

Verification and Strategy Synthesis for Coalition Announcement Logic

Natasha Alechina* Hans van Ditmarsch† Rustam Galimullin‡
Tuo Wang§

Abstract

Coalition announcement logic (CAL) is one of the family of the logics of quantified announcements. It allows us to reason about what a coalition of agents can achieve by making announcements in the setting where the anti-coalition may have an announcement of their own to preclude the former from reaching its epistemic goals. In this paper, we describe a PSPACE-complete model checking algorithm for CAL that produces winning strategies for coalitions. The algorithm is implemented in a proof-of-concept model checker.

1 Introduction

In the multi-agent logic of knowledge we investigate what agents know about their factual environment and what they know about knowledge of each other [27]. (Truthful) Public announcement logic (PAL) is an extension of the multi-agent logic of knowledge with modalities for public announcements. Such modalities model the event of incorporating trusted information that is similarly observed by all agents [32]. The ‘truthful’ part relates to the trusted aspect of the information: we assume that the novel information is true.

In [2] the authors propose two generalisations of public announcement logic, GAL (group announcement logic) and CAL (coalition announcement logic). These logics allow for quantification over public announcements made by agents modelled in the system. In particular, the GAL quantifier $\langle G \rangle \varphi$ (parametrised by a subset G of the set of all agents A) says ‘there is a truthful announcement made by the agents in G , after which φ (holds)’. Here, the truthful aspect means that the agents in G only announce what they *know*: if a in G announces φ_a , this is interpreted as a public announcement $K_a \varphi_a$, and a truthful group announcement by agents in G is a conjunction of such known announcements. The CAL quantifier $\langle [G] \rangle \varphi$ is motivated by game logic [31, 30] and Van Benthem’s playability

*Utrecht University, Utrecht, The Netherlands; n.a.alechina@uu.nl

†Open University of the Netherlands, Heerlen, The Netherlands; hans.vanditmarsch@ou.nl

‡University of Bergen, Bergen, Norway; rustam.galimullin@uib.no

§University of Edinburgh, Edinburgh, UK; toy.haagend.wang@gmail.com

operator [9]. Here, the modality means ‘there is a truthful announcement made by the agents in G such that no matter what the agents not in G simultaneously announce, φ holds afterwards’. In [2] it is, for example, shown that CAL subsumes game logic.

CAL has been far less investigated than other logics of quantified announcements, such as APAL [6] and GAL, although some combined results have been achieved [4, 19, 15]. In particular, model checking for CAL, which has potential practical implications, has not been studied. For example, in CAL it is possible to express that a group of agents (for instance, a subset of bidders in an auction) can make an announcement such that no matter what other agents announce simultaneously, after this announcement certain knowledge is increased (all agents know that G has won the bid) but certain ignorance also remains (for example, the maximal amount of money G could have offered). The main contribution of this paper is a thorough analysis of the model checking problem for CAL and a description of an implemented open source model checker for CAL and GAL formulas.

This paper is a revised and extended version of [21], with detailed proofs and a new section on the implementation of the model-checking algorithm, which also contains a large worked example. The structure of the paper is as follows. Section 2 provides the necessary background on GAL and CAL, and in Section 3 we use distinguishing formulas to make a shift from an infinite number of agents’ announcements to a finite number of strategies available to them. The model checking algorithm is given in Section 4. The algorithm presented here differs from the one presented in [21] in several respects. Instead of iterating over an explicit list of strategies, it generates and tests strategies one at a time, so that it only uses polynomial space. Instead of returning *true* and *false*, the version presented here returns a set of states satisfying the formula; for true formulas starting with a GAL or CAL modality, it also outputs a strategy. The model checking algorithm and the proof of PSPACE-completeness build on those for GAL [1], but the algorithm for CAL requires some modifications; in particular the algorithms in [1] runs in APTIME by ‘guessing’ strategies while our algorithm is deterministic. We also describe an efficient (PTIME) special case. The algorithm is implemented in a proof of concept model checker MCCAL available on <https://github.com/Twelvelines/MCCAL>. The implementation and its performance are described in detail in [33], and briefly in Section 5 of this paper.

2 Background

2.1 Introductory Example

Two agents, a and b , want to buy the same item, and whoever offers the greatest sum, gets it. Agents may have 5, 10, or 15 pounds, and they do not know which sum the opponent has. Let agent a have 15 pounds, and agent b have 5 pounds. This situation is presented in Figure 1.

In this model (let us call it M), state names denote money distribution. Thus, 10_a5_b means that agent a has 10 pounds, and agent b has 5 pounds. Labelled edges connect the states that a corresponding agent cannot distinguish. For example, in the actual state

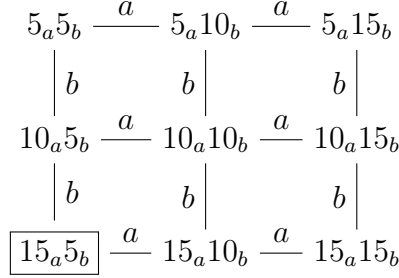


Figure 1: Initial model $M_{15_a 5_b}$

(boxed), agent a knows that she has 15 pounds, but she does not know how much money agent b has. Formally, $M_{15_a 5_b} \models K_a 15_a \wedge \neg(K_a 5_b \vee K_a 10_b \vee K_a 15_b)$ (which means $M_{15_a 5_b}$ satisfies the formula, where $K_i \varphi$ stands for ‘agent i knows that φ ’, \wedge is logical ‘and’, \neg is ‘not’, and \vee is ‘or’). Note that edges represent equivalence relations, and in the figure we omit transitive and reflexive transitions.

Next, suppose that agents bid in order to buy the item. Once one of the agents, let us say a , announces her bid, she also wants the other agent to remain ignorant of the total sum at her disposal. Formally, we can express this goal as formula $\varphi := K_b(10_a \vee 15_a) \wedge \neg(K_b 10_a \vee K_b 15_a)$ (for bid 10 by agent a). Informally, if a commits to pay 10 pounds, agent b knows that a has 10 or more pounds, but b does not know the exact amount. If agent b does not participate in announcing (bidding), a can achieve the target formula φ by announcing $K_a 10_a \vee K_a 15_a$. In other words, agent a commits to pay 10 pounds, which denotes that she has at least that sum at her disposal. In general, this means that there is an announcement by a such that after this announcements φ holds. Formally, $M_{15_a 5_b} \models \langle a \rangle \varphi$. The updated model $M_{15_a 5_b}^{K_a 10_a \vee K_a 15_a}$, which is a restriction of the original model to the states where $K_a 10_a \vee K_a 15_a$ holds, is presented in Figure 2.

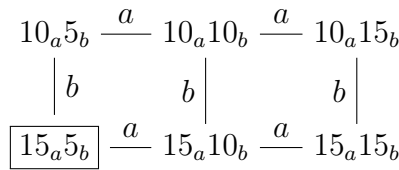


Figure 2: Updated model $M_{15_a 5_b}^{K_a 10_a \vee K_a 15_a}$

Indeed, in the updated model agent b knows that a has at least 10 pounds, but not the exact sum. The same holds if agent b announces her bid simultaneously with a in the initial situation. Moreover, a can achieve φ no matter what agent b announces, e.g. $K_b 5_b$, $K_b(5_b \vee 10_b)$, or $K_b(5_b \vee 10_b \vee 15_b)$, since all their announcements made in the conjunction with a ’s announcement $K_a 10_a \vee K_a 15_a$ result in the updated models satisfying φ . Formally, $M_{15_a 5_b} \models \langle a \rangle \varphi$.

2.2 Syntax and Semantics of CAL

Let A denote a finite set of agents, and P denote a countable set of propositional variables.

Definition 1. The *language of coalition announcement logic* \mathcal{L}_{CAL} is defined by the following BNF:

$$\varphi, \psi ::= p \mid \neg\varphi \mid (\varphi \wedge \psi) \mid K_a\varphi \mid [\psi]\varphi \mid \llbracket G \rrbracket\varphi,$$

where $p \in P$, $a \in A$, $G \subseteq A$, and all the usual abbreviations of propositional logic and conventions for deleting parentheses hold. The dual operators are defined as follows: $\widehat{K}_a\varphi := \neg K_a\neg\varphi$, $\langle \psi \rangle\varphi := \neg[\psi]\neg\varphi$, and $\llbracket G \rrbracket\varphi := \neg\llbracket G \rrbracket\neg\varphi$. The language of group announcement logic \mathcal{L}_{GAL} is obtained by replacing $\llbracket G \rrbracket\varphi$ with $[G]\varphi$ (the dual is written as $\langle G \rangle\varphi$). Language \mathcal{L}_{PAL} is the language without the operator $\llbracket G \rrbracket\varphi$, and \mathcal{L}_{EL} is the pure epistemic language without the operators $[\psi]\varphi$ and $\llbracket G \rrbracket\varphi$.

Formulas of CAL are interpreted in epistemic models.

Definition 2. An *epistemic model* is a triple $M = (W, \sim, V)$, where W is a non-empty set of states, $\sim: A \rightarrow \mathcal{P}(W \times W)$ assigns an equivalence relation to each agent, and $V: P \rightarrow \mathcal{P}(W)$ assigns a set of states to each propositional variable. If necessary, we refer to the elements of the model as W^M , \sim^M , and V^M .

Epistemic model M is called *finite* if W is finite. A pair M_w with $w \in W$ is called a *pointed model*. Also, we write $M \subseteq N$ if $W^M \subseteq W^N$, and \sim^M and V^M are restrictions of \sim^N and V^N to W^M , and then call M a *submodel* of N .

Definition 3. For a model $M = (W, \sim, V)$ and $X \subseteq W$, an *updated model* M_w^X is the tuple (W^X, \sim^X, V^X) , where $X \subseteq W$, $w \in X$, $W^X = X$, $\sim_a^X = \sim_a \cap (X \times X)$ for all $a \in A$, and $V^X(p) = V(p) \cap X$.

Let \mathcal{L}_{EL}^G denote the set of formulas of the form $\bigwedge_{i \in G} K_i\varphi_i$, where for every $i \in G$ it holds that $\varphi_i \in \mathcal{L}_{EL}$.

Definition 4. Let M_w be a pointed epistemic model. The *semantics* is inductively defined as follows:

$$\begin{aligned} M_w \models p & \quad \text{iff } w \in V^M(p) \\ M_w \models \neg\varphi & \quad \text{iff } M_w \not\models \varphi \\ M_w \models \varphi \wedge \psi & \quad \text{iff } M_w \models \varphi \text{ and } M_w \models \psi \\ M_w \models K_a\varphi & \quad \text{iff } \forall v \in W : w \sim_a^M v \text{ implies } M_v \models \varphi \\ M_w \models [\varphi]\psi & \quad \text{iff } M_w \models \varphi \text{ implies } M_w^X \models \psi, \text{ where } X = \{v \in W \mid M_v \models \varphi\} \\ M_w \models \llbracket G \rrbracket\varphi & \quad \text{iff } \forall \psi \in \mathcal{L}_{EL}^G \exists \chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \rightarrow \langle \psi \wedge \chi \rangle\varphi \end{aligned}$$

For clarity, we will write $M_w^\varphi = (W^\varphi, \sim^\varphi, V^\varphi)$ for $M_w^X = (W^X, \sim^X, V^X)$ whenever $X = \{v \in W \mid M_v \models \varphi\}$. Also note that, in order to avoid circularity, quantification in the condition for coalition announcements is restricted to formulas of epistemic logic

Since in the model checking procedure we will also be considering GAL modalities, we provide a truth definition for $[G]\varphi$:

$$M_w \models [G]\varphi \text{ iff } \forall \psi \in \mathcal{L}_{EL}^G : M_w \models [\psi]\varphi$$

Formula $[G]\varphi$ is read as ‘whatever agents from G announce, φ holds.’ The operator for coalition announcements $\llbracket G \rrbracket \varphi$ is read as ‘whatever agents from G announce, there is a simultaneous announcement by agents from $A \setminus G$ such that φ holds.’

