.107	58,708	39,960
		φ_{TR2}	65,025	62,803	7.497	193.948	63,496	48,200
		φ_{TR3}	16,581,375	15,253.335	1,127.170	TIMEOUT†	62,316	TIMEOUT†
10	$7.769 \cdot 10^6$	φ_{TR1}	1,023	11	6.236	17.052	58,444	53,756
		φ_{TR2}	1,046,529	1,035,387	65.875	TIMEOUT‡	61,788	TIMEOUT‡
		$\varphi_{TR3}{}^*$	1,070,599,167	$-^*$	TIMEOUT*			
12	$1.157 \cdot 10^8$	φ_{TR1}	4,095	13	15.556	204.853	66,224	109,280$^\bullet$
		φ_{TR2}	16,769,025	16,715,947	1,423.12	TIMEOUT*	71,160	TIMEOUT*
		$\varphi_{TR3}{}^*$	68,669,157,375	$-^*$	TIMEOUT*			

† Calculated 8.35% of all the satisfiable groups within the timeout period.
‡ Calculated 73.92% of all the satisfiable groups within the timeout period.
• Represents an anomalous result with MCMAS-NAÏVE/CUDD. The memory usage of MCMAS-NAÏVE should, under a correctly functioning system, always be lower than that of MCMAS-PARAMETRIC.
⋆ Calculated 1.82% of all the satisfiable groups within the timeout period.
* Both MCMAS-PARAMETRIC and MCMAS-NAÏVE failed to halt-and-succeed within the one-hour time limit.

[5] P. Gammie and R. van der Meyden. MCK: Model Checking the Logic of Knowledge. In *Proc. of CAV'04*, pages 479–483, 2004.

[6] M. Huth and M. Ryan. *Logic in Computer Science: Modelling and Reasoning About Systems (Second Edition)*. Cambridge University Press, 2004.

[7] 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. *Fundam. Inform.*, 85(1-4):313–328, 2008.

[8] M. Knapik, W. Penczek, M. Szreter, and A. Pólrola. Bounded Parametric Verification for Distributed Time Petri Nets with Discrete-Time Semantics. *Fundam. Inform.*, 101(1–2):9–27, 2010.

[9] A. Lomuscio, H. Qu, and F. Raimondi. MCMAS: A Model Checker for the Verification of Multi-Agent Systems. In *Proc. of CAV'09*, pages 682–688, 2009.

[10] R. v. d. Meyden and K. Su. Symbolic Model Checking the Knowledge of the Dining Cryptographers. In *Proc. of CSFW'04*, pages 280–291, 2004.

[11] W. Penczek and A. Lomuscio. Verifying Epistemic Properties of Multi-Agent Systems via Bounded Model Checking. *Fundam. Inform.*, 55(2):167–185, 2003.

[12] F. Raimondi and A. Lomuscio. Automatic Verification of Multi-Agent Systems by Model Checking via Ordered Binary Decision Diagrams. *J. Applied Logic*, 5(2):235–251, 2007.

[13] F. Wang. Timing Behavior Analysis for Real-Time Systems. In *Proc. of LICS'95*, pages 112–122, 1995.