Model checking against arbitrary public announcement logic: A first-order-logic prover approach for the existential fragment

Tristan Charrier, Sophie Pinchinat, François Schwarzentruber

Model checking against arbitrary public announcement logic: A first-order-logic prover approach for the existential fragment

Tristan Charrier[1], Sophie Pinchinat[2], and François Schwarzentruber[3]

[1] Université de Rennes 1, tristan.charrier@irisa.fr
[2] IRISA, sophie.pinchinat@irisa.fr
[3] ENS Rennes, francois.schwarzentruber@ens-rennes.fr

Abstract. In this paper, we investigate the model checking problem of symbolic models against epistemic logic with arbitrary public announcements and group announcements. We reduce this problem to the satisfiability of Monadic Monadic Second Order Logic (MMSO), the fragment of monadic-second order logic restricted to monadic predicates. In particular, for the case of epistemic formulas in which all arbitrary and group announcements are existential, the proposed reduction lands in monadic first-order logic. We take advantage of this situation to report on few experiments we made with first-order provers.

1 Introduction

In a multi-robot system, agents collect knowledge from what they perceive with their sensors and from the information acquired from some communication channel [31,32]. In order to formalize the notion of knowledge, epistemic modal logics have been developed. For instance, Dynamic epistemic logic [9,43] aims at expressing properties about the knowledge of agents and at modeling information change in multi-agent settings. Public announcement logic PAL [38] is a noticeable fragment of Dynamic epistemic logic, where possible events are public announcements. Since then, variants/extensions of PAL have been developed: typically, arbitrary public announcement logic APAL [6] and group announcement logic GAL [2]. The family of announcement logics has been the subject of much work as they open the way to formal reasoning in many practical applications. We here mention a few, at the intuitive level only. For example, such logics enable one to reason about human/robot interaction via a public channel of communication: message exchanges between robots can be modeled by public announcements when there is common knowledge of the reliability of the network and when it is assumed that messages are received instantaneously [32]. Announcement logics, as well as dynamic epistemic logic, are also relevant in games [34]: in the Battleships, players publicly announce that there is a ship in a given cell. In card games, players often publicly show some cards to other players or announce something. Some issues in security may also be approached with announcement logic: for example, one may wish to verify that no announcement leads the system to a critical/bad state, say, where Intruder knows some secret [17]. Finally, gossip-based algorithms in distributed systems, where agents privately share their secrets in order to achieve shared knowledge of all secrets[4], may be analyzed with announcement logic [28,41].

In order to get started with announcement logic, we develop the classic Russian card example.

Example 1 (Russian card [44,42]).

We consider three agents, a, b and c. Agent a has 3 cards in her hand, b has 3 cards in his hand and c has 1 card in his hand. The cards range from 1 to 7. Given a hand, say as in Figure 1, the question is whether a and b can publicly announce truthful facts so that they commonly know all players' hands but c not learning any card from a's or b's hands from the course of announcements.

In the case where a and b have 3 cards and c 1 card, it is shown in [44] that it is possible for a and b to share information about their hands in any possible configuration in one single public announcement for a and one single public announcement for b.

[4] in a minimal number of communications.

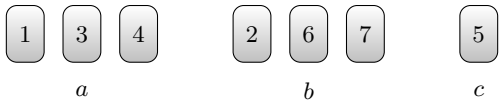

Fig. 1. Example of hands for the Russian cards puzzle

Of course, a cannot just announce what her hand is, because it would cause c to learn the content of her hand. The trick for a consists in announcing a set of possible hands such that b can deduce what a's hand is, and c cannot. In the example of Figure 1, if a announces the sentence (Δ) "My hand is either 134, 126, 367, 465 or 275", she ensures that for any possible configuration of hands for b and c, b will always be able to deduce a's hand and c will never deduce any card of a's hand. After a has announced (Δ), b actually knows all hands of the players. Therefore, b announces "c has card 5 is his hand" so that a knows all hands.

Regarding logics APAL and GAL, it has been proved that their satisfiability problem are undecidable ([3], [25]). It has been shown that the satisfiability problem with iterations over public announcements is undecidable too [36], so the satisfiability problem with any protocol is also undecidable. Nevertheless, these logics are very relevant for model checking, that is verifying that a given model satisfies a given property. The model checking problem is at the heart of this contribution. Additionally, the setting we consider is the one of *symbolic* models. These models are not specified in extension but described by means of all the possible valuations of a finite set of propositions (each valuation denotes a possible world) and the indistinguishability relations (one for each agent) are specified by *accessibility programs*.

We introduce a second example, the standard muddy children puzzle [43], and we pull its definition to a symbolic model. Both Russian cards and muddy children examples will be useful in the paper.

Example 2 (muddy children). We consider n children playing in their garden. Some of them have mud on their forehead, some have not. Each child can see the others' forehead[5], but she cannot see her own. We suppose that all children are honest and clever. Their father comes to them and says: "At least one of you has mud on her head.". Then he repeatedly asks "Does any one of you know for sure whether he/she is muddy?". He stops asking when at least one child tells that she knows.

The solution to this very classic puzzle is that if k children are muddy with $k \leq n$, no child knows its status before round k, and the muddy children know their status in round k [6].

Formally, the initial situation is modeled by a Kripke model containing all combinations of possible children's forehead's status, that is 2^n possible worlds. In a given situation/world, each child considers one other possible world that differs from the current one regarding her own forehead's status. Figure 2 shows the Kripke model for two agents. Proposition p_a stands for "a is muddy" while proposition p_b symmetrically stands for "b is muddy".

Because Kripke models may be large – in the muddy children example the model is exponential in the number of children – many symbolic representations have been considered in the model checking literature (see for example [5]) and more recently in epistemic logic [40,19,20]. We use here the notion of *symbolic accessibility relations* that we call *accessibility programs*, or simply *programs*, that can modify propositional variables. These programs are akin to a dialect used in PDL [24], called *DL-PA*, for "dynamic logic of propositional assignments" [8]. These programs turn out to be the natural way of defining Kripke models. For instance, for the muddy children puzzle, the program of agent a (resp. b) is: *Non-deterministically choose between setting the value of p_a (resp. p_b) to false or to true.* As observed in [20], the size of a symbolic Kripke model (that is the size needed to describe the collection of agent programs) may be exponentially smaller than the size any equivalent non-symbolic Kripke model [7]. Thus it is polynomial in the number of children in the muddy children's example.

[5] henceforth if there is mud.

[6] Clean children know their status during round $k+1$.

[7] For non-symbolic Kripke models, the size is the one of its graph.

The symbolic model checking of APAL was already studied in [19]. Its complexity was proved to be A_{pol}EXPTIME-complete, and NEXPTIME-complete when restricted to existential arbitrary announcements. Recall that the class A_{pol}EXPTIME [29,14,15] stands for the class of problems decided by alternating Turing machines [16] that run in exponential time but with only a polynomial number of alternations along the computation, hence it is in between EXPTIME and AEXPTIME (= EXPSPACE).

In this paper, instead of building specific algorithms for model checking symbolic models against arbitrary public announcement and group announcement logic (AGPAL, the natural combination of APAL and GAL), we bring closer this logic and first-order logic. More precisely:

1. We show a polynomial reduction from the symbolic model checking[8] against AGPAL to the satisfiability problem of the *monadic monadic second order logic*, written MMSO here, that is the fragment of monadic second order logic where all predicates in the formula are monadic.
2. We prove that this reduction leads to a reduction from the symbolic model checking of *existential* AGPAL[9] (∃AGPAL) to the satisfiability problem of *monadic first-order logic*, that we write MFO. This reduction is supported by the fact that the symbolic model checking against ∃AGPAL and the satisfiability problem of monadic first-order logic are both NEXPTIME-complete (see respectively [19] and [4], [33], [35]).
3. We build a set of benchmarks for FO provers and report on our experiments.