The semantics for the ‘diamond’ versions of group/strategic announcement operators is as follows:

$$\begin{aligned} M_w \models \langle G \rangle \varphi &\text{ iff } \exists \psi \in \mathcal{L}_{EL}^G : M_w \models \langle \psi \rangle \varphi \\ M_w \models \llbracket G \rrbracket \varphi &\text{ iff } \exists \psi \in \mathcal{L}_{EL}^G \forall \chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \wedge [\psi \wedge \chi] \varphi \end{aligned}$$

They are read as ‘there is a combined simultaneous announcements by agents from G , such that φ holds,’ and ‘there is an announcement by agents from G , such that whatever agents from $A \setminus G$ announce at the same time, φ holds’ correspondingly.

Definition 5. We call formula φ a *validity* if and only if for any pointed model M_w it holds that $M_w \models \varphi$. Given M_w and φ , we say that φ is *satisfied in M_w* if and only if $M_w \models \varphi$.

2.3 Bisimulation

The notion of bisimulation [11, Chapter 2] plays a significant role in the paper.

Definition 6. Let $M = (W^M, \sim^M, V^M)$ and $N = (W^N, \sim^N, V^N)$ be two models. A non-empty binary relation $Z \subseteq W^M \times W^N$ is called a *bisimulation* if and only if for all $w \in W^M$ and $t \in W^N$ with $(w, t) \in Z$:

Atoms for all $p \in P$, $w \in V^M(p)$ if and only if $t \in V^N(p)$;

Forth for all $a \in A$ and all $v \in W^M$: if $w \sim_a^M v$, then there is a $u \in W^N$ such that $t \sim_a^N u$ and $(v, u) \in Z$;

Back for all $a \in A$ and all $u \in W^N$: if $t \sim_a^N u$, then there is a $v \in W^M$ such that $w \sim_a^M v$ and $(v, u) \in Z$.

If there is a bisimulation between models M and N linking states w and t , we say that M_w and N_t are bisimilar, and write $M_w \rightleftharpoons N_t$.

Next, we show an extension of the well-known result that bisimulation between states implies that these states satisfy the same formulas.

Proposition 1. Let $M = (W^M, \sim^M, V^M)$ and $N = (W^N, \sim^N, V^N)$ be epistemic models such that $M_w \rightleftharpoons N_v$ for some $w \in W^M$ and $v \in W^N$. Then for all $\varphi \in \mathcal{L}_{GAL} \cup \mathcal{L}_{CAL}$, $M_w \models \varphi$ if and only if $N_v \models \varphi$.

Proof. The proof is by induction on the structure of φ . Note that it is straightforward to define a *size* relation between formulas in such a way that the quantifier depth of formulas is considered before the modal depth and subformula relation.

The boolean cases are immediate, and the proof for the case of public announcements can be found, for example, in [14]. Here we prove only the case for coalition announcements, since the case for group announcements is similar (and simpler).

Induction hypothesis. For all $\varphi, \psi \in \mathcal{L}_{GAL} \cup \mathcal{L}_{CAL}$ such that $size(\varphi) < size(\psi)$, and all $w' \in W^M, v' \in W^N$: $M_{w'} \models \varphi$ if and only if $N_{v'} \models \varphi$.

Case $\langle\langle G \rangle\rangle\varphi$. \Rightarrow : Let $M_w \models \langle\langle G \rangle\rangle\varphi$. By the definition of semantics, this is equivalent to $\exists\psi \in \mathcal{L}_{EL}^G \forall\chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \wedge [\psi \wedge \chi]\varphi$. Because the quantifier depth of $\langle\langle G \rangle\rangle\varphi$ is greater than the quantifier depth of ψ or $[\psi \wedge \chi]\varphi$, we have that $size(\psi) < size(\langle\langle G \rangle\rangle\varphi)$ and $size([\psi \wedge \chi]\varphi) < size(\langle\langle G \rangle\rangle\varphi)$. By the induction hypothesis, we have that for all formulas $\psi \in \mathcal{L}_{EL}^G, M_w \models \psi$ if and only if $N_v \models \psi$, and for all $\psi \in \mathcal{L}_{EL}^G$ and $\chi \in \mathcal{L}_{EL}^{A \setminus G}, M_w \models [\psi \wedge \chi]\varphi$ if and only if $N_v \models [\psi \wedge \chi]\varphi$. Hence, $\exists\psi \in \mathcal{L}_{EL}^G \forall\chi \in \mathcal{L}_{EL}^{A \setminus G} : N_v \models \psi \wedge [\psi \wedge \chi]\varphi$ which is equivalent to $N_v \models \langle\langle G \rangle\rangle\varphi$ by the semantics. \square

The bisimulation contraction of a model is, informally, the most compact representation of that model.

Definition 7. Let $M = (W, \sim, V)$ be a model. The *bisimulation contraction* of M is the model $\|M\| = (\|W\|, \|\sim\|, \|V\|)$, where $\|W\| = \{[w] \mid w \in W\}$ and $[w] = \{v \in W \mid M_w \Leftrightarrow M_v\}$, $[w] \|\sim\|_a [v]$ if and only if $\exists w' \in [w], \exists v' \in [v]$ such that $w' \sim_a v'$, and $[w] \in \|V\|(p)$ if and only if $w \in V(p)$. We call a model M *bisimulation contracted* if M is isomorphic to $\|M\|$.

It is a standard result that $M_w \Leftrightarrow \|M\|_{[w]}$ (see, for example, [23]).

Corollary 1. For all $\varphi \in \mathcal{L}_{GAL} \cup \mathcal{L}_{CAL}$, $\|M\|_{[w]} \models \varphi$ if and only if $M_w \models \varphi$.

3 Strategies of Groups of Agents on Finite Models

3.1 Distinguishing Formulas

In this section we introduce distinguishing formulas that are satisfied in only one (up to bisimulation) state in a finite model. The discussion is based on [13]. Although agents know and can possibly announce an infinite number of formulas, using distinguishing formulas allows us to consider only finitely many different announcements. This is done by associating strategies of agents with corresponding distinguishing formulas, where a strategy of agent a is a union of a -equivalence classes.

Without loss of generality, we assume that the set of propositional variables P is finite. This is justified by the fact that in a finite epistemic model $M = (W, \sim, V)$ there are $2^{|W|}$ unique truth assignments for a propositional variable, and a truth assignment for any $p_{2^{|W|}+1}$ will repeat one from $p_1, \dots, p_{2^{|W|}}$.

We continue with the formal definition of distinguishing formulas.

Definition 8. Let a finite epistemic model $M = (W, \sim, V)$ be given. Formula $\delta_{S,S'}$ is called *distinguishing* for $S, S' \subseteq W$ if $S \subseteq W^{\delta_{S,S'}}$ and $S' \cap W^{\delta_{S,S'}} = \emptyset$. If a formula distinguishes state w from all other non-bisimilar states in M , we abuse the notation and write δ_w .

Proposition 2 ([7],[13]). Let $M = (W, \sim, V)$ be a finite epistemic model. Every state w in M has a distinguishing formula $\delta_w \in \mathcal{L}_{EL}$.

Proof. The construction and proof are found in [13, Lemma 8.1], that in its turn copies the argument found in [7, Section 3.1]. Instead of technical detail we sketch the argument.

Let M_w be a finite pointed epistemic model. Without loss of generality we assume that M is bisimulation contracted. A distinguishing formula δ_w is constructed recursively as follows, where $k \in \mathbb{N}$:

$$\begin{aligned} \delta_w^0 &:= \bigwedge_{w \in V(p)} p \wedge \bigwedge_{w \notin V(p)} \neg p \\ \delta_w^{k+1} &:= \delta_w^k \wedge \bigwedge_{a \in A} (\bigwedge_{w \sim_a v} \widehat{K}_a \delta_v^k \wedge K_a \bigvee_{w \sim_a v} \delta_v^k) \\ \delta_w &:= \delta_w^{|\mathcal{W}|^2} \end{aligned}$$

Conjuncts δ_w^0 , $\bigwedge_{w \sim_a v} \widehat{K}_a \delta_v^k$, and $K_a \bigvee_{w \sim_a v} \delta_v^k$ respectively emulate conditions **Atoms**, **Forth**, and **Back** of the definition of bisimulation. Indeed, it is then easy to see that the binary relation Z on W defined by:

$$\text{For all } w, v \in W, (w, v) \in Z \text{ iff for all } k \in \mathbb{N}, M_w \models \delta_v^k,$$

is a bisimulation on finite models. Therefore, for each $w \in W$ that is not bisimilar to v there is a $\max \in \mathbb{N}$ such that $M_w \not\models \delta_v^{\max}$. One can take $\max = |W|^2$, as also observed in [7, Section 5]. \square

Assumptions regarding some given model being finite and bisimulation contracted are of vital importance for the construction of distinguishing formulas. If the model is infinite, then we may either need an infinite amount of propositional variables to describe the given state, or there may be infinite branches of accessibility relations. If the model is not bisimulation contracted, i.e. there are bisimilar states in the model, then distinguishing formulas cease to be unique — the same formula describes all bisimilar states in the model. This becomes a problem if we want to switch from agents announcing formulas to agents ‘choosing’ a definable submodel: in the latter case agents may distinguish between bisimilar states.

Having defined distinguishing formulas for states, we can define distinguishing formulas for sets of states.

Definition 9. Let M_w be a finite model and S be a set of states in M . A *distinguishing formula for S* is

$$\delta_S := \bigvee_{w \in S} \delta_w.$$

Let us recall the bidding example from Section 2.1, and construct the distinguishing formula $\delta_{15_a5_b}$. Note that for this particular example it is enough to construct distinguishing formulas of depth 0 only. This is due to the fact that each state in the example has a unique valuation of propositional variables. We, however, proceed with the full construction for illustrative purposes.

First, we start with the propositional description of the state:

$$\delta_{15_a5_b}^0 := 15_a \wedge 5_b \wedge \neg 10_a \wedge \neg 5_a \wedge \neg 10_b \wedge \neg 15_b.$$

Let us assume that we calculated δ^0 's in the same fashion for all other states. Next, we proceed with the first iteration of $\delta_{15_a5_b}$:

$$\delta_{15_a5_b}^1 := \delta_{15_a5_b}^0 \wedge \bigwedge \left(\begin{array}{l} \widehat{K}_a \delta_{15_a5_b}^0 \wedge \widehat{K}_a \delta_{15_a10_b}^0 \wedge \widehat{K}_a \delta_{15_a15_b}^0 \wedge K_a \left(\bigvee \begin{array}{l} \delta_{15_a5_b}^0 \\ \delta_{15_a10_b}^0 \\ \delta_{15_a15_b}^0 \end{array} \right) \\ \widehat{K}_b \delta_{15_a5_b}^0 \wedge \widehat{K}_b \delta_{10_a5_b}^0 \wedge \widehat{K}_b \delta_{5_a5_b}^0 \wedge K_b \left(\bigvee \begin{array}{l} \delta_{15_a5_b}^0 \\ \delta_{10_a5_b}^0 \\ \delta_{5_a5_b}^0 \end{array} \right) \end{array} \right).$$

The process continues for $|W|^2$ iterations. Informally, each iteration of a distinguishing formula construction adds one layer for each state in a model. Hence, in our example with 9 states distinguishing formula $\delta_{15_a5_b}$ looks as follows (assuming that all previous $\delta_{15_a5_b}^k$'s with $k \leq |W|^2 - 1$ have been calculated):

$$\delta_{15_a5_b}^{k+1} := \delta_{15_a5_b}^0 \wedge \bigwedge \left(\begin{array}{l} \widehat{K}_a \delta_{15_a5_b}^k \wedge \widehat{K}_a \delta_{15_a10_b}^k \wedge \widehat{K}_a \delta_{15_a15_b}^k \wedge K_a \left(\bigvee \begin{array}{l} \delta_{15_a5_b}^k \\ \delta_{15_a10_b}^k \\ \delta_{15_a15_b}^k \end{array} \right) \\ \widehat{K}_b \delta_{15_a5_b}^k \wedge \widehat{K}_b \delta_{10_a5_b}^k \wedge \widehat{K}_b \delta_{5_a5_b}^k \wedge K_b \left(\bigvee \begin{array}{l} \delta_{15_a5_b}^k \\ \delta_{10_a5_b}^k \\ \delta_{5_a5_b}^k \end{array} \right) \end{array} \right).$$

Note that since models we are dealing with in this paper are finite, distinguishing formulas always exist.

3.2 Strategies

In this section we introduce strategies and connect them to public announcements using distinguishing formulas. In the setting of GAL, strategies are sets of states that agents can ensure to be in the updated model by announcing a formula that holds in those states. For CAL, however, this is not always true, as the anti-coalition may have a counter-strategy to reduce the set initially chosen by the coalition. Still, we use word ‘strategy’ in both contexts. The formal definition of a strategy is presented below.

Definition 10. A *strategy for an agent a* in a finite model M_w is a union of equivalence classes of a containing the a -equivalence class of w . Let $S(a, w)$ be the set of all strategies

for agent a in M_w . A *strategy for group G* is defined as $\bigcap_{i \in G} X_i$ such that for all $i \in G$, $X_i \in S(i, w)$. The set of available strategies for a group of agents G in M_w is denoted by $S(G, w)$.

Strategies are implemented by agents, and generally public announcements do not correspond to strategies. Consider model $M_{15_a 5_b}$ in Figure 1 and formula $\varphi := (15_a \wedge 5_b) \vee (15_a \wedge 10_b) \vee (10_a \wedge 5_b)$. It is easy to see that public announcement of φ does not correspond to any strategy of a and b , that is $W^\varphi \notin S(\{a, b\}, 15_a 5_b)$.

Note that for any M_w and $G \subseteq A$, $S(G, w)$ is not empty, since the trivial strategy that includes all the states of the current model is available to all agents. We denote the trivial strategy by X^\top .

Proposition 3. In a finite model M_w , for any $G \subseteq A$, $S(G, w)$ is finite.

Proof. This is due to the fact that in a finite model there is a finite number of equivalence classes for each agent. \square

Thus, in Figure 1 there are three a -equivalence classes: $\{15_a 5_b, 15_a 10_b, 15_a 15_b\}$, $\{10_a 5_b, 10_a 10_b, 10_a 15_b\}$, and $\{5_a 5_b, 5_a 10_b, 5_a 15_b\}$. Let us designate them by the first element of a corresponding set, printed in typewriter font, i.e. $15_a 5_b$, $10_a 5_b$, and $5_a 5_b$. The set of all available strategies of agent a in $M_{15_a 5_b}$ is $\{15_a 5_b, 15_a 5_b \cup 10_a 5_b, 15_a 5_b \cup 5_a 5_b, 15_a 5_b \cup 10_a 5_b \cup 5_a 5_b\}$. Similarly, the set of all available strategies of agent b in $M_{15_a 5_b}$ is $\{15_a 5_b, 15_a 5_b \cup 15_a 10_b, 15_a 5_b \cup 15_a 15_b, 15_a 5_b \cup 10_a 5_b \cup 15_a 15_b\}$. Finally, all possible intersections of a 's and b 's strategies from the set of all available strategies of group $\{a, b\}$. For example, there is a group strategy for agents a and b that contains only two states – $15_a 5_b$ and $10_a 5_b$. This strategy is an intersection of a 's $15_a 5_b \cup 10_a 5_b$ and b 's $15_a 5_b$, that is $\{15_a 5_b, 15_a 10_b, 15_a 15_b, 10_a 5_b, 10_a 10_b, 10_a 15_b\} \cap \{15_a 5_b, 10_a 5_b, 5_a 5_b\}$.

Now we tie together announcements and strategies. Each of infinitely many possible announcements by agents in a finite model corresponds to a set of states where it is true (a strategy). In a finite bisimulation contracted model, each strategy is definable by a distinguishing formula, hence it corresponds to an announcement. This allows us to consider finitely many strategies instead of considering infinitely many possible announcements: there are only finitely many non-equivalent (in terms of model updates) announcements for each finite model, and each of them has a corresponding distinguishing formula of some strategy.

Given a finite and bisimulation contracted model M_w and strategy $X \in S(G, w)$, a distinguishing formula δ_X for X can be obtained from Definition 9 as $\bigvee_{v \in X} \delta_v$.

Next, we show that agents know their strategies and thus can make corresponding announcements.

Proposition 4. Let M_w be a finite bisimulation contracted model, and $X \in S(a, w)$. Then $M_w \models K_a \delta_X$. Also, let $X_G := \bigcap_{i \in G} X_i$, where for all $i \in G$, $X_i \in S(i, w)$, be a group strategy, then $M_w \models \bigwedge_{i \in G} K_i \delta_{X_i}$.

Proof. We show just the first part of the proposition, since the second part follows easily. By the definition of a strategy, $X = [w_1]_a \cup \dots \cup [w_n]_a$, where $[w_1]_a, \dots, [w_n]_a$ belong to the set of a -equivalence classes. Since M_w is bisimulation contracted, for every equivalence class $[w_i]_a$ there is a corresponding distinguishing formula $\delta_{[w_i]_a}$. From the fact that for all $v \in [w_i]_a$, $M_v \models \delta_{[w_i]_a}$ (by Proposition 2 and Definition 9), we have that $M_v \models K_a \delta_{[w_i]_a}$. The same holds for other equivalence classes of a including the one with w , and we have $M_w \models (K_a \delta_{[w_1]_a} \vee \dots \vee K_a \delta_{[w_n]_a})$, which implies $M_w \models K_a (\delta_{[w_1]_a} \vee \dots \vee \delta_{[w_n]_a})$. Note that $\delta_{[w_1]_a} \vee \dots \vee \delta_{[w_n]_a}$ is a distinguishing formula of strategy X , so we can write $M_w \models K_a \delta_X$. Finally, having defined $K_j \delta_{X_j}$ for all $j \in G$, such that $X_j \in S(j, w)$, the group strategy $X_G = \bigcap_{j \in G} X_j$ in M_w corresponds to $M_w \models \bigwedge_{j \in G} K_j \delta_{X_j}$. \square

The following proposition states that given a strategy, the corresponding public announcement yields exactly the model with states specified by the strategy.

Proposition 5. Given a finite bisimulation contracted model $M = (W, \sim, V)$ and a strategy $X \in S(a, w)$, $W^{K_a \delta_X} = X$. More generally, $W^{\bigwedge_{i \in G} K_i \delta_{X_i}} = X_G$, where $X_G := \bigcap_{i \in G} X_i$ such that for all $i \in G$, $X_i \in S(i, w)$.

Proof. In order to prove that $W^{K_a \delta_X} = X$ for $X \in S(a, w)$, we need to show that for all $v \in W$, $v \in W^{K_a \delta_X}$ if and only if $v \in X$.

\Rightarrow : Let for some arbitrary $v \in W$, $v \in W^{K_a \delta_X}$. By the definition of $W^{K_a \delta_X}$, this means that $M_v \models K_a \delta_X$. Expanding δ_X , we get $M_v \models K_a \bigvee_{u \in X} \delta_u$. Since \sim_a is an equivalence relation, the latter implies that $M_v \models \bigvee_{u \in X} \delta_u$. Due to the fact that M is bisimulation contracted, $M_v \models \bigvee_{u \in X} \delta_u$ if and only if δ_v is one of δ_u 's, i.e. $M_v \models \delta_v$ and $v \in X$.

\Leftarrow : Let for some arbitrary $v \in W$, $v \in X$. Since X is a strategy, by Proposition 4 we have that $M_w \models K_a \bigvee_{u \in X} \delta_u$. Due to the fact that M is bisimulation contracted and $v \in X$, v also satisfies $K_a \bigvee_{u \in X} \delta_u$. Hence, $v \in W^{K_a \bigvee_{u \in X} \delta_u}$, or, equivalently, $v \in W^{K_a \delta_X}$.

Finally, let us consider the case of group G . It is clear that $W^{\bigwedge_{i \in G} K_i \delta_{X_i}} = \bigcap_{i \in G} W^{K_i \delta_{X_i}}$, where for all $i \in G$, $X_i \in S(i, w)$. Each $W^{K_i \delta_{X_i}}$ is equal to X_i for all $i \in G$. Hence, $\bigcap_{i \in G} W^{K_i \delta_{X_i}} = \bigcap_{i \in G} X_i$. The latter is equivalent to X_G by the definition of a group strategy. \square

We also show that true group announcements correspond to group strategies.

Proposition 6. Let M_w be a finite bisimulation contracted epistemic model, and $\psi \in \mathcal{L}_{EL}^G$ such that $M_w \models \psi$. Then there is a strategy $X \in S(G, w)$ such that $X = W^\psi$.

Proof. Assume that $M_w \models \psi$. Formula ψ is an abbreviation for $\bigwedge_{i \in G} K_i \psi_i$, where $\psi_i \in \mathcal{L}_{EL}$. Let us consider some particular $K_a \psi_a$. By the semantics we have that $M_w \models K_a \psi_a$ holds if and only if for all v reachable from w via a , $M_v \models \psi_a$. Note that all states reachable from the given one via a form an a -equivalence class $[w]_a$. In the same way, $K_a \psi_a$ may be true in other a -equivalence classes $[u]_a, \dots, [t]_a$. Hence, formula $K_a \psi_a$ holds in the union of these equivalence classes, i.e. it holds in $W^{K_a \psi_a} = [w]_a \cup \dots \cup [t]_a$. By Definition 10, $[w]_a \cup \dots \cup [t]_a$ is a strategy X of agent a in M_w .

Now assume that we have defined strategies X_i in M_w of all $i \in G$ in this fashion. From the fact that $W^{\bigwedge_{i \in G} K_i \psi_i} = \bigcap_{i \in G} W^{K_i \psi_i}$ we have that $W^{\bigwedge_{i \in G} K_i \psi_i} = \bigcap_{i \in G} W^{K_i \psi_i} = \bigcap_{i \in G} X_i$, and the latter is a group strategy $X \in S(G, w)$. \square

Now, let us reformulate semantics for the group and coalition announcement operators in terms of strategies.

Proposition 7. For a finite bisimulation contracted model M_w we have that

$$\begin{aligned} M_w \models \langle G \rangle \varphi & \text{ iff } \exists X \in S(G, w) : M_w^X \models \varphi, \\ M_w \models \langle\!\langle G \rangle\!\rangle \varphi & \text{ iff } \exists X \in S(G, w) \forall Y \in S(A \setminus G, w) : M_w^{X \cap Y} \models \varphi. \end{aligned}$$

Proof. *Case* $\langle G \rangle \varphi$. \Rightarrow : Assume that for some pointed model we have that $M_w \models \langle G \rangle \varphi$. By the semantics this means that $\exists \psi \in \mathcal{L}_{EL}^G : M_w \models \langle \psi \rangle \varphi$. The latter is equivalent to $M_w \models \psi$ and $M_w^\psi \models \varphi$. By Definition 3 and Proposition 6, this implies $M_w^X \models \varphi$ for some $X \in S(G, w)$.

\Leftarrow : Let $X \in S(G, w)$ be a group strategy such that $M_w^X \models \varphi$. Then, by Propositions 4 and 5, there is an announcement of distinguishing formulas by agents from G such that $X = W^{\bigwedge_{i \in G} K_i \delta_{X_i}}$, where $X_i \in S(i, w)$ for all $i \in G$. Hence $M_w^{\bigwedge_{i \in G} K_i \delta_{X_i}} \models \varphi$ and $M_w \models \bigwedge_{i \in G} K_i \delta_{X_i}$. The latter is equivalent to $M_w \models \langle G \rangle \varphi$ by the semantics.

Case $\langle\!\langle G \rangle\!\rangle \varphi$. \Rightarrow : Suppose that for some M_w it holds that $M_w \models \langle\!\langle G \rangle\!\rangle \varphi$. By the definition of semantics this is equivalent to $\exists \psi \in \mathcal{L}_{EL}^G, \forall \chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \wedge [\psi \wedge \chi] \varphi$.