We claim that the relationship we establish between announcement logics and first-order logic cross-fertilizes two communities: the one in dynamic epistemic logic would benefit from the expertise of researchers in first-order provers in term of efficiency of algorithms and theorem proving techniques; the other community from first-order logic will collect new benchmarks that correspond to instances of the symbolic model checking problem of ∃AGPAL.

The article is organized as follows. In Section 2, we recall the setting of MMSO and MFO. Next, in Section 3, we describe the language AGPAL and its existential fragment ∃AGPAL. Sections 4 (resp. Section 5) is dedicated to the reduction of the symbolic model checking problem against AGPAL (resp. ∃AGPAL) to the satisfiability problem for MMSO (resp. MFO). In Section 6, we benefit from the use of FO provers to solve the symbolic model checking problem against ∃AGPAL, and report on our experiments. Finally, we open perspectives for future work in Section 7.

In the rest of this paper, we fix a countable set of atomic propositions $\mathtt{AP} = \{p, q, p_1, p_2, \ldots\}$.

2 Brief recall on first and second-order logics

Monadic monadic second-order logic MMSO and its fragment monadic first-order logic MFO are central in the proposed approach. These monadic fragments of MSO and FO respectively disallow the use of non-unary predicates and of function symbols: MMSO-formulas are thus monadic second-order formulas with first-order and second-order variables but with no occurrence of non-unary predicates; MFO-formulas have only first-order variables. The signature of MMSO mimics the set of atomic propositions $\mathtt{AP}$: to each atomic proposition $p \in \mathtt{AP}$, we introduce a corresponding unary predicate symbol $P(.)$[10].

A model $\mathscr{M}$ of MMSO is a structure $(D, (P^{\mathscr{M}})_{p\in\mathtt{AP}})$ where D is a non-empty domain and each $P^{\mathscr{M}} \subseteq D$. We will use the classical notation of the form $\mathscr{M}[...]$ for the model $\mathscr{M}$ extended with (first-order and second-order) variable assignments: for instance, $\mathscr{M}[\mathsf{x} \leftarrow e, \mathsf{y} \leftarrow e', \mathsf{X} \leftarrow D', \mathsf{Y} \leftarrow D'']$ is the model $\mathscr{M}$ in which first-order variables x and y are interpreted by element $e \in D$ and $e' \in D$ respectively, and second-order variables X and Y are interpreted by element $D' \subseteq D$ and $D'' \subseteq D$ respectively.

[8] a short way for model checking of symbolic models.
[9] the fragment of AGPAL with only existential arbitrary and group announcements.
[10] We take the convention that atomic propositions are written in lowercase while the corresponding predicates are written in uppercase.

Regarding the properties of MMSO and MFO, it is known that the satisfiability problem of a MFO-formula is NEXPTIME-complete [4,33]. Also, there are plenty of FO provers: Isabelle, iprover, Z3 [21], CVC4 [10]. In particular, the prover iprover won CASC 2016 in EPR division [39].

3 Background on arbitrary/group public announcement

In this section, we define the logic AGPAL that extends both arbitrary public announcement logic and group announcement logic, as well at its fragment ∃AGPAL. Moreover, we consider symbolic models to interpret these logics, and state the symbolic model checking problem.

3.1 Syntax of AGPAL

Let $\mathtt{AP}$ be a countable set of atomic propositions. Let $\mathtt{Agt}$ be a finite set of agents. We define the logic AGPAL that extends both arbitrary public announcement logic and group announcement logic, but we simply call it *announcement logic*.

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid \langle\varphi!\rangle\varphi \mid \langle\bullet!\rangle\varphi \mid \langle\bullet!_G\rangle\varphi$$

where p ranges over $\mathtt{AP}$ and a over $\mathtt{Agt}$. Formula $K_a\varphi$ reads as "agent a knows that φ holds". Construction $\langle\psi!\rangle\varphi$ reads as "ψ is true and after having announced ψ, formula φ holds". $\langle\bullet!\rangle\varphi$ reads as "there exists a true formula ψ such that makes φ true after announcing it". Formula $\langle\bullet!_G\rangle\varphi$ reads as "agents of group G can make φ hold by announcing at the same time each a formula she knows". In other words, it means that "there exists a true formula of the form $\bigwedge_{a\in G} K_a\psi_a$ such that make φ hold after announcing it". As usual, we write $(\varphi \vee \psi)$ for $\neg(\neg\varphi \wedge \neg\psi)$, $\hat{K}_a\varphi$ for $\neg K_a\neg\varphi$, $[\psi!]\varphi$ for $\neg\langle\psi!\rangle\neg\varphi$. We concisely write $\langle\psi!\rangle^n\varphi$ for $\langle\psi!\rangle\ldots\langle\psi!\rangle\varphi$ where the announcement of ψ takes place n times.

Example 3 (Muddy children with n children). Suppose that all children are muddy. Formula $\langle\bigvee_{a\in\mathtt{Agt}} p_a!\rangle\langle(\bigwedge_{a\in\mathtt{Agt}}\neg K_a p_a)!\rangle^n \bigwedge_{a\in\mathtt{Agt}} K_a p_a$ states that all children know that they are muddy after the father announces that one of them is muddy and then announces n times that no child knows that she is muddy. It is known that this formula holds in the initial situation of the muddy children puzzle.

Example 4 (Russian cards). We introduce propositions $p_{i,a}$ for "agent a has card i". Let $\mathtt{AP}_h$ be the set of all propositions $p_{i,a}$, $p_{i,b}$, $p_{i,c}$ for $i \in \{1,\ldots,7\}$.

Let S_7 be the set of all permutations of $\{1,\ldots,7\}$. Given $\boldsymbol{h} = (h_1, ..., h_7)$ an element of S_7, we define

$$\varphi_{Rh}(\boldsymbol{h}) = p_{h_1,a} \wedge p_{h_2,a} \wedge p_{h_3,a} \wedge p_{h_4,b} \wedge p_{h_5,b} \wedge p_{h_6,b} \wedge p_{h_7,c} \wedge \bigwedge_{p\in\mathtt{AP}_h\setminus\{p_{h_1,a},\ldots,p_{h_7,c}\}} \neg p.$$

$\varphi_{Rh}(\boldsymbol{h})$ describes a particular configuration $\boldsymbol{h}$ of the hands for the players. The rules of the game are defined by the formula $\varphi_R = \bigvee_{\boldsymbol{h}\in S_7} \varphi_{Rh}(\boldsymbol{h})$.

The following formula φ_G states that both a and b know the card configurations while c does not:

$$\varphi_G = \bigvee_{\boldsymbol{h}\in S_7} (K_a\varphi_{Rh}(\boldsymbol{h}) \wedge K_b\varphi_{Rh}(\boldsymbol{h}))) \wedge \bigwedge_{p\in\{p_{1,a},\ldots,p_{7,a},p_{1,b},\ldots,p_{7,b}\}} \neg(K_c p) \wedge \neg(K_c\neg p)$$

In the Russian card situation, the goal is to check that $\langle\bullet!_a\rangle\langle\bullet!_b\rangle\varphi_G$ holds.

3.2 Syntax of ∃AGPAL

We now define the fragment ∃AGPAL of AGPAL, where arbitrary and group announcement operators are only existential. Formally, ∃AGPAL is defined by the following grammar.

$$\begin{aligned}\exists\text{AGPAL} \ni \varphi &::= \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \hat{K}_a\varphi \mid \langle\varphi!\rangle\varphi \mid \langle\bullet!\rangle\varphi \mid \langle\bullet!_G\rangle\varphi \\ \psi &::= p \mid \neg\psi \mid \psi \vee \psi \mid K_a\psi\end{aligned}$$

where $p \in \mathtt{AP}$ and a is an agent.