Let $\mathcal{L}_{M_w}^{A \setminus G} = \{\chi \in \mathcal{L}_{EL}^{A \setminus G} \mid M_w \models \chi\}$ be the set of truthful announcements by $A \setminus G$ in M_w . Since $\mathcal{L}_{M_w}^{A \setminus G} \subset \mathcal{L}_{EL}^{A \setminus G}$, we have that $\exists \psi \in \mathcal{L}_{EL}^G, \forall \chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \wedge [\psi \wedge \chi] \varphi$ implies $\exists \psi \in \mathcal{L}_{EL}^G, \forall \chi \in \mathcal{L}_{M_w}^{A \setminus G} : M_w \models \psi \wedge \chi$ and $M_w^{\psi \wedge \chi} \models \varphi$. Hence, by Proposition 6, we have $M_w^{X \cap Y} \models \varphi$ for some $X \in S(G, w)$ and all $Y \in S(A \setminus G, w)$.

\Leftarrow : Assume that there is some strategy $X \in S(G, w)$ such that for all strategies $Y \in S(A \setminus G, w)$ it holds that $M_w^{X \cap Y} \models \varphi$. We need to show that

$$\exists \psi \in \mathcal{L}_{EL}^G, \forall \chi \in \mathcal{L}_{EL}^{A \setminus G} : M_w \models \psi \wedge [\psi \wedge \chi] \varphi.$$

Let $\psi = \bigwedge_{i \in G} K_i \delta_{X_i}$ (assuming $X = \bigcap_{i \in G} X_i$). By Proposition 4, $M_w \models \psi$, and by Proposition 5, $W^\psi = X$. Take an arbitrary $\chi \in \mathcal{L}_{EL}^{A \setminus G}$. If $M_w \not\models \chi$, then trivially $M_w \models \psi \wedge [\psi \wedge \chi] \varphi$ holds. If $M_w \models \chi$, then by Proposition 6, for some $Y \in S(A \setminus G, w)$, $Y = W^\chi$. Since for all $Y \in S(A \setminus G, w)$, $M_w^{X \cap Y} \models \varphi$, we have that $M_w^{\psi \wedge \chi} \models \varphi$, hence $M_w \models \psi \wedge [\psi \wedge \chi] \varphi$ holds for all $\chi \in \mathcal{L}_{EL}^{A \setminus G}$. The latter is equivalent to $M_w \models \langle\!\langle G \rangle\!\rangle \varphi$ by the semantics. \square

Sometimes we may be interested in situations where it is beneficial for agents to be as informative as possible (or, equivalently, leave as little uncertainty as possible). We recall the *Maxim of Quantity* postulated by Grice [24, 25]:

Make your contribution as informative as is required.

However, he also adds

Do not make your contribution more informative than is required.

What is as informative as required depends on the goal of the communication. So, in terms of epistemic logic, it depends on the epistemic goal to be satisfied in the model restriction resulting from the announcement, or from the sequence of announcements (as in a conversation consisting of various statements by different people, exactly the CAL setting). If the epistemic goal is full information on the value of all propositional variables, then the most informative announcement is the adequate announcement. However, there are other settings wherein the most informative announcement is not adequate. Typical settings of that kind are security protocols wherein the communicating principals want to be as informative as required (namely satisfying the information goal) but not more than that. They should guarantee safety: the eavesdropper should not be able to learn the information. The most informative announcement may then backfire. For example, a Bridge player had better not declare that she has the Queen of Hearts. This is very informative for her partner, but equally informative for the opposing team.

The type of announcements that fulfill the requirement that they are as informative as possible is defined in Definition 11.

Definition 11. Let M_w be a finite bisimulation contracted model. A *maximally informative announcement* by G is a formula $\psi \in \mathcal{L}_{EL}^G$ such that $w \in W^\psi$ and for all $\chi \in \mathcal{L}_{EL}^G$ such that $w \in W^\chi$ it holds that $W^\psi \subseteq W^\chi$. For finite models such an announcement always exists [3]. We will call the corresponding strategy $X \in S(G, w)$ *the strongest strategy* on a given model.

Intuitively, the strongest strategy is the smallest available strategy. Note that in a bisimulation contracted model M_w , the strongest strategy of agents G is $X = \bigcap_{i \in G} [w]_i$, that is agents' strategies consist of the single equivalence classes that include the current state.

In model $M_{15_a 5_b}$ in Figure 1 a 's strongest strategy is $\{15_a 5_b, 15_a 10_b, 15_a 15_b\}$, and b 's strongest strategy is $\{15_a 5_b, 10_a 5_b, 5_a 5_b\}$. So, the strongest strategy of group $\{a, b\}$ is the intersection of strongest strategies of agents from the group: $\{15_a 5_b, 15_a 10_b, 15_a 15_b\} \cap \{15_a 5_b, 10_a 5_b, 5_a 5_b\} = \{15_a 5_b\}$. Corresponding announcements are, respectively, $K_a(\delta_{15_a 5_b} \vee \delta_{15_a 10_b} \vee \delta_{15_a 15_b})$, $K_b(\delta_{15_a 5_b} \vee \delta_{10_a 5_b} \vee \delta_{5_a 5_b})$, and $K_a(\delta_{15_a 5_b} \vee \delta_{15_a 10_b} \vee \delta_{15_a 15_b}) \wedge K_b(\delta_{15_a 5_b} \vee \delta_{10_a 5_b} \vee \delta_{5_a 5_b})$.

4 Model Checking for CAL

Employing strategies allows for a rather simple model checking algorithm for CAL. We switch from quantification over an infinite number of epistemic formulas to quantification over a finite set of strategies (Section 4.1). Moreover, we show that if the target formula is a positive PAL formula, then model checking is even more effective (Section 4.2).

4.1 General Case

First, let us define the model checking problem.

Definition 12. Let M_w be a finite epistemic model, and $\varphi \in \mathcal{L}_{CAL} \cup \mathcal{L}_{GAL}$. *The model checking problem* is the problem to determine whether φ is satisfied in M_w .

We are going to solve this problem by providing an algorithm mc that, given a finite epistemic model $M = (W, \sim, V)$ and some formula φ , computes W^φ . Then the answer to the model checking problem for M_w will be *yes* if $w \in W^\varphi$, and *no* otherwise.

As a side effect, for formulas of the form $\langle G \rangle \psi$ or $\llbracket G \rrbracket \psi$ and for each state in W^φ , mc also writes out a strategy of G (a set of states) that ensures ψ . We could have defined mc to return a pair consisting of W^φ and a strategy (or an empty set, for formulas that are not of form $\langle G \rangle \varphi$ or $\llbracket G \rrbracket \varphi$) but we have decided to output the strategy as a side effect for ease of presentation.

Algorithm 1 takes a finite model M and $\varphi_0 \in \mathcal{L}_{CAL} \cup \mathcal{L}_{GAL}$ as an input, and returns W^{φ_0} , while also writing out a list of ‘witness’ strategies for group and coalition announcement operators. The case for GAL modalities is treated similarly to the model checking algorithm introduced in [1], apart from also printing out the witness strategy. The case for CAL modalities requires checking each strategy against all possible strategies by the opponents. Unlike the algorithm in [1] which runs in APTIME, we state a deterministic PSPACE algorithm.

But before providing the algorithm, we first need to introduce a function $next(G, M, w, X)$, that given a group of agents G , a model M , a state w and a strategy X , returns the next strategy X' in $S(G, w)$.

We assume that in the input $M = (W, \sim, V)$, \sim_a for each a is given as a set of equivalence classes of states, and that for each agent a there are n_a such classes (clearly $n_a \leq |W|$; observe also that this way of specifying the equivalence relation is linear rather than quadratic in $|W|$). Each strategy in $S(\{a\}, w)$ should include the equivalence class $[w]_a$. There are 2^{n_a-1} subsets of the set of the remaining \sim_a -equivalence classes, and hence $|S(\{a\}, w)| = 2^{n_a-1}$.

The set $S(G, w)$ can be ordered using the order on the set of agents A and on the equivalence classes of each agent a in G . For example, if an agent a has equivalence classes e_1, \dots, e_m in M , and e_1 contains w ,¹ then the order on $S(\{a\}, w)$ by shortest first and then lexicographically on $S(\{a\}, w)$ is:

$$e_1 = [w]_a$$

$$e_1 \cup e_2$$

$$e_1 \cup e_3$$

...

¹Clearly, w could belong to an equivalence class other than e_1 , e.g. to e_3 , but the point remains that the remaining unions of equivalence classes can be ordered in the same way.

$$e_1 \cup e_m$$

$$e_1 \cup e_2 \cup e_3$$

$$e_1 \cup e_2 \cup e_4$$

...

$$e_1 \cup e_2 \dots \cup e_m = W$$

Note that the first and the last strategies of a can be computed in time and space at most linear in the size of the model. Given an arbitrary element X in this order, the next one in the order (the function $next(\{a\}, M, w, X)$) can be computed in time and space polynomial in the size of the model. For a union of length j , we first check whether the last element can be ‘incremented’ (whether it is not e_1 or e_m) and if yes, increment it. If it cannot be incremented, then we check if the element $j - 1$ can be incremented (if it is not e_1 or e_{m-1}). If it can be incremented to the next equivalence class e' , we increment $j - 1$ and change j to e'' where e'' follows e' in the order of equivalence classes. If it cannot be incremented, we repeat until we either produce the next union of length j or we produce the first union of length $j + 1$ which is $e_1 \cup e_2 \cup \dots \cup e_{j+1}$.

Similarly, given the order on agents in G , say a_1, \dots, a_k , each agent’s strategies, $s_1^i, \dots, s_{N_i}^i$, where $N_i = 2^{n_i-1}$, the set $S(G, w)$ can be ordered lexicographically (below, $N_i = 2^{n_i-1}$, and $s_{N_i}^i = W$ is the last strategy of agent i):

$$s_1^1 \cap \dots \cap s_1^k = \bigcap_{i \in G} [w]_i$$

$$s_1^1 \cap \dots \cap s_1^{k-1} \cap s_2^k$$

...

$$s_{N_1}^1 \cap \dots \cap s_{N_k}^k = W$$

Again, the first and the last strategies of G can be computed in time at most linear in the size of the model, and the next element in time and space polynomial in the size of the model. Similarly to the single agent case, if the k th agent’s strategy X is not W , we increment it by calling $next(\{a_k\}, M, w, X)$, else we attempt to increment the strategy of a_{k-1} and reset the strategy of a_k to its first strategy $[w]_{a_k}$, etc.

For a group G of k agents, each with $2^{n_{a_i}-1}$ strategies, we have that $|S(G, w)| = 2^{\sum_k n_{a_i}-1} \leq 2^{|W|-1}$. Hence the size of $S(G, w)$ is bounded by the exponential in the size of the model (although not in $|G|$). A straightforward model-checking algorithm would generate $S(G, w)$ (and $S(A \setminus G, w)$ for coalition announcements) and iterate over it to check if the group or coalition announcement formula is satisfied. However generating $S(G, w)$ explicitly requires exponential amount of space. Instead of use the function $next(G, M, w, X)$ to generate the strategy that follows X in the ordering of $S(G, w)$. Generating and testing

strategies one at a time only requires polynomial amount of space. For technical convenience, we also define

$$\text{next}(G, M, w, \emptyset) = \bigcap_{i \in G} s_1^i$$

(the first strategy in $S(G, w)$ follows \emptyset) and

$$\text{next}(G, M, w, \bigcap_{i \in G} s_{N_i}^i) = \emptyset$$

(calling *next* on the last strategy returns \emptyset). Note that $\emptyset \notin S(G, w)$.

Algorithm 1 $mc(M, \varphi_0)$

```

1: function  $mc(M, \varphi_0)$ 
2:   case  $\varphi_0 = p$ 
3:     return  $V(p)$ 
4:   case  $\varphi_0 = \neg\varphi$ 
5:     return  $(W \setminus mc(M, \varphi))$ 
6:   case  $\varphi_0 = \varphi \wedge \psi$ 
7:     return  $mc(M, \varphi) \cap mc(M, \psi)$ 
8:   case  $\varphi_0 = K_a\varphi$ 
9:     return  $\{w \mid [w]_a \subseteq mc(M, \varphi)\}$ 
10:  case  $\varphi_0 = [\psi]\varphi$ 
11:    return  $(W \setminus mc(M, \psi)) \cup mc(M^\psi, \varphi)$ 
12:  case  $\varphi_0 = \langle G \rangle\varphi$ 
13:    compute the bisimulation contraction  $\|M\| = (\|W\|, \|\sim\|, \|V\|)$ 
14:     $\rho \leftarrow \emptyset$ 
15:    for  $w \in W$  do
16:       $X \leftarrow \emptyset$ 
17:       $found \leftarrow false$ 
18:      while  $\text{next}(G, \|M\|, [w], X) \neq \emptyset \wedge found = false$  do
19:         $X \leftarrow \text{next}(G, \|M\|, [w], X)$ 
20:        if  $[w] \in mc(\|M\|^X, \varphi)$  then
21:           $found \leftarrow true$ 
22:           $\rho \leftarrow \rho \cup \{w\}$ 
23:          print $(\langle G \rangle\varphi, \|M\|, [w], X)$ 
24:    return  $\rho$ 
25:  case  $\varphi_0 = \langle\langle G \rangle\rangle\varphi$ 
26:    compute the bisimulation contraction  $\|M\| = (\|W\|, \|\sim\|, \|V\|)$ 
27:     $\rho \leftarrow \emptyset$ 
28:    for  $w \in W$  do
29:       $X \leftarrow \emptyset$ 
30:       $found \leftarrow false$ 

```

```

31:         while  $next(G, \|M\|, [w], X) \neq \emptyset \wedge found = false$  do
32:              $X \leftarrow next(G, \|M\|, [w], X)$ 
33:              $check \leftarrow true$ 
34:              $Y \leftarrow \emptyset$ 
35:             while  $next(A \setminus G, \|M\|, [w], Y) \neq \emptyset$  do
36:                  $Y \leftarrow next(A \setminus G, \|M\|, [w], Y)$ 
37:                  $check \leftarrow check \wedge ([w] \in mc(\|M\|^{X \cap Y}, \varphi))$ 
38:             if  $check$  then
39:                  $\rho \leftarrow \rho \cup \{w\}$ 
40:                 print $(\langle G \rangle \varphi, \|M\|, [w], X)$ 
41:                  $found \leftarrow true$ 
42:         return  $\rho$ 

```

Now we show the correctness of the algorithm.

Proposition 8. Let M_w be a finite epistemic model, and $\varphi \in \mathcal{L}_{CAL} \cup \mathcal{L}_{GAL}$. Then $w \in mc(M, \varphi)$ if and only if $M_w \models \varphi$.

Proof. The proof is by induction on the complexity of φ . Boolean cases are straightforward and we omit them. In the proofs for cases $\langle G \rangle \varphi$ and $\langle\!\langle G \rangle\!\rangle \varphi$, we use Proposition 7.

Case $[\psi]\varphi$. Let $w \in mc(M, [\psi]\varphi)$. This is equivalent to $w \in (W \setminus mc(M, \psi)) \cup mc(M^\psi, \varphi)$ by line 11 of the algorithm. The latter is equivalent to $M_w \models \neg\psi$ or $M_w^\psi \models \varphi$ by the induction hypothesis. This is equivalent to $M_w \models [\psi]\varphi$ by the semantics.

Case $\langle G \rangle \varphi$. \Rightarrow : Suppose $w \in mc(M, \langle G \rangle \varphi)$. By lines 18-22 this means that for some strategy $X \in S(G, [w])$, $[w] \in mc(\|M\|^X, \varphi)$. By the induction hypothesis, $\|M\|_{[w]}^X \models \varphi$, and $\|M\|_{[w]} \models \langle G \rangle \varphi$ by the semantics. The latter implies $M_w \models \langle G \rangle \varphi$.

\Leftarrow : Let $M_w \models \langle G \rangle \varphi$, which is equivalent to $\|M\|_{[w]} \models \langle G \rangle \varphi$, and means that there is some strategy $X \in S(G, [w])$, such that $\|M\|_{[w]}^X \models \varphi$. By the induction hypothesis, the latter holds if and only if $[w] \in mc(\|M\|^X, \varphi)$. By lines 18-22, we have that $w \in mc(M, \langle G \rangle \varphi)$.

Case $\langle\!\langle G \rangle\!\rangle \varphi$. \Rightarrow : Suppose $w \in mc(M, \langle\!\langle G \rangle\!\rangle \varphi)$. By lines 32-39, this means that there exists strategy $X \in S(G, [w])$ such that for all strategies $Y \in S(A \setminus G, [w])$, $[w] \in mc(\|M\|^{X \cap Y}, \varphi)$. By the induction hypothesis, $\exists X \in S(G, [w])$, $\forall Y \in S(A \setminus G, [w])$: $\|M\|_{[w]}^{X \cap Y} \models \varphi$, which is $\|M\|_{[w]} \models \langle\!\langle G \rangle\!\rangle \varphi$ by Proposition 7. The latter is equivalent to $M_w \models \langle\!\langle G \rangle\!\rangle \varphi$ by Corollary 1.

\Leftarrow : Let $M_w \models \langle\!\langle G \rangle\!\rangle \varphi$, which is equivalent to $\|M\|_{[w]} \models \langle\!\langle G \rangle\!\rangle \varphi$ by Corollary 1. According to Proposition 7, this in turn is equivalent to $\exists X \in S(G, [w])$, $\forall Y \in S(A \setminus G, [w])$: $\|M\|_{[w]}^{X \cap Y} \models \varphi$. By the induction hypothesis, the latter holds if and only if there is $X \in S(G, [w])$ such that for all $Y \in S(A \setminus G, [w])$, $[w] \in mc(\|M\|^{X \cap Y}, \varphi)$. By lines 32-39, we have that $w \in mc(M, \langle\!\langle G \rangle\!\rangle \varphi)$. \square

Proposition 9. Model checking for CAL is PSPACE-complete.

Proof. All the cases of the model checking algorithm apart from the case for $\langle\!\langle G \rangle\!\rangle$ (and $\langle G \rangle$ for GAL) require polynomial time, both in the size of the model and the size of the formula (hence, polynomial space as a consequence).

The cases for $\langle G \rangle$ and $\langle\!\langle G \rangle\!\rangle$ generate and test exponentially many strategies. The running time of the algorithm is therefore exponential in the size of the model (but polynomial in the size of the formula).

However the cases for $\langle G \rangle$ and $\langle\!\langle G \rangle\!\rangle$ use only polynomial amount of *space*. Observe that $\text{next}(G, M, w, X)$ can be implemented to generate and return the successor strategy of X in time and space polynomial in M and G . Each check of a particular strategy can be computed using only polynomial amount of space to represent $\|M\|_{[w]}$ (which contains at most the same number of states as the input model M , and can be computed in polynomial time (see Appendix A)) and the result of the update (which at most the size of $\|M\|_{[w]}$) and make a recursive call to check whether φ holds in the update.

Hardness can be obtained by a slight modification of the proof of PSPACE-hardness of the model-checking problem for GAL in [1]. The proof encodes satisfiability of a quantified boolean formula (QBF) as a problem whether a particular GAL formula is true in a model corresponding to the QBF. We highlight just some parts of the proof from [1]. Given some QBF $\Psi := Q_1x_1 \dots Q_nx_n\Phi(x_1, \dots, x_n)$, the authors construct a model that depends on the number of variables in the formula. We have depicted the model in Figure 3, wherein those variables possibly indexed with 0 or 1 have become the names of the states.

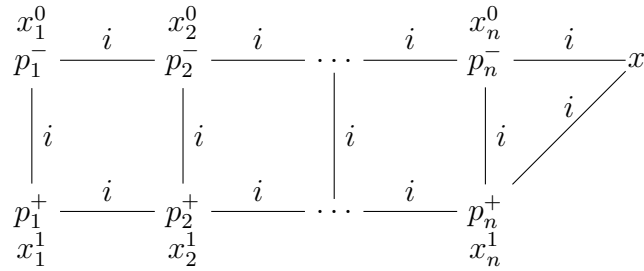


Figure 3: Model M that corresponds to a QBF.

Apart from agent i , whose relation is universal, there is also agent g , whose relation is the identity. Next, the authors define properties q_j ‘only one of x_j^0 and x_j^1 is in the model’ and r_j ‘both x_j^0 and x_j^1 are in the model’. These properties are used to recursively define a GAL formula $\psi(\Psi)$ that will be then evaluated in model $M_x \models \psi(\Psi)$. An example of a corresponding GAL formula for the given QBF $\forall x_1 \exists x_2 \forall x_3 : \Phi(x_1, x_2, x_3)$ is $K_i[g](q_1 \wedge r_2 \wedge r_3 \rightarrow \widehat{K}_i\langle g \rangle(q_1 \wedge q_2 \wedge r_3 \wedge K_i[g](q_1 \wedge q_2 \wedge q_3 \rightarrow \Phi(\widehat{K}_ip_1^+, \widehat{K}_ip_2^+, \widehat{K}_ip_3^+))))$.

For our proof, however, it is enough to notice the following. Since the encoding uses only two agents: an omniscient g and a universal i , we can replace $[g]$ and $\langle g \rangle$ with $\llbracket g \rrbracket$ and $\langle\!\langle g \rangle\!\rangle$ (since i ’s only strategy is equivalent to \top and no other GAL operators are used in the encoding) and obtain a CAL encoding. \square

4.2 Positive Case

In this section we demonstrate the following result: if in a given formula the subformulas within the scope of coalition and group announcement operators are positive PAL formulas, then complexity of model checking is polynomial.

Allowing coalition announcement modalities to bind only positive formulas is a natural restriction. Positive formulas have a special property: if the sum of knowledge of agents in G (their distributed knowledge) includes a positive formula φ , then φ can be made common knowledge by a group or coalition announcement by G . Formally, for a positive φ , $M_w \models D_G\varphi$ implies $M_w \models \langle\langle G \rangle\rangle C_G\varphi$, where D_G stands for distributed knowledge which is interpreted by the intersection of all \sim_a relations, and C_G stands for common knowledge which is interpreted by the transitive closure of the union of all \sim_a relations. See [17], and also [5] where the process of making distributed knowledge common knowledge is called *resolving* distributed knowledge. In other words, positive epistemic formulas can always be resolved by cooperative communication.

Negative formulas do not have this property. For example, it can be distributed knowledge of agents a and b that p and $\neg K_b p$: $D_{\{a,b\}}(p \wedge \neg K_b p)$. However it is impossible to achieve common knowledge of this formula: $C_{\{a,b\}}(p \wedge \neg K_b p)$ is inconsistent, since it implies both $K_b p$ and $\neg K_b p$. Going back to the example in Section 2.1, it is distributed knowledge of a and b that $K_a 15_a$ and $K_b 5_b$. Both formulas are positive and can be made common knowledge if a and b honestly report the amount of money they have. However it is also distributed knowledge that $\neg K_a 5_b$ and $\neg K_b 15_a$. The conjunction

$$K_a 15_a \wedge K_b 5_b \wedge \neg K_a 5_b \wedge \neg K_b 15_a$$

is distributed knowledge, but it cannot be made common knowledge for the same reasons as above.

We should also observe that positive formulas are maybe not as rare as it may appear on first sight. In the first place, in models where all states have different valuations, every announcement is equivalent to the disjunction of the characteristic formulas of depth 0 in the states in the denotation of the announcement. In particular, this is the case for the model in Figure 1.

However, in the second place, there are still other cases where announcement formulas are equivalent to positive formulas on some given model. This is not well-explored territory. A very relevant result by Van Benthem is that on finite models any epistemic formula ψ is equivalent to a formula φ that remains true after being announced. Such formulas φ are now often known as *successful* formulas [8, 17] (the term employed in [8] is *persistent*).

The formula constructed in Van Benthem's proof is a disjunction of characteristic formulas of states in the original and in the restricted models. This successful formula contains diamonds \widehat{K}_i and may not be positive (another problematic issue is that it also contains common knowledge modalities). The standard example of a successful formula that is not positive is the formula $\neg K_a p$. However, this and similar constructions may well lead to expand the use of the positive fragment. It is further relevant to observe that such positive formulas are a good candidate to characterise what are known as the *preserved*

formulas (those that remain true after any update, see Definition 13 below), which is also shown in [8], but for the slightly smaller positive fragment excluding the clause $[\neg\psi]\varphi$ for announcements given below.