Example 5. The formula $\langle \bullet!_a \rangle \langle \bullet!_b \rangle \varphi_G$ given in the Russian card Example is in ∃AGPAL.

Example 6. Formula $K_b \langle \bullet!_a \rangle K_c p$ is not in ∃AGPAL since $\langle \bullet!_a \rangle$ occurs after K_b. Formula $\hat{K}_b \langle \bullet!_a \rangle K_c p$ is in ∃AGPAL.

3.3 Semantics of AGPAL

Formulas of AGPAL are interpreted on classic Kripke models with the *possible world* semantics, widely used in logics of knowledge [23].

Definition 1. *A* Kripke model *is a tuple* $\mathcal{M} = (W, \{\xrightarrow{a}\}_{a \in \mathtt{Agt}}, V)$*, where:*

- W *is the non-empty set of* worlds*,*
- *for each* $a \in \mathtt{Agt}$*,* $\xrightarrow{a} \subseteq W \times W$ *is the* accessibility relation *for agent* a*,*
- $V : W \to 2^{AP}$ *is the* valuation *on worlds, that reveals the set of propositions that hold.*

For the sake of generality, we do not require the accessibility relations to be equivalence relations.

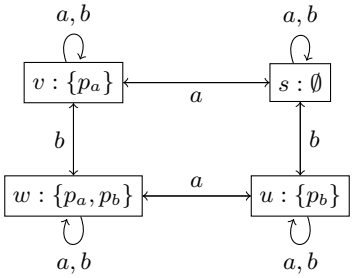

Fig. 2. Kripke model for the muddy children puzzle for two agents

Example 7 (muddy children). Figure 2 shows a Kripke model for muddy children with $n = 2$ agents. It has four worlds w, u, v, s. The arrows represent the agents' accessibility relations. For an arbitrary number n of agents, the Kripke model is $\mathcal{M} = (W, \{\xrightarrow{i}\}_{i \in \mathtt{Agt}}, V)$ where:

- $W = 2^{\{p_a \mid a \in \mathtt{Agt}\}}$;
- $\xrightarrow{a} = \{(w, u) \mid w \setminus \{p_a\} = u \setminus \{p_a\}\}$;
- $V(w) = w$.

This Kripke model is a graph containing 2^n nodes and $2^{n+1} \times |\mathtt{Agt}|$ edges.

Example 8 (Russian cards). A Kripke model corresponding to the Russian card puzzle is $\mathcal{M} = (W, \{\xrightarrow{a}\}_{a \in \mathtt{Agt}}, V)$ where:

- W is the set of valuations over $\mathtt{AP}_h$ that satisfy formula φ_R; where φ_R is defined in Example 4;
- $w \xrightarrow{a} u$ if $w \cap \{p_{i,a} \mid i \in \{1, \ldots, 7\}\} = u \cap \{p_{i,a} \mid i \in \{1, \ldots, 7\}\}$;
- $V(w) = w$.

Informally, W is the set of all distributions of cards, $w \xrightarrow{a} u$ if a holds the same cards in both worlds w and u, and the valuation $V(w)$ is given by w.

Back to the semantics of AGPAL, we now define the truth conditions for $\mathcal{M}, w \models \varphi$ (read as "formula φ is true in world w of model $\mathcal{M}$") and the restriction $\mathcal{M}^\psi$ of a model $\mathcal{M}$ to a formula ψ.

Definition 2. *We define $\mathcal{M}, w \models \varphi$ (read as "formula φ is true in world w of model $\mathcal{M}$") and $\mathcal{M}^\psi$ (the* ψ-restriction *of $\mathcal{M}$) by mutual induction:*

- $\mathcal{M}, w \models p$ *if* $p \in V(w)$;
- $\mathcal{M}, w \models (\varphi_1 \wedge \varphi_2)$ *if* $\mathcal{M}, w \models \varphi_1$ *and* $\mathcal{M}, w \models \varphi_2$;
- $\mathcal{M}, w \models \neg\varphi$ *if* $\mathcal{M}, w \not\models \varphi$;
- $\mathcal{M}, w \models K_a\varphi$ *if for all* u *such that* $w \xrightarrow{a} u$, $\mathcal{M}, u \models \varphi$;
- $\mathcal{M}, w \models \langle\psi!\rangle\varphi$ *if* $\mathcal{M}, w \models \psi$ *and* $\mathcal{M}^\psi, w \models \varphi$;
- $\mathcal{M}, w \models \langle\bullet!\rangle\varphi$ *if there exists a formula* ψ *without any occurrence of* $\langle\bullet!\rangle$ *or* $\langle\bullet!_G\rangle$ *such that* $\mathcal{M}, w \models \langle\psi!\rangle\varphi$;
- $\mathcal{M}, w \models \langle\bullet!_G\rangle\varphi$ *if there exist formulas* $(\psi_a)_{a\in G}$ *without any occurrence of* $\langle\bullet!\rangle$ *or* $\langle\bullet!_G\rangle$, *such that* $\mathcal{M}, w \models \langle\bigwedge_{a\in G} K_a\psi_a!\rangle\varphi$.

and $\mathcal{M}^\psi$ is the model $(W^\psi, \{\xrightarrow{a}{}^{\psi}\}_{i\in \mathtt{Agt}}, V^\psi)$ where

- $W^\psi = \{u \in W \mid \mathcal{M}, u \models \psi\}$ *(namely, only worlds satisfying* ψ *are preserved);*
- $\xrightarrow{a}{}^{\psi} = \xrightarrow{a} \cap (W^\psi \times W^\psi)$;
- V^ψ *is the restriction of* V *to* W^ψ.

Example 9 (muddy children continued). Let $\mathcal{M}$ be the model of Figure 2. We have:

$$\mathcal{M}, w \models \langle K_b p_a!\rangle K_a p_a \wedge \langle\bullet!\rangle K_a p_a \wedge \langle\bullet!_{\{b\}}\rangle K_a p_a.$$

3.4 Symbolic presentations of models

As in [19], [20], a *symbolic accessibility relation*, simply called an *accessibility program*, or even a *program*, describes a relation between valuations by executing an explicit sequence of propositional variable assignments. We write $\mathtt{u} \xrightarrow{\pi} \mathtt{v}$ for "$\mathtt{v}$ is a π-successor of $\mathtt{u}$ by π". The syntax for symbolic programs is the following.

$$\pi \ ::= \ p{\leftarrow}\beta \mid \beta? \mid (\pi;\pi) \mid (\pi \cup \pi) \mid (\pi \cap \pi) \mid \pi^{-1}$$

where $p \in \mathtt{AP}$, β is a Boolean formula over $\mathtt{AP}$.
The intuitive meaning of the constructions for programs is given in Table 1.

$p{\leftarrow}\beta$	Set p to the value of Boolean formula β
$\beta?$	Test that β holds.
$\pi;\pi'$	Execute π then execute π'.
$\pi \cup \pi'$	Non-deterministically execute π or π'.
$\pi \cap \pi'$	Execute the intersection of π and π'
π^{-1}	Converse of π

Table 1.

In what follows, we let $set(p_1, \ldots, p_n)$ denote the program $(p_1{\leftarrow}\bot \cup p_1{\leftarrow}\top); \ldots; (p_n{\leftarrow}\bot \cup p_n{\leftarrow}\top)$ that sets arbitrary values to $p_1, \ldots, p_n$.

Example 10 (Programs for the muddy children example). Since child a sees the forehead of child b but not her own, the program of a amounts to varying the truth value of p_a. That is, $\pi_a = set(p_a)$, and symmetrically for b, $\pi_b = set(p_b)$.

The semantics of programs is defined by induction.