The positive formulas are also relevant in an entirely different way for logics with quantification over announcements, namely in the logic called APAL⁺ wherein the quantification is over positive formulas only (in contrast to the situation investigated in this section, as the CAL quantifier is over all known formulas, which need not be positive, whereas the formula bound by the CAL quantifier must be positive). This logic is investigated in [16]. It is incomparable in expressivity to APAL, and it is also reputed to be decidable. No version of CAL quantifying over known *positive* formulas has been investigated to our knowledge.

Definition 13. The language \mathcal{L}_{PAL^+} of the positive fragment of public announcement logic PAL is defined by the following BNF:

$$\varphi, \psi ::= p \mid \neg p \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid K_a \varphi \mid [\neg\psi]\varphi,$$

where $p \in P$ and $a \in A$.

Definition 14. Formula φ is *preserved under submodels* if for any models M and N , $N \subseteq M$ and $M_w \models \varphi$ implies $N_w \models \varphi$.

A known result that we use in this section states that formulas of \mathcal{L}_{PAL^+} are preserved under submodels [17].

Proposition 10. Let M_w be a finite epistemic model, and let $\varphi \in \mathcal{L}_{CAL} \cup \mathcal{L}_{GAL}$ be a formula such that for all its subformulas of form $\langle G \rangle \psi$ and $\langle G \rangle \psi$, ψ belongs to the positive fragment \mathcal{L}_{PAL^+} . It is possible to decide by means of a deterministic algorithm working in polynomial time whether $M_w \models \varphi$.

Proof. For positive formulas, we can replace Algorithm 1 by Algorithm 2.

Algorithm 2 Model checking for positive formulas

```

1: function  $mcp(M, \varphi_0)$ 
2:   case  $\varphi_0 = p, \neg\varphi, \varphi \wedge \psi, K_a \varphi, [\psi]\varphi$ 
3:     as Algorithm 1
13:  case  $\varphi_0 = \langle G \rangle \varphi$ 
14:    compute the bisimulation contraction  $\|M\| = (\|W\|, \|\sim\|, \|V\|)$ 
15:     $\rho \leftarrow \emptyset$ 
16:    for  $w \in W$  do
17:       $X \leftarrow$  the strongest strategy of  $G$  in  $[w]$ 
18:      if  $[w] \in mcp(\|M\|^X, \varphi)$  then
19:         $\rho \leftarrow \rho \cup \{w\}$ 
20:      print $(\langle G \rangle \varphi, \|M\|, [w], X)$ 
21:    return  $\rho$ 
22:  case  $\varphi_0 = \langle G \rangle \varphi$ 

```

```

23:     compute the bisimulation contraction  $\|M\| = (\|W\|, \|\sim\|, \|V\|)$ 
24:      $\rho \leftarrow \emptyset$ 
25:     for  $w \in W$  do
26:          $X \leftarrow$  the strongest strategy of  $G$  in  $[w]$ 
27:         if  $[w] \in mcp(\|M\|^X, \varphi)$  then
28:              $\rho \leftarrow \rho \cup \{w\}$ 
29:             print( $\langle G \rangle \varphi, \|M\|, [w], X$ )
30:     return  $\rho$ 

```

For all subformulas of φ_0 , the algorithm runs in polynomial time. Consider the modified call for $\langle G \rangle \varphi$ and $\langle\!\langle G \rangle\!\rangle \varphi$. Instead of checking all possible strategies as in the general case, it requires constructing a single update model given a single (strongest) strategy, which is a simple case of restricting the input model to the set of states in the strategy. This can be done in polynomial time. Then we call the algorithm on the updated model for φ , which by assumption requires polynomial time. \square

Observe that the cases of CAL and GAL modalities for the positive fragment are treated in an identical way: we check whether the strongest strategy of G can be used to make the goal formula true. Intuitively, this is because every positive formula that can be made true with any strategy, can be made true with the strongest strategy. And in the case of CAL, the announcement by the opponents does not matter, since $\langle\!\langle G \rangle\!\rangle \varphi$ implies $\langle G \rangle \varphi$, and any further restrictions of a model do not change the valuation of positive φ .

Now, let us show that Algorithm 2 is correct.

Proposition 11. Let M_w be a finite epistemic model, and $\varphi \in \mathcal{L}_{PAL+}$. Then $w \in mcp(M, \varphi)$ if and only if $M_w \models \varphi$.

Proof. The proof is by induction on the size of φ .

Case $\langle G \rangle \varphi$. \Rightarrow : Suppose that $w \in mcp(M, \langle G \rangle \varphi)$. This means that $[w] \in mcp(\|M\|^X, \varphi)$, where X is the strongest strategy of G in $[w]$. By the induction hypothesis, we have that $\|M\|_{[w]}^X \models \varphi$, hence by Proposition 7 $\|M\|_{[w]} \models \langle G \rangle \varphi$. Since $\|M\|_{[w]}$ is bisimilar to M_w , $M_w \models \langle G \rangle \varphi$.

\Leftarrow : Let $M_w \models \langle G \rangle \varphi$. By Corollary 1 this is equivalent to $\|M\|_{[w]} \models \langle G \rangle \varphi$, and by Proposition 7 the latter is equivalent to $\exists X \in S(G, [w]) : \|M\|_{[w]}^X \models \varphi$. Since φ is positive and hence preserved under submodels, $\exists X \in S(G, [w]) : \|M\|_{[w]}^X \models \varphi$ implies $\|M\|_{[w]}^Y \models \varphi$, where Y is the strongest strategy of G . By the induction hypothesis we have that $[w] \in mcp(\|M\|^Y, \varphi)$. And by lines 17-19 of Algorithm 2, we conclude that $w \in mcp(M, \langle G \rangle \varphi)$.

Case $\langle\!\langle G \rangle\!\rangle \varphi$. \Rightarrow : Suppose that $w \in mcp(M, \langle\!\langle G \rangle\!\rangle \varphi)$. This means that $[w] \in mcp(\|M\|^X, \varphi)$, where X is the strongest strategy of G in $[w]$. By the induction hypothesis, we have that $\|M\|_{[w]}^X \models \varphi$. Since φ is positive, for all stronger updates $X \cap Y$, where $Y \in S(A \setminus G, [w])$, it holds that $\|M\|_{[w]}^{X \cap Y} \models \varphi$, which is $\|M\|_{[w]} \models \langle\!\langle G \rangle\!\rangle \varphi$ by Proposition 7. Finally, the latter model is bisimilar to M_w and hence $M_w \models \langle\!\langle G \rangle\!\rangle \varphi$.

\Leftarrow : Let $M_w \models \langle\langle G \rangle\rangle\varphi$. By Corollary 1 this is equivalent to $\|M\|_{[w]} \models \langle\langle G \rangle\rangle\varphi$, and by Proposition 7 the latter is equivalent to $\exists X \in S(G, [w]), \forall Y \in S(A \setminus G, [w]): \|M\|_{[w]}^{X \cap Y} \models \varphi$. Note that the trivial strategy $Y^\top \in S(A \setminus G, [w])$. Hence, $\exists X \in S(G, [w]), \forall Y \in S(A \setminus G, [w]): \|M\|_{[w]}^{X \cap Y} \models \varphi$ implies $\exists X \in S(G, [w]): \|M\|_{[w]}^{X \cap Y^\top} \models \varphi$, which is equivalent to $\|M\|_{[w]}^X \models \varphi$. Since φ is positive and hence preserved under submodels, $\|M\|_{[w]}^Z \models \varphi$, where Z is the strongest strategy of G in $[w]$. By the induction hypothesis, we have that $[w] \in mcp(\|M\|_{[w]}^Z, \varphi)$. And by lines 26-28 of Algorithm 2, we conclude that $w \in mcp(M, \langle\langle G \rangle\rangle\varphi)$. \square

5 Implementation and Experimental Results

There are several model checking tools for epistemic logic and its extensions. MCMAS [28] and MCK [22] can be used for verifying properties of epistemic temporal and strategy logics. DEMO [18] can be used to verify PAL properties, and SMDEL [10] to symbolically verify DEL properties. Recently DEMO has been used by Hagland [26] to check for the existence of group strategies in the Russian Cards problem [12]. There are no general purpose model checkers for GAL and CAL.

The model checker MCCAL is implemented in Java by Wang [33]. The code is available on <https://github.com/Twelvelines/MCCAL>. The model checker implementation is not optimised and is intended as a proof of concept. A non-trivial example from [20] is presented in the next section.

5.1 Households and Burglars: An Example

In the city of N^2 the local authorities have decided to gather information about, and publish statistics on, electricity consumption in each neighbourhood. Consumption information is submitted by each neighbourhood in the city, indicating the total number of households that have been using electricity in the last month. Data about neighbourhoods is public, and data about individual households is private, i.e., particular users of electricity are not revealed, but the total number of such users in the area is common knowledge. And there is a reason for such a requirement.

A group of local burglars is also interested in the public report on electricity consumption: they hope to deduce which households have not used electricity recently since it is an indication that property occupiers are not in their houses (most probably, they are on vacation). However, the burglars want to be certain that a house is empty, and will not risk burglary unless they know for sure that the property occupiers are away. They are also very reluctant to lurk around a neighbourhood trying to learn who is away, as such behaviour is very suspicious. Therefore, the only way to know about ‘vacant’ households is through the public energy consumption report.

²The work on the model checking of CAL started when three authors were based in Nottingham, and the fourth author was in Nancy. Moreover, city N is a well-known fictional location in Russian literature.

In \mathcal{N} , there is a small neighbourhood of only four houses: a , b , c , and d . They are situated around a park in a circular fashion such that neighbours on the left and on the right are equidistant. The park is quite large and the occupants of each house knows only their immediate neighbours on the left and on the right. Thus, for example, the occupant of c knows the occupants of b and d , and about their plans, but she is unaware of the plans of the occupants of a .

The epistemic model TES describing the neighbourhood containing a , b , c , and d is shown in Figure 4. In the model, the names of states indicate who is at home; for instance, 1001 means that the occupants of a and d are at home, and that the occupants of b and c are not. Burglars v (for ‘villains’) do not have any information regarding occupancy, and their epistemic relation is universal. We do not present the v -relation in the figure, for readability. We will refer to the occupant of house i as agent i .

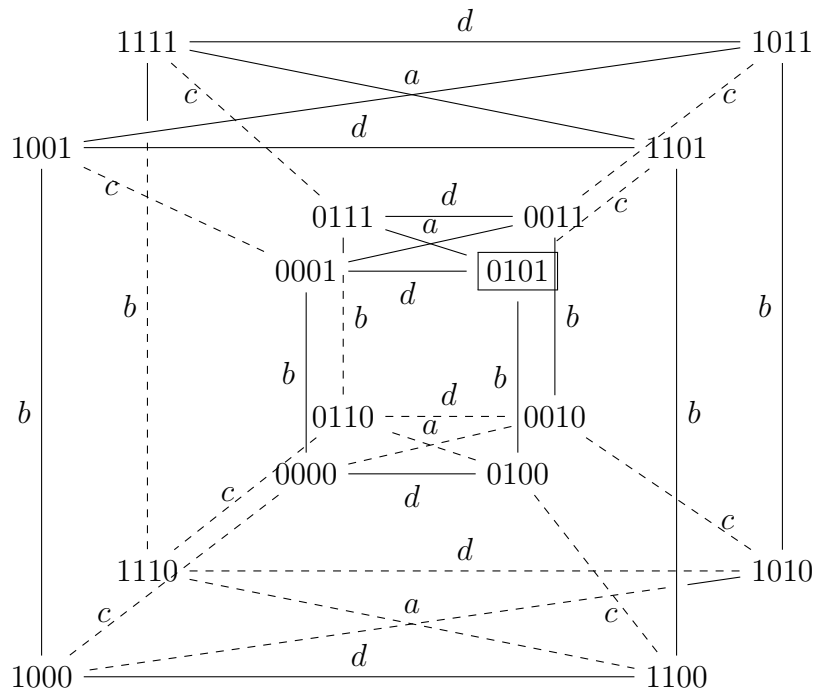


Figure 4: Model TES_{0101}

Let the actual state be 0101, and let 0101 also abbreviate $\neg p_a \wedge p_b \wedge \neg p_c \wedge p_d$, where p_i stands for ‘agent i is at home.’ Note that neither burglars nor the householders possess the full information about the neighbourhood: $TES_{0101} \models \neg(K_a 0101 \vee K_b 0101 \vee K_c 0101 \vee K_d 0101 \vee K_v 0101)$. Also note that householders are aware of their own state and of the states of their left- and right-hand-side neighbours, but not about the state of the furthest house. E.g. $TES_{0101} \models K_c \neg p_c \wedge K_c p_b \wedge K_c p_d \wedge \neg(K_c \neg p_a \vee K_c p_a)$.

The information that agents a, b, c and d want to submit is ‘two households in our neighbourhood have been using electricity.’ This sentence, however, should conform to the requirement that exact households remain unknown to the public outside the neighbour-

hood. We can express this goal as the formula

$$sofa := K_v \bigvee \left(\begin{array}{l} p_a \wedge p_b \wedge \neg p_c \wedge \neg p_d \\ p_a \wedge \neg p_b \wedge p_c \wedge \neg p_d \\ p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d \\ \neg p_a \wedge p_b \wedge p_c \wedge \neg p_d \\ \neg p_a \wedge p_b \wedge \neg p_c \wedge p_d \\ \neg p_a \wedge \neg p_b \wedge p_c \wedge p_d \end{array} \right) \wedge \neg \bigvee \left(\begin{array}{l} K_v p_a \vee K_v \neg p_a \\ K_v p_b \vee K_v \neg p_b \\ K_v p_c \vee K_v \neg p_c \\ K_v p_d \vee K_v \neg p_d \end{array} \right),$$

where *sofa* stands for ‘the state of affairs.’ A group announcement by agents to achieve this goal is when everyone announces ‘I know that if I have not been using electricity recently, then at least one of my neighbours on the left and on the right has, and if I have been using it, then one of the neighbours must be on vacation’. Formally, such an announcement can be expressed by the following formula

$$mis := \bigwedge \left(\begin{array}{l} K_a((\neg p_a \rightarrow (p_d \vee p_b)) \wedge (p_a \rightarrow \neg(p_d \wedge p_b))) \\ K_b((\neg p_b \rightarrow (p_a \vee p_c)) \wedge (p_b \rightarrow \neg(p_a \wedge p_c))) \\ K_c((\neg p_c \rightarrow (p_b \vee p_d)) \wedge (p_c \rightarrow \neg(p_b \wedge p_d))) \\ K_d((\neg p_d \rightarrow (p_c \vee p_a)) \wedge (p_d \rightarrow \neg(p_c \wedge p_a))) \end{array} \right),$$

where *mis* stands for ‘mutual informative statement.’

Thus we have that $TES_{0101} \models \langle mis \rangle sofa$. Since *mis* is an announcement of agents’ knowledge, we can conclude that there is an announcement by *a, b, c* and *d* such that *sofa* holds in the resulting model, i.e. $TES_{0101} \models \langle \{a, b, c, d\} \rangle sofa$. Result of updating TES_{0101} with *mis* is presented in Figure 5.

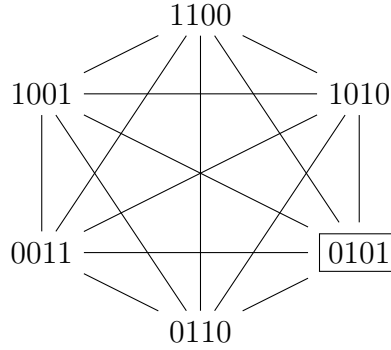


Figure 5: Model TES_{0101}^{mis}

All the relations in the model are *v* equivalence relations. Hence, indeed, in TES_{0101}^{mis} exactly two households have been using electricity recently, and although the public (and burglars as well) knows that fact, it cannot name particular houses that are ‘vacant’. A ‘side-effect’ of group announcement *mis* is that all residents in the neighbourhood know exactly who is on vacation, and it is common knowledge.

Note that we can state a fact stronger than $TES_{0101} \models \langle \{a, b, c, d\} \rangle sofa$. Since v 's relation is universal, they cannot prevent the group to make *sofa* true whatever they (i.e. v) announce. In other words, $TES_{0101} \models \langle \{a, b, c, d\} \rangle sofa$.

Interestingly, in this particular example even two agents can make an announcement such that *sofa* holds in the resulting model. Consider the following announcement by agents a and b :

$$mis_{a,b} := K_a((p_a \rightarrow \neg p_d) \wedge (\neg p_a \rightarrow p_d)) \wedge K_b((p_b \rightarrow \neg p_c) \wedge (\neg p_b \rightarrow p_c)).$$

The resulting updated model is shown in Figure 6 (all the relations are v -relations).

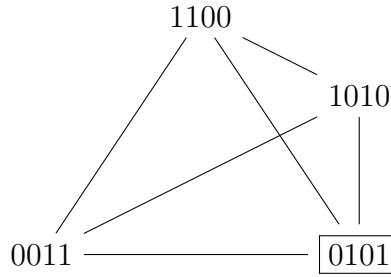


Figure 6: Model $TES_{0101}^{mis_{a,b}}$

The reader can verify that $TES_{0101}^{mis_{a,b}} \models sofa$, and hence $TES_{0101} \models \langle \{a, b\} \rangle sofa$. Note that compared to model TES_{0101}^{mis} (Figure 5), model $TES_{0101}^{mis_{a,b}}$ has fewer states. This means that householders gave a bit more information than necessary, but they still managed to inform authorities that exactly two households have been using electricity while not revealing the exact state of affairs.

Even though two householders can make a successful announcement, they must ensure that none of the other agents has been conspiring with burglars. For assume this is the case that agent c , for example, decides to reveal to burglars which houses are empty. She can pass the following information with a 's and b 's submission: $K_c(\neg p_c \wedge p_b \wedge p_d)$. This announcement made in conjunction with $mis_{a,b}$ results in a singleton model with 0101 as the only state. Moreover, whatever a and b announce, c always has an announcement to make *sofa* false in the resulting model (and, alas, to let the burglars know that she is on vacation). Formally, we have that $TES_{0101} \models \neg \langle \{a, b\} \rangle sofa$, or, equivalently, $TES_{0101} \models \langle \{a, b\} \rangle \neg sofa$. Hence, in this particular example, property householders should always cooperate if they want to inform authorities about electricity consumption and keep the burglars away.

We have seen that an announcement by two householders is enough to make *sofa* true. What about the single-agent case? As householders possess information about themselves and two closest neighbours, they do not know the actual state of the world, i.e. they do not have enough information about their furthest neighbour. However, it is possible for some agents to make an announcement such that it informs the public that at least two

of the households have been using electricity recently, and particular users and non-users remain incognito. Formally, such a target formula is as follows:

$$sofa_1 := K_v \bigvee \left(\begin{array}{l} p_a \wedge p_b \wedge \neg p_c \wedge \neg p_d \\ p_a \wedge \neg p_b \wedge p_c \wedge \neg p_d \\ p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d \\ \neg p_a \wedge p_b \wedge p_c \wedge \neg p_d \\ \neg p_a \wedge p_b \wedge \neg p_c \wedge p_d \\ \neg p_a \wedge \neg p_b \wedge p_c \wedge p_d \\ p_a \wedge p_b \wedge p_c \wedge \neg p_d \\ p_a \wedge p_b \wedge \neg p_c \wedge p_d \\ p_a \wedge \neg p_b \wedge p_c \wedge p_d \\ \neg p_a \wedge p_b \wedge p_c \wedge p_d \\ p_a \wedge p_b \wedge p_c \wedge p_d \end{array} \right) \wedge \neg \bigvee \left(\begin{array}{l} K_v p_a \vee K_v \neg p_a \\ K_v p_b \vee K_v \neg p_b \\ K_v p_c \vee K_v \neg p_c \\ K_v p_d \vee K_v \neg p_d \end{array} \right).$$

Agent a , for instance, can make $sofa_1$ true in TES_{0101} by announcing

$$mis_a := K_a((\neg p_a \rightarrow (p_b \wedge p_d)) \wedge (p_a \rightarrow (p_b \vee p_d))).$$

The result of such an announcement is presented in Figure 7 (relation v is universal).

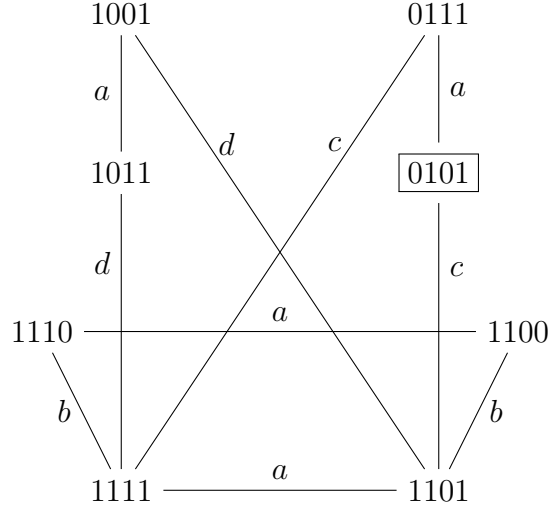


Figure 7: Model $TES_{0101}^{mis_a}$

It is easy to verify that $TES_{0101}^{mis_a} \models sofa_1$. Hence, it also holds that $TES_{0101} \models \langle \{a\} \rangle sofa_1$, and, obviously, $TES_{0101} \models \neg \langle \{a\} \rangle sofa_1$.

5.2 Experiments

For trivial examples with two or three agents and two or three states, the running time of MCCAL is less than the time needed to print the output to the screen.

To have more interesting examples, we have checked the following formulas on the TES_{0101} model:

1. $[mis]sofa$
2. $\langle\{a, b, c, d\}\rangle sofa$
3. $\langle\!\langle\{a, b, c, d\}\rangle\!\rangle sofa$
4. $\langle\{a, b\}\rangle sofa$
5. $\neg\langle\!\langle\{a, b\}\rangle\!\rangle sofa$
6. $\langle\{a\}\rangle sofa_1$
7. $\neg\langle\!\langle\{a\}\rangle\!\rangle sofa_1$

The experiment was carried out on a quad-core 64-bit Processor running at 2.2 GHz with 16GB of memory. The results of model checking these formulas and the average runtime (including outputting lists of strategies) of 10 computations is presented in Table 1. The times taken to check $\langle\{a, b, c, d\}\rangle sofa$ and $\langle\!\langle\{a, b, c, d\}\rangle\!\rangle sofa$ are significantly longer because the current implementation explicitly computes the set of all strategies for $\{a, b, c, d\}$, and this set is larger than the set of $\{a, b\}$ and $\{a\}$ strategies. However, the set of strategies does not grow exponentially with the size of the group.

Formula	Runtime
$[mis]sofa$	$4ms$
$\langle\{a, b, c, d\}\rangle sofa$	$41s137ms$
$\langle\!\langle\{a, b, c, d\}\rangle\!\rangle sofa$	$37s883ms$
$\langle\{a, b\}\rangle sofa$	$496ms$
$\neg\langle\!\langle\{a, b\}\rangle\!\rangle sofa$	$3s475ms$
$\langle\{a\}\rangle sofa_1$	$211ms$
$\neg\langle\!\langle\{a\}\rangle\!\rangle sofa_1$	$1s907ms$

Table 1: Formulas and average runtime of MCCAL

For formulas with the outermost occurrence of diamond versions of GAL and CAL modalities (formulas 2, 3, 4, and 6), MCCAL returns the corresponding group and individual strategies. The output of the model checker for state 0101 is presented in Table 2.

The reader can verify that strategies for formulas $\langle\{a, b, c, d\}\rangle sofa$ and $\langle\!\langle\{a, b, c, d\}\rangle\!\rangle sofa$ are identical. Indeed, the only agent outside of group $\{a, b, c, d\}$ is v , whose relation is universal. For formula $\langle\!\langle\{a\}\rangle\!\rangle sofa_1$, strategy of the group consisting of a single agent coincides with the agent's individual strategy. Strategies in the table differ from the ones presented in Section 5.1. Our algorithm can be easily modified to return all successful strategies.