- $\mathtt{w} \xrightarrow{p\leftarrow\beta} \mathtt{u}$ iff ($\mathtt{w} \not\models \beta$ and $\mathtt{u} = \mathtt{w}\backslash\{p\}$) or ($\mathtt{w} \models \beta$ and $\mathtt{u} = \mathtt{w} \cup \{p\}$);

– $\mathtt{w} \xrightarrow{\beta?} \mathtt{u}$ iff $\mathtt{w} \models \beta$ and $\mathtt{w} = \mathtt{u}$;
– $\mathtt{w} \xrightarrow{\pi_1;\pi_2} \mathtt{u}$ iff there exists $\mathtt{v}$ s.t. $\mathtt{w} \xrightarrow{\pi_1} \mathtt{v}$ and $\mathtt{v} \xrightarrow{\pi_2} \mathtt{u}$;
– $\mathtt{w} \xrightarrow{\pi_1 \cup \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ or $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$;
– $\mathtt{w} \xrightarrow{\pi_1 \cap \pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ and $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$;
– $\mathtt{w} \xrightarrow{\pi^{-1}} \mathtt{u}$ iff $\mathtt{u} \xrightarrow{\pi} \mathtt{w}$.

The size of a program is the number of nodes its syntax tree, or equivalently the number of symbols needed to write it, parenthesis omitted. For instance, the program $(p \leftarrow \top) \cup (q?; p \leftarrow \bot)$ has size 10.

As we have seen, the models are symbolically described by means of programs. They yield *symbolic Kripke models* that denote classic Kripke models[11]. However, the former may be exponentially more succinct than the latter

Definition 3 (Symbolic Kripke models). *A* symbolic Kripke model *is a tuple* $\mathfrak{M} = \langle \mathit{AP}_M, (\pi_a)_{a \in \mathit{Agt}} \rangle$ *where* $\mathit{AP}_M \subseteq \mathit{AP}$ *is a finite set of atomic propositions and* π_a *is a program over* AP_M *for each agent* a.

Intuitively, each program π_a symbolically describes the accessibility relation for an agent a.

Example 11. The symbolic Kripke model corresponding to the initial situation of the muddy children puzzle is $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ where:

– $\mathtt{AP}_M = \{p_a \mid a \in \mathtt{Agt}\}$;
– $\pi_a = \mathit{set}(p_a)$ for all agents a.

A pointed symbolic Kripke model is a pair $(\mathfrak{M}, \mathtt{w})$ where $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ is a symbolic Kripke model and $\mathtt{w}$ is a valuation over $\mathtt{AP}_M$.

We define the explicit Kripke model $\hat{M}(\mathfrak{M})$ associated to the symbolic Kripke model $\mathfrak{M}$: the set of worlds is the set of valuations over $\mathtt{AP}_M$ and the accessibility relation $\xrightarrow{a}$ is the relation $\xrightarrow{\pi_a}$.

Definition 4. *Given a symbolic Kripke model* $\mathfrak{M} = \langle \mathit{AP}_M, (\pi_a)_{a \in \mathit{Agt}} \rangle$, *the* Kripke model represented by $\mathfrak{M}$, *noted* $\hat{M}(\mathfrak{M})$ *is the model* $(W, (\xrightarrow{a})_{a \in \mathit{Agt}}, V)$ *where:*

– $W = \mathcal{V}(\mathit{AP}_M)$ *where* $\mathcal{V}(\mathit{AP}_M)$ *is the set of valuations over* AP_M;
– $\xrightarrow{a} = \{(\mathtt{w}, \mathtt{u}) \in W^2 \mid \mathtt{w} \xrightarrow{\pi_a} \mathtt{u}\}$;
– $V(\mathtt{w}) = \mathtt{w}$.

We write $\mathfrak{M}, w \models \varphi$ instead of $\hat{M}(\mathfrak{M}), w \models \varphi$.

Example 12 (muddy children continued). The Kripke model corresponding to M is $\hat{M}(\mathfrak{M}) = (W, \{\xrightarrow{a}\}_{a \in \mathtt{Agt}}, V)$ where $W = \mathcal{V}(\mathtt{AP}_M)$; for every $a \in \mathtt{Agt}$, $\xrightarrow{a} = \{(\mathtt{w}, \mathtt{u}) \in W^2 \mid \mathtt{w} \setminus p_a = \mathtt{u} \setminus p_a\}$; $V(\mathtt{w}) = \mathtt{w}$. Compared to the Kripke model given in Example 7 whose size is exponential in $|\mathtt{Agt}|$, the symbolic Kripke model is of size $3|\mathtt{Agt}|$.

Example 13 (Russian cards). First we consider the following symbolic Kripke model $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a \in \mathtt{Agt}} \rangle$ where: $\mathtt{AP}_M = \{p_{i,a}, p_{i,b}, p_{i,c} \mid i \in \{1, \ldots, 7\}\}$; $\pi_x = \mathit{set}\{p_{i,y} \mid i \in \{1, \ldots, 7\}$ and $y \in \{a, b, c\} \setminus \{x\}\}$ for agent $x \in \{a, b, c\}$. The Kripke model corresponding to the initial situation of the Russian card is $\hat{M}(\mathfrak{M})^{\varphi_R}$, which corresponds to model $\hat{M}(\mathfrak{M})$ after the *fake* announcement φ_R that enforces common knowledge that agents a and b have 3 cards each and c has 1.

We finally define the symbolic model checking problem against AGPAL which is central in our contribution, and that we write AGPAL-**mc**.

– Input: a symbolic model $\mathfrak{M}$, a valuation w, and a formula φ;
– Output: yes if $\mathfrak{M}, w \models \varphi$, no otherwise.

[11] Actually, and *vice versa* [20].

4 Announcement logic into monadic monadic second-order logic

We reduce the model checking against AGPAL to the satisfiability problem of MMSO. Intuitively, second-order variables denote current sets of valuations, called *contexts*, and first-order variables denote possible worlds/valuations. We present the reduction in four steps:

1. we define an MMSO-theory that enforce the MMSO-model to contain all valuations (Theorem 1);
2. we translate arbitrary accessibility programs into first-order logic (Theorem 2);
3. we translate AGPAL formulas into MMSO (Theorem 3);
4. we give the reduction of the AGPAL-model checking into the MMSO-satisfiability problem (Theorem 4).

4.1 The theory of models of valuations

In this section, we fix a set of atomic propositions A. Since we evaluate AGPAL-formulas on a symbolic model $\mathfrak{M}$ meant to denote the Kripke model with all valuations, we therefore need to enforce that all such valuations are captured.

Definition 5. *The model of valuations $\mathscr{M}_A$ on A is the structure $\mathscr{M}_A = (D, (P^{\mathscr{M}_A})_{p\in A})$ with D is the domain of all valuations on A and the interpretation of P is defined by as $P^{\mathscr{M}_A}(\mathtt{w})$ iff $p \in \mathtt{w}$.*

In what follows, we write P_A for the set of atomic predicates associated to some $p \in A$.

Definition 6. *Let β be a Boolean formula over A. We define the first-order formula $tr(\beta)(\mathsf{x})$ to be formula β in which each occurrence of $p \in \mathtt{AP}$ is replaced by $P(\mathsf{x})$. Similarly, for a valuation $\mathtt{w}$, we define $tr(\mathtt{w})(\mathsf{x})$ for the formula describing $\mathtt{w}$ where all p are replaced by $P(\mathsf{x})$.*

Example 14. Let $\beta = (p \vee q) \wedge (\neg p \vee q)$. Then $tr(\beta)(\mathsf{x}) = (P(\mathsf{x}) \vee Q(\mathsf{x})) \wedge (\neg P(\mathsf{x}) \vee Q(\mathsf{x}))$.

Example 15. Let $\mathtt{w} = \{p, q\}$ a valuation over $A = \{p, q, r\}$. $tr(\mathtt{w})(\mathsf{x}) = P(\mathsf{x}) \wedge Q(\mathsf{x}) \wedge \neg R(\mathsf{x})$.