	Formula	Group strategy
	$\langle\{a, b, c, d\}\rangle_{sofa}$	$\{0101, 1001, 1010\}$
a	$\{1010, 1011, 1110, 1000, 1001, 0110, 1100, 0111, 0100, 0101\}$	
b	$\{1010, 1000, 1110, 1011, 1001, 1111, 0000, 0110, 1100, 0100, 0001, 0111, 1101, 0101\}$	
c	$\{1010, 1000, 0010, 1011, 1111, 1001, 0000, 1100, 0011, 1101, 0111, 0001, 0100, 0101\}$	
d	$\{1010, 1110, 1001, 0001, 1101, 0101\}$	
	$\langle\{\{a, b, c, d\}\}\rangle_{sofa}$	$\{0101, 1001, 1010\}$
a	$\{1010, 1011, 1110, 1000, 1001, 0110, 1100, 0111, 0100, 0101\}$	
b	$\{1010, 1000, 1110, 1011, 1001, 1111, 0000, 0110, 1100, 0100, 0001, 0111, 1101, 0101\}$	
c	$\{1010, 1000, 0010, 1011, 1111, 1001, 0000, 1100, 0011, 1101, 0111, 0001, 0100, 0101\}$	
d	$\{1010, 1110, 1001, 0001, 1101, 0101\}$	
	$\langle\{a, b\}\rangle_{sofa}$	$\{0101, 1010\}$
a	$\{1010, 0010, 1000, 0011, 0000, 0111, 0101\}$	
b	$\{1010, 1110, 1101, 1111, 0100, 0101\}$	
	$\langle\{a\}\rangle_{sofa_1}$	$\{1011, 1110, 1001, 1100, 0111, 0101\}$
a	$\{1011, 1110, 1001, 1100, 0111, 0101\}$	

Table 2: Formulas and generated strategies

6 Concluding Remarks

We have shown that the model checking problem for CAL is PSPACE-complete, just like the one for GAL [1] and APAL [6]. We also presented a model checker for both CAL and GAL formulas. An interesting direction for future work is to optimise the performance of MCCAL.

In the special case when formulas within scopes of coalition modalities are positive PAL formulas, the model checking problem is in P. The same result would apply to GAL and APAL; in fact, in those cases the formulas in the scope of group and arbitrary announcement modalities can belong to a larger positive fragment (the positive fragment of GAL and of APAL, respectively, rather than of PAL). The latter is due to the fact that GAL and APAL operators are purely universal, while CAL operators combine universal and existential quantification, and CAL does not appear to have a non-trivial positive fragment extending that of PAL.

An interesting special case we would like to consider in the future is the case of models where each state has a different assignment of propositional variables such that the models are already bisimulation contracted.

Acknowledgements

We thank anonymous IJCAI 2018 and KI 2018 referees for constructive comments, and IJCAI 2018 referees for finding an error in the earlier version of this paper. We thank

Brian Logan for helpful discussions about model checking CAL, and Brian Logan and Franco Raimondi for advice on the implementation of the model checker. We are also grateful to the two anonymous reviewers for their suggestions and careful reading of the manuscript. This work was carried out while Hans van Ditmarsch was affiliated to LORIA, CNRS, University of Lorraine, France.

Appendix: Bisimulation Contraction Algorithm

To state the bisimulation contraction algorithm, we need the following operations:

- for two sets X and Y ,

$$\mathit{split}(X, Y) = \begin{cases} \{X \cap Y, X \cap \bar{Y}\} & \text{if } X \cap Y \neq \emptyset, X \cap \bar{Y} \neq \emptyset \\ \{X\} & \text{otherwise} \end{cases}$$

We will refer to Y as a splitter of X if $\mathit{split}(X, Y) = \{X \cap Y, X \cap \bar{Y}\}$.

- If Q is a set of sets, and Y a set, then

$$\mathit{split}(Q, Y) = \bigcup_{X \in Q} \mathit{split}(X, Y)$$

We will refer to Y as a splitter of Q if for some $X \in Q$, $\mathit{split}(X, Y) = \{X \cap Y, X \cap \bar{Y}\}$.

- If Y is a set of states, and \sim_a an indistinguishability relation, $\sim_a(Y) = \{x \mid \exists y \in Y \text{ such that } x \sim_a y\}$ (the preimage of Y with respect to \sim_a). Clearly,

$$\sim_a(Y) = \bigcup_{y \in Y} [y]_a.$$

The algorithm below is essentially the naive version of the relational coarsest partition algorithm by [29]. It starts with all states placed in a single block of the partition, and repeatedly splits the blocks until the states in the same block are bisimilar (that is, until the blocks correspond to bisimulation equivalence classes).

The first loop of the algorithm makes sure that the states in the same block satisfy the same propositional variables. In this loop, each set $V(p)$ is used as a (potential) splitter. This enforces the **Atoms** condition of the bisimulation relation.

The second loop enforces the property that the Paige-Tarjan relational coarsest partition algorithm was designed to achieve: for every indistinguishability relation \sim_a , and every pair of blocks X, Y , either $X \subseteq \sim_a(Y)$, or $X \cap \sim_a(Y) = \emptyset$. This ensures that either every element of X has an a -successor in Y , or none do (the **Back** and **Forth** conditions of

bisimulation relation).³ We repeatedly use splitters of the form $\sim_a(Y)$ until the partition does not change.

The algorithm returns a partition of W into bisimulation equivalence classes, which corresponds to $\|W\|$.

Algorithm 3 Computing $\|W\|$ given $M = (W, \sim, V)$

```

1: function RCP( $M$ )
2:    $Q = \{W\}$ 
3:   for  $p \in P$  do
4:      $Q \leftarrow \text{split}(Q, V(p))$ 
5:   repeat
6:     pick  $Y \in Q$  and  $a \in A$  such that  $\sim_a(Y)$  is a splitter for  $Q$ 
7:      $Q \leftarrow \text{split}(Q, \sim_a(Y))$ 
8:   until there is no change to  $Q$ 
9:   return  $Q$ 

```

The algorithm runs in time polynomial in $|W|$, $|P|$ and $|\sim|$.

The first loop in the algorithm iterates over $|P|$ and in the worst case terminates when every block in Q is a singleton set, that is, the size of Q at the end of the loop is at most $|W|$, so at most $|W|$ splits are performed. This means that the time complexity of the first loop is $O(|W| \times |P|)$.

If sets $[w]_a$ are given as part of the model, then computing the splitters is $O(|\sim|)$, and the number of times the splits are performed is $O(|W|)$ again. So the time complexity of the second loop is $O(|W| \times |\sim|)$.

The total time complexity of the algorithm is $O(|W| \times (|P| + |\sim|))$. A more efficient (logarithmic in $|W|$) version is possible as shown by [29], but here we state the simplest polynomial algorithm which is implemented in the model-checking tool.

References

- [1] Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *Journal of Applied Logic*, 8(1):62–81, 2010.
- [2] Thomas Ågotnes and Hans van Ditmarsch. Coalitions and announcements. In Lin Padgham, David C. Parkes, Jörg P. Müller, and Simon Parsons, editors, *Proceedings of the 7th AAMAS*, pages 673–680. IFAAMAS, 2008.
- [3] Thomas Ågotnes and Hans van Ditmarsch. What will they say? - public announcement games. *Synthese*, 179(Supplement-1):57–85, 2011.

³Paige and Tarjan worked with a single edge relation E rather than a set of relations \sim_a . They used notation $E^{-1}(Y)$ for the preimage of Y with respect to E , but for symmetric \sim , $\sim_a(Y) = \sim_a^{-1}(Y)$, so we can simplify the notation.

- [4] Thomas Ågotnes, Hans van Ditmarsch, and Tim French. The undecidability of quantified announcements. *Studia Logica*, 104(4):597–640, 2016.
- [5] Thomas Ågotnes and Yi N. Wáng. Resolving distributed knowledge. *Artificial Intelligence*, 252:1–21, 2017.
- [6] Philippe Balbiani, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima. ‘Knowable’ as ‘known after an announcement’. *Review of Symbolic Logic*, 1(3):305–334, 2008.
- [7] Johan van Benthem. Dynamic odds and ends. Technical report, University of Amsterdam, 1998. ILLC Research Report ML-1998-08.
- [8] Johan van Benthem. “One is a lonely number”: logic and communication. In Zoé Chatzidakis, Peter Koepke, and Wolfram Pohlers, editors, *Logic Colloquium ‘02*, volume 27 of *Lecture Notes in Logic*, pages 96–129. Cambridge University Press, 2006.
- [9] Johan van Benthem. *Logic in Games*. MIT Press, 2014.
- [10] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic - S5 and beyond. *Journal of Logic and Computation*, 28(2):367–402, 2018.
- [11] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
- [12] Hans van Ditmarsch. The russian cards problem. *Studia Logica*, 75(1):31–62, 2003.
- [13] Hans van Ditmarsch, David Fernández-Duque, and Wiebe van der Hoek. On the definability of simulation and bisimulation in epistemic logic. *Journal of Logic and Computation*, 24(6):1209–1227, 2014.
- [14] Hans van Ditmarsch and Tim French. Quantifying over boolean announcements. *CoRR*, abs/1712.05310, 2020.
- [15] Hans van Ditmarsch, Tim French, and Rustam Galimullin. No finite model property for logics of quantified announcements. In Joseph Halpern and Andrés Perea, editors, *Proceedings of the 18th TARK*. EPTCS, 2021. To appear.
- [16] Hans van Ditmarsch, Tim French, and James Hales. Positive announcements. *Studia Logica*, <https://doi.org/10.1007/s11225-020-09922-1>, 2020.
- [17] Hans van Ditmarsch and Barteld Kooi. The secret of my success. *Synthese*, 153(2):339–339, 2006.

- [18] Jan van Eijck. DEMO – a demo of epistemic modelling. In Johan van Benthem, Dov Gabbay, and Benedikt Löwe, editors, *Interactive Logic. Proceedings of the 7th Augustus de Morgan Workshop*, volume 1 of *Texts in Logic and Games*, pages 305–363. Amsterdam University Press, 2007.
- [19] Tim French, Rustam Galimullin, Hans van Ditmarsch, and Natasha Alechina. Groups versus coalitions: On the relative expressivity of GAL and CAL. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th AAMAS*, pages 953–961. IFAAMAS, 2019.
- [20] Rustam Galimullin. *Coalition Announcements*. PhD thesis, University of Nottingham, 2019.
- [21] Rustam Galimullin, Natasha Alechina, and Hans van Ditmarsch. Model checking for coalition announcement logic. In Frank Trollmann and Anni-Yasmin Turhan, editors, *KI 2018: Advances in Artificial Intelligence*, volume 11117 of *LNCS*, pages 11–23. Springer, 2018.
- [22] Peter Gammie and Ron van der Meyden. MCK: model checking the logic of knowledge. In Rajeev Alur and Doron A. Peled, editors, *Proceedings of the 16th CAV*, volume 3114 of *LNCS*, pages 479–483. Springer, 2004.
- [23] Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 249–329. Elsevier, 2007.
- [24] Herbert Paul Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics 3: Speech Arts*, pages 41–58. Academic Press, 1975.
- [25] Herbert Paul Grice. *Studies in the Way of Words*. Harvard University Press, 1989.
- [26] Tor Hagland. Crossing hands in the russian cards problem, 2018. MSc Thesis, Department of Information Science and Media Studies, University of Bergen.
- [27] Jaakko Hintikka. *Knowledge and belief. An introduction to the logic of the two notions*. Cornell University Press, Ithaca, NY, 1962.
- [28] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. MCMAS: A model checker for the verification of multi-agent systems. In Ahmed Bouajjani and Oded Maler, editors, *Proceedings of the 21st CAV*, volume 5643 of *LNCS*, pages 682–688. Springer, 2009.
- [29] Robert Paige and Robert Endre Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

- [30] Rohit Parikh. The logic of games and its applications. In Marek Karplinski and Jan van Leeuwen, editors, *Topics in the Theory of Computation*, volume 24 of *Annals of Discrete Mathematics*, pages 111–139. Elsevier Science, Amsterdam, 1985.
- [31] Marc Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.
- [32] Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
- [33] Tuo Wang. Model checker for coalition announcement logic, 2019. BSc Thesis, School of Computer Science, University of Nottingham.