We define a theory $\mathscr{T}_A$ such that $\mathscr{M}_A$ satisfies $\mathscr{T}_A$ and every model satisfying $\mathscr{T}_A$ is isomorphic to $\mathscr{M}_A$.

Currently, in an arbitrary structure $(D, (P_i^{\mathscr{M}})_{p_i\in\mathtt{AP}})$, two distinct elements e, e' in D may be such that $e \in P_i^{\mathscr{M}}$ iff $e' \in P_i^{\mathscr{M}}$ for all $p_i \in \mathtt{AP}$. To prevent it, we define $\varphi_{unique} = \forall\mathsf{x}\forall\mathsf{y}(\mathsf{x} = \mathsf{y}) \leftrightarrow \bigwedge_{p\in A}(P(\mathsf{x}) \leftrightarrow P(\mathsf{y}))$. It says that two elements satisfy the same predicates (i.e. are the same valuation) iff they are equal. We define too φ_{exists} says that for each valuation, for each atomic proposition p, there exists another valuation that differs only on p. In other words, $\varphi_{exists} = \forall\mathsf{x} \bigwedge_{p\in A} \Big(\exists\mathsf{y} \Big((P(\mathsf{x}) \leftrightarrow \neg P(\mathsf{y})) \wedge \bigwedge_{q\in A, q\neq p}(Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y}))\Big)\Big)$, imposing all valuations to appear in the model.

By letting $\mathscr{T}_A = \{\varphi_{unique}, \varphi_{exists}\}$, we get the following.

Theorem 1. *For all* MMSO*-models $\mathscr{M}$, we have $\mathscr{M} \models \mathscr{T}_A$ iff $\mathscr{M}$ is isomorphic to $\mathscr{M}_A$.*

Proof. $\Leftarrow$: It is sufficient to prove that $\mathscr{M}_A \models \mathscr{T}_A$:

- $\mathscr{M}_A \models \varphi_{unique}$ because each valuation is represented exactly one time in D and by Definition 5, P mimics the role of the atomic propositions in the valuations.
- $\mathscr{M}_A \models \varphi_{exists}$ because all valuations are represented in D.

Therefore $\mathscr{M}_A \models \mathscr{T}_A$ and thus $\mathscr{M} \models \mathscr{T}_A$.

$\Rightarrow$: Let $\mathscr{M}$ be such that $\mathscr{M} \models \mathscr{T}_A$. Let D' be the domain of $\mathscr{M}$ and P' be the monadic predicates of $\mathscr{M}$. We define the mapping $f : D' \to D$ such that for all $e \in D$, $f(e)$ is the valuation $\{p \mid e \in P'\} \in D$. We conclude by showing that f is an isomorphism.

- f is injective: if $f(e) = f(e')$, it means that for all P, $e \in P^{\mathscr{M}}$ iff $e' \in P^{\mathscr{M}}$. With $\mathscr{M} \models \varphi_{unique}$, we conclude that $e = e'$.
- f is surjective: let $\mathtt{w}$ be an element of D. As D' is non-empty, let e be in D'. As $\mathscr{M} \models \varphi_{exists}$, we can, from e, guarantee the existence of an element e' of D' such that $f(e') = \mathtt{w}$.

From Theorem 1, we obtain the following.

Corollary 1. *Let φ be an* MMSO*-formula. Then $\mathscr{M}_A \models \varphi$ if, and only if, $\mathscr{T}_A \wedge \varphi$ is* MMSO*-satisfiable.*

4.2 From programs to FO-formulas

Definition 7. *Let π be a program and x,y be two first-order variables. We define the first-order formula $\pi(\mathsf{x},\mathsf{y})$ by induction π as follows:*

$$\begin{array}{ll}(p \leftarrow \beta)(\mathsf{x},\mathsf{y}) & = (P(\mathsf{y}) \leftrightarrow tr(\beta)(\mathsf{x})) \wedge \bigwedge_{q\in A, q\neq p} (Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y}));\\ \beta?(\mathsf{x},\mathsf{y}) & = tr(\beta)(\mathsf{x}) \wedge (\mathsf{x}=\mathsf{y});\\ (\pi_1;\pi_2)(\mathsf{x},\mathsf{y}) & = \exists \mathsf{z}\ \pi_1(\mathsf{x},\mathsf{z}) \wedge \pi_2(\mathsf{z},\mathsf{y}).\\ (\pi_1 \cup \pi_2)(\mathsf{x},\mathsf{y}) & = \pi_1(\mathsf{x},\mathsf{y}) \vee \pi_2(\mathsf{x},\mathsf{y});\\ (\pi_1 \cap \pi_2)(\mathsf{x},\mathsf{y}) & = \pi_1(\mathsf{x},\mathsf{y}) \wedge \pi_2(\mathsf{x},\mathsf{y});\\ \pi^{-1}(\mathsf{x},\mathsf{y}) & = \pi(\mathsf{y},\mathsf{x}).\end{array}$$

The formula $\pi(\mathsf{x},\mathsf{y})$ expresses that y is a π-successor of x. It should be noticed that formulas $\pi(\mathsf{x},\mathsf{y})$ are in MFO, although the notation might be misleading. Formally:

Theorem 2. *For all worlds $\mathtt{w},\mathtt{u}$ and π, $\mathtt{w} \xrightarrow{\pi} \mathtt{u}$ if, and only if, $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi(\mathsf{x},\mathsf{y})$.*

Proof. By induction on π.

- $\pi = p \leftarrow \beta$:
  $\mathtt{w} \xrightarrow{p\leftarrow\beta} \mathtt{u}$ iff ($p \in \mathtt{u}$ iff $\mathtt{w} \models \beta$) and for all $q \neq p$, ($q \in \mathtt{w}$ iff $q \in \mathtt{u}$).
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models P(\mathsf{y}) \leftrightarrow tr(\beta)(\mathsf{x})$ and for all $q \neq p, \mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models Q(\mathsf{x}) \leftrightarrow Q(\mathsf{y})$.
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models (p \leftarrow \beta)(\mathsf{x},\mathsf{y})$.
- $\pi = \beta?$:
  $\mathtt{w} \xrightarrow{\beta?} \mathtt{u}$ iff $\mathtt{w} = \mathtt{u}$ and $\mathtt{w} \models \beta$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models (\mathsf{x}=\mathsf{y})$ and $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models tr(\beta)(\mathsf{x})$.
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \beta?(\mathsf{x},\mathsf{y})$.
- $\pi = \pi_1;\pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1;\pi_2} \mathtt{u}$ iff there exists $\mathtt{v}$ such that $\mathtt{w} \xrightarrow{\pi_1} \mathtt{v}$ and $\mathtt{v} \xrightarrow{\pi_2} \mathtt{u}$
  iff there exists $\mathtt{v}$ such that $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}, \mathsf{z}\leftarrow \mathtt{v}] \models \pi_1(\mathsf{x},\mathsf{z}) \wedge \pi_2(\mathsf{z},\mathsf{y})$.
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models (\pi_1;\pi_2)(\mathsf{x},\mathsf{y})$.
- $\pi = \pi_1 \cup \pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1\cup\pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ or $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi_1(\mathsf{x},\mathsf{y})$ or $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi_2(\mathsf{x},\mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models (\pi_1 \cup \pi_2)(\mathsf{x},\mathsf{y})$.
- $\pi = \pi_1 \cap \pi_2$:
  $\mathtt{w} \xrightarrow{\pi_1\cap\pi_2} \mathtt{u}$ iff $\mathtt{w} \xrightarrow{\pi_1} \mathtt{u}$ and $\mathtt{w} \xrightarrow{\pi_2} \mathtt{u}$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi_1(\mathsf{x},\mathsf{y})$ and $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi_2(\mathsf{x},\mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models (\pi_1 \cap \pi_2)(\mathsf{x},\mathsf{y})$.
- $\pi = \pi'^{-1}$:
  $\mathtt{w} \xrightarrow{\pi'^{-1}} \mathtt{u}$ iff $\mathtt{u} \xrightarrow{\pi'} \mathtt{w}$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi(\mathsf{y},\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x}\leftarrow \mathtt{w}, \mathsf{y}\leftarrow \mathtt{u}] \models \pi'^{-1}(\mathsf{x},\mathsf{y})$.

4.3 From AGPAL-formulas to MMSO-formulas

In the following definition, we define $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ to be the translation of the AGPAL-formula φ, where x is a first-order variable representing the valuation in which the formula φ is evaluated and X is a second-order variable representing the context (namely, the set of valuations that survived the previous announcements). Both variables x and X are the sole free variables of $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$.

Definition 8. *Let $\mathfrak{M} = \langle \mathtt{AP}_M, (\pi_a)_{a\in\mathtt{Agt}}\rangle$ be a symbolic model, φ be a* AGPAL*-formula, X be a second-order variable, and x be a first-order variable. We define the* MMSO*-formula $tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ by induction over φ, with the notation $\mathsf{Y} \subseteq \mathsf{X}$ for $\forall \mathsf{x}(\mathsf{Y}(\mathsf{x}) \rightarrow \mathsf{X}(\mathsf{x}))$.*

$tr_{\mathsf{X}}(p)(\mathsf{x}) = P(\mathsf{x})$;
$tr_{\mathsf{X}}(\neg\varphi)(\mathsf{x}) = \neg tr_{\mathsf{X}}(\varphi)(\mathsf{x})$;
$tr_{\mathsf{X}}(\varphi_1 \vee \varphi_2)(\mathsf{x}) = tr_{\mathsf{X}}(\varphi_1)(\mathsf{x}) \vee tr_{\mathsf{X}}(\varphi_2)(\mathsf{x})$;
$tr_{\mathsf{X}}(K_a\varphi)(\mathsf{x}) = \forall \mathsf{y}\ [(\mathsf{X}(\mathsf{y}) \wedge \pi_a(\mathsf{x},\mathsf{y})) \rightarrow tr_{\mathsf{X}}(\varphi)(\mathsf{y})]$;
$tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x}) = \exists \mathsf{Y}\ (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x})$;
$tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x}) = \exists \mathsf{Y}\ \mathsf{Y} \subseteq \mathsf{X} \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x})$;
$tr_{\mathsf{X}}((\langle\bullet!_G\rangle\varphi))(\mathsf{x}) = \exists \mathsf{Y}\ \mathsf{Y} \subseteq \mathsf{X} \wedge \mathit{isGroupAnnouncement}_G(\mathsf{Y}) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x})$.
where $\mathit{isGroupAnnouncement}_G(\mathsf{Y}) = \bigwedge_{a\in G} \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x},\mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z},\mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(\mathsf{x})$.

Formula $tr_{\mathsf{X}}(K_a\varphi)(\mathsf{x})$ mimics the standard translation of modal logic into first-order logic ([11], p. 84), except that we use the MFO-formula $\pi_a(\mathsf{x},\mathsf{y})$ instead of $R_a(\mathsf{x},\mathsf{y})$. In formula $tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x})$, we ask for the existence of a context Y that corresponds to the set of valuations in which ψ holds ($\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))$), that contains x ($\mathsf{Y}(\mathsf{x})$) and where φ holds. Formula $tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x})$ is similar to formula $tr_{\mathsf{X}}(\langle\varphi!\rangle\psi)(\mathsf{x})$, except that, as the announcement is arbitrary, we only impose that the context Y is included in X. Formula $tr_{\mathsf{X}}((\langle\bullet!_G\rangle\varphi))(\mathsf{x})$ is similar to $tr_{\mathsf{X}}(\langle\bullet!\rangle\varphi)(\mathsf{x})$ but we impose that the announcement is a group announcement. This constraint is guaranteed by formula $\mathtt{isGroupAnnouncement}_G(\mathsf{Y})$ that is a characterization of submodels generated by a group announcement.

We now state and prove the correctness of the translation.

Theorem 3. *Let $\mathfrak{M}$ be a symbolic model on A, φ be an* AGPAL*-formula on A and $\mathtt{w} \in \mathfrak{M}$. Let $D_{\mathfrak{M}}$ be the set of valuations of $\mathfrak{M}$. Then $\mathfrak{M}, \mathtt{w} \models \varphi$ iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi)(\mathsf{x})$.*

Proof. By induction on φ.

- $\varphi = p$:
  $\mathfrak{M}, \mathtt{w} \models \varphi$ iff $p \in \mathtt{w}$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models P(\mathsf{x})$
- $\varphi = \neg\psi$:
  $\mathfrak{M}, \mathtt{w} \models \neg\psi$ iff $\mathfrak{M}, \mathtt{w} \not\models \psi$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \not\models tr_{\mathsf{X}}(\varphi)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \neg tr_{\mathsf{X}}(\varphi)(\mathsf{x})$
- $\varphi = \varphi_1 \vee \varphi_2$:
  $\mathfrak{M}, \mathtt{w} \models \varphi_1 \vee \varphi_2$ iff $\mathfrak{M}, \mathtt{w} \models \varphi_1$ or $\mathfrak{M}, \mathtt{w} \models \varphi_2$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_1)(\mathsf{x})$ or $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_2)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\varphi_1)(\mathsf{x}) \vee tr_{\mathsf{X}}(\varphi_2)(\mathsf{x})$
- $\varphi = (K_a\varphi)$:
  $\mathfrak{M}, \mathtt{w} \models (K_a\psi)$ iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, $\mathfrak{M}, \mathtt{u} \models \psi$
  iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y})$
  iff for all $\mathtt{u} \in D_{\mathfrak{M}}$ such that $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \pi_a(\mathsf{x},\mathsf{y})$,
  $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \forall \mathsf{y}\ (\mathsf{X}(\mathsf{y}) \wedge \pi_a(\mathsf{x},\mathsf{y}) \rightarrow tr_{\mathsf{X}}(\varphi)(\mathsf{y}))$
- $\varphi = (\langle\chi!\rangle\psi)$:
  $\mathfrak{M}, \mathtt{w} \models (\langle\chi!\rangle\psi)$ iff $\mathfrak{M}, \mathtt{w} \models \chi$ and $\mathfrak{M}^\chi, \mathtt{w} \models \psi$
  iff $\mathfrak{M}, \mathtt{w} \models \chi$; for all $\mathtt{u}$, ($\mathtt{u} \in D_{\mathfrak{M}^\chi}$ iff $\mathtt{u} \in D_{\mathfrak{M}}$ and $\mathfrak{M}, \mathtt{u} \models \chi$); and $\mathfrak{M}^\chi, \mathtt{w} \models \psi$
  iff $\mathtt{w} \in D_{\mathfrak{M}^\chi}$; and for all $\mathtt{u}$, ($\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^\chi}] \models \mathsf{Y}(\mathsf{y})$ iff
  $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \mathsf{X}(\mathsf{y})$ and $\mathscr{M}_A[\mathsf{y} \leftarrow \mathtt{u}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}(\chi)(\mathsf{y}))$;
  and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^\chi}] \models tr_{\mathsf{X}}(\psi)(\mathsf{y}))$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D_{\mathfrak{M}^\chi}] \models \mathsf{Y}(\mathsf{x}) \wedge (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge tr_{\mathsf{Y}}(\chi)(\mathsf{x})$
  iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \exists \mathsf{Y}\ \mathsf{Y}(\mathsf{x}) \wedge (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \leftrightarrow (\mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\psi)(\mathsf{y}))) \wedge tr_{\mathsf{Y}}(\chi)(\mathsf{x})$
- $\varphi = (\langle\bullet!\rangle\psi)$:

$\mathfrak{M}, \mathtt{w} \models (\langle \bullet ! \rangle \psi)$ iff there exists a formula χ such that $\mathfrak{M}, \mathtt{w} \models \langle \chi ! \rangle \psi$.
iff there exists $D' \subseteq D_{\mathfrak{M}}$ such that $\mathtt{w} \in D'$ and $\mathfrak{M}', \mathtt{w} \models \psi$
(where $\mathfrak{M}'$ is $\mathfrak{M}$ restricted to D'.)[12]
iff there exists D' such that $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models \forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})$ and
$\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D'] \models \mathsf{Y}(\mathsf{x})$ and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow D'] \models tr_{\mathsf{Y}}(\psi)(\mathsf{x}))$
iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models \exists \mathsf{Y}\ (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\varphi)(\mathsf{x})$

– $\varphi = (\langle \bullet !_G \rangle \psi)$: to prove this case, we first prove the following lemma.

Lemma 1. *Let $\mathfrak{M}$ be a Kripke model on $\mathtt{AP}$, ψ be a formula on $\mathtt{AP}_M$, a an agent. Then for all contexts $D' \subseteq D_{\mathfrak{M}}$, there exists χ such that $D' = D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ iff*

$$\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$$

Proof. $\Rightarrow$ If there exists χ such that $D' = D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ then $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y}))$. For the other formula, let $\mathtt{w}$ be a world such that for all $\mathtt{u}$ with $\mathtt{w} \xrightarrow{\pi_a} \mathtt{u}$, there exists a world $\mathtt{v}$ with $\mathtt{v} \xrightarrow{\pi_a} \mathtt{u}$. Then by definition, $\mathfrak{M}, \mathtt{u} \models \chi$ and so $\mathfrak{M}, \mathtt{w} \models K_{\pi_a}\chi$. We conclude that $\mathtt{w} \in D'$, so $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$.

$\Leftarrow$ If $\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D'] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y}) \rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$ then $D' \subseteq D_{\mathfrak{M}}$. Let χ be the formula characterizing $post_{\pi_a}(D') = \{\mathtt{u} \in D_{\mathfrak{M}}$, there exists $\mathtt{v} \in D'$ such that $\mathtt{v} \xrightarrow{\pi_a} \mathtt{u}\}$ (the successors of D' via π_a). Then we obtain $D' \subseteq D_{\mathfrak{M}^{K_{\pi_a}\chi}}$. For the other implication, we observe that any element of $D' \subseteq D_{\mathfrak{M}^{K_{\pi_a}\chi}}$ has all its π_a-successors in $post_{\pi_a}(D')$, so is in D'.

Now back to the proof of the $\varphi = (\langle \bullet !_G \rangle \psi)$ case. Thanks to Lemma 1, we obtain:
$\mathfrak{M}, \mathtt{w} \models (\langle \bullet ! \rangle \psi)$ iff there exists formulas $\{\chi_g, g \in G\}$ such that $\mathfrak{M}, \mathtt{w} \models \langle \bigwedge_{g \in G} K_{\pi_g}\chi_g ! \rangle \psi$.
iff there exists $\{D_g, g \in G\}$ such that for all $g \in G$
$\mathscr{M}_A[\mathsf{X} \leftarrow D_{\mathfrak{M}}, \mathsf{Y} \leftarrow D_g] \models (\forall \mathsf{y}\ \mathsf{Y}(\mathsf{y}) \rightarrow \mathsf{X}(\mathsf{y})) \wedge \forall \mathsf{x}\ (\forall \mathsf{y}\ \pi_a(\mathsf{x}, \mathsf{y})$
$\rightarrow (\exists \mathsf{z}\ \pi_a(\mathsf{z}, \mathsf{y}) \wedge \mathsf{Y}(\mathsf{z}))) \rightarrow \mathsf{Y}(x))$
and $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{Y} \leftarrow \bigcap_{g \in G} D_g] \models \mathsf{Y}(\mathsf{x}) \wedge tr_{\mathsf{Y}}(\psi)(\mathsf{x})$.
iff $\mathscr{M}_A[\mathsf{x} \leftarrow \mathtt{w}, \mathsf{X} \leftarrow D_{\mathfrak{M}}] \models tr_{\mathsf{X}}((\langle \bullet !_G \rangle \psi))(\mathsf{x})$.

4.4 Reduction from AGPAL-mc to MMSO-sat

We wrap up our results obtained so far to define the reduction from the symbolic model checking problem against AGPAL to the MMSO-satisfiability problem.

Definition 9 (reduction). *Given a pointed symbolic Kripke model $(\mathfrak{M}, w)$ and an* AGPAL*-formula φ, we let $\tau(\mathfrak{M}, w, \varphi)$ be the* MMSO *formula $\mathscr{T}_A \wedge tr(\mathtt{w})(\mathsf{x}) \wedge \forall \mathsf{y} \mathsf{X}(\mathsf{y}) \wedge tr_{\mathsf{X}}(\varphi)(\mathsf{x})$ that is computable in polynomial time in the size of $\mathfrak{M}$.*

By Corollary 1 and Theorem 3 we get the following.

Theorem 4. *$\mathfrak{M}, w, \models \varphi$ iff $\tau(\mathfrak{M}, w, \varphi)$ is* MMSO*-satisfiable.*

Because the symbolic model checking of AGPAL is $\mathrm{A}_{\mathrm{pol}}$EXPTIME-hard [19], we obtain:

Corollary 2. MMSO*-satisfiability problem is* A_{pol}EXPTIME*-hard.*

However, as discussed in the next section, restricting to logic ∃AGPAL yields a reduction to the satisfiability problem of monadic first-order logic MFO.

[12] The right-to-left implication is proven by considering $\chi = \bigvee_{\mathtt{w} \in D'} \bigwedge_{p \in A, p \in \mathtt{w}} p \wedge \bigwedge_{q \in A, q \notin \mathtt{w}} \neg q$.

5 Existential announcement logic into monadic first-order logic

If we restrict inputs $\mathfrak{M}, w, \varphi$ of the AGPAL-model checking by letting $\varphi \in \exists$AGPAL, then $\tau(\mathfrak{M}, w, \varphi)$ is an MMSO-formula where all second-order quantifiers are existential and are not under the scope of universal quantifiers. Such second-order quantifiers can be removed from the formula $\tau(\mathfrak{M}, w, \varphi)$ resulting in a MFO-formula.

Since the symbolic model checking against $\exists$AGPAL is NExptime-hard [19], the icing on the cake is the following already well-known lower-bound.

Corollary 3. MFO-*satisfiability problem is* NExptime-*hard.*

In the next section, we make use of this reduction to solve the symbolic model checking problem against $\exists$AGPAL.

6 Implementation

We implemented the reduction from $\exists$AGPAL to MFO in OCaml. We also built benchmarks. The code and a readme file can be found at the following link

`https://github.com/tcharrie/agpal-mmso`

6.1 Description of the implementation

The input is an $\exists$AGPAL formula of the type `agpal_formula` in the source code. The type `acc_program` represents accessibility programs, the type `bool_formula` boolean formulas, and the type `fo_formula` MFO-formulas (the output of the code). The function `agpal_formula_to_mfo` defines the translation from $\exists$AGPAL formulas to MFO formulas (as in Definition 9).

In addition to the algorithm for the reduction, we implemented a function from existential formulas to the TPTP format [1] used by the FO-SAT-solvers, called `agpal_formula_to_tptp`. It first calls the function `agpal_formula_to_mfo`, then calls the function `mfo_formula_to_tptp` that transforms a MFO-formula into its TPTP representation.

6.2 Benchmarks

We provide benchmarks for FO-provers built from the muddy children and the Russian card puzzles in order to tests the combinatorial ability of FO-provers.

Muddy children. We consider the following true properties:

- $\varphi^{muddy}_{standard} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bigwedge_{a\in \mathtt{Agt}} \neg(K_a p_a \wedge \neg K_a \neg p_a)! \rangle \dots \langle \bigwedge_{a\in \mathtt{Agt}} \neg(K_a p_a \wedge \neg K_a \neg p_a)! \rangle \bigvee_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: standard formalization of the muddy children.
- $\varphi^{muddy}_{arbitrary} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bullet! \rangle \bigwedge_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: variant with an arbitrary announcement.
- $\varphi^{muddy}_{group} = \langle \bigvee_{a\in \mathtt{Agt}} p_a! \rangle \langle \bullet!_{\mathtt{Agt}} \rangle \bigwedge_{a\in \mathtt{Agt}} (K_a p_a \vee K_a \neg p_a)$: variant with a group announcement.

where $\mathtt{Agt} = \{1, \dots, n\}$.

Russian cards. For this example, agents a and b holds the same number of cards n. For instance, the classical Russian cards problem corresponds to $n = 3$. Let $\varphi^{Russian}_{goal} = \bigwedge_{i=1}^{2n+1} (K_a p_{i,b} \vee K_a \neg p_{i,b}) \wedge (K_b p_{i,a} \vee K_b \neg p_{i,a}) \wedge \neg K_c p_{i,a} \wedge \neg K_c \neg p_{i,a} \wedge \neg K_c p_{i,b} \wedge \neg K_c \neg p_{i,b}$. We consider three types of properties:

- $\varphi^{Russian}_{arbitrary} = \langle \varphi_R! \rangle \langle \bullet! \rangle \varphi^{Russian}_{goal}$: formalization of the Russian cards with a unique arbitrary announcement.
- $\varphi^{Russian}_{group_1} = \langle \varphi_R! \rangle \langle \bullet!_a \rangle \varphi^{Russian}_{goal}$: formalization with only one announcement from a. This formula is not satisfiable.
- $\varphi^{Russian}_{group_2} = \langle \varphi_R! \rangle \langle \bullet!_a \rangle \langle \bullet!_b \rangle \varphi^{Russian}_{goal}$: normal formalization of the Russian cards problem.

6.3 Experiments

To perform the tests, we used the FO-solver Iprover [30] on a HP EliteBook 840 G2. The prover Iprover enabled us to test whether a FO-formula is satisfiable or not. The results are summarized in Figure 6.3.

$n =$	$\varphi^{muddy}_{arbitrary}$	$n =$	$\varphi^{muddy}_{standard}$	φ^{muddy}_{group}	$n =$	$\varphi^{Russian}_{arbitrary}$	$\varphi^{Russian}_{group_1}$	$\varphi^{Russian}_{group_2}$
3	0.03s	3	0.07s	0.04s	2	0.18s	0.32s	0.45s
10	0.20s	4	0.09s	0.08s	3	0.44s	0.85s	0.92s
25	1.32s	5	0.19s	0.22s	4	3.80s	3.51s	3.32s
40	3.23s	6	0.24s	0.25s	5	23.48s	26.80s	24.20s
55	9.405s	7	$>$ 10min	$>$ 10min	6	$>$ 10min	$>$ 10min	$>$ 10min

Fig. 3. Results for the implementation of the reduction from ∃AGPAL to MFO, using the FO-SAT-solver Iprover.

We now briefly comment on the experiments.

Muddy children. For $\varphi^{muddy}_{arbitrary}$, the FO-SAT solver seems to perform well in all cases, as arbitrary announcements only require the new context to be included in the previous one. Hence, in this example, it is sufficient to restrict the model to the current world in order to satisfy the goal of $\varphi^{muddy}_{arbitrary}$. However, for the other tests, namely $\varphi^{muddy}_{standard}$ and φ^{muddy}_{group}, the FO-SAT-solver is able to test up to $n = 6$ agents. This can be explained by the fact public announcements and group announcements add significant combinatorial constraints to the specification.

Russian cards. For the three properties, the tests cannot exceed $n = 6$ cards, the main reason being that the rules of the game are very combinatorial, as for the muddy children.

Notice that the problems we have considered are puzzles, thus highly combinatorial. For the muddy children puzzle, the existential second-order quantification ranges over 2^{2^n} subsets. For $n = 7$, we have $2^{2^7} = 2^{128} \sim 10^{38}$, that is, about the number of positions $1.15868.. \times 10^{42}$ of a chess board.

Still, our implementation is promising and provides some interesting benchmarks for FO-provers.

7 Conclusion

We have reduced the problem of model checking symbolic Kripke models against AGPAL formulas to the satisfiability problem of MMSO, and shown that for the fragment ∃AGPAL, the reduction yields a satisfiability problem of some MFO formulas, which is known to be decidable [4,33]. We then have conducted experiments with FO provers. Our experiments show that the symbolic model checking problem against ∃AGPAL is difficult. As this problem is equivalent[13] to the MFO-satisfiability problem (they are both NEXPTIME-complete), we claim that efforts to obtain efficient algorithms are alike.

An interesting future work would be to effectively synthesize announcements. To this aim, we would like to generate the most simple formula to be announced so that a given property holds. This is close to the problem of generating a first-order model for a given MFO-formula.

We believe that our work is important since it would give efficient algorithms for several symbolic models in epistemic logic [7,26,27,18]. We also believe that the work done can improve epistemic planning specifications: in epistemic planning instances [13], the set of available actions is finite and described explicitly. Arbitrary announcement is a way to describe them implicitly. One can think of them as an action type while a specific announcement is an action token. Having efficient algorithms in this context would be very relevant.

[13] A reversed reduction can be proved.

Besides, we strongly believe that efficient data structures as in [37] for representing sets of sets of valuations are useful. Indeed, as Boolean formulas correspond to a set of valuations (and thus to binary decision diagrams [22]), an AGPAL-formula corresponds to a set of pair context/world, that, in a nutshell, could be represented by a set of sets of valuations.

On a more theoretical side, we would like to investigate on the relationship between announcement logics and MSO. Indeed, in MSO, second-order quantifications range over arbitrary sets (or over finite sets in weak-MSO) while announcements restrict the model to sets that are bisimulation-closed. We are not aware of any results regarding such second-order quantifiers.

Acknowledgments We thank Konstantin Korovin who helped us to use iprover. We thank Ocan Sankur for pin-pointing us the article [12] where the authors reduce the model checking of safety properties into FO.

References

1. The tptp (thousands of problems for theorem provers) library. `http://www.cs.miami.edu/~{}tptp/`.
2. Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *J. Applied Logic*, 8(1):62–81, 2010.
3. Thomas Ågotnes, Hans van Ditmarsch, and Tim French. The undecidability of group announcements. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pages 893–900, 2014.
4. Leo Bachmair, Harald Ganzinger, and Uwe Waldmann. Set constraints are the monadic class. In *Proceedings of the Eighth Annual Symposium on Logic in Computer Science (LICS '93), Montreal, Canada, June 19-23, 1993*, pages 75–83, 1993.
5. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
6. Philippe Balbiani, Alexandru Baltag, Hans P. van Ditmarsch, Andreas Herzig, T. Hoshi, and Tiago De Lima. What can we achieve by arbitrary announcements?: A dynamic take on fitch's knowability. In *TARK*, pages 42–51, 2007.
7. Philippe Balbiani, Olivier Gasquet, and François Schwarzentruber. Agents that look at one another. *Logic Journal of the IGPL*, 21(3):438–467, 2013.
8. Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: A well-behaved variant of pdl. In *LICS*, pages 143–152, 2013.
9. Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.
10. Clark Barrett, Christopher L. Conway, Morgan Deters, Liana Hadarean, Dejan Jovanović, Tim King, Andrew Reynolds, and Cesare Tinelli. Cvc4. In *Proceedings of the 23rd International Conference on Computer Aided Verification*, CAV'11, pages 171–177, Berlin, Heidelberg, 2011. Springer-Verlag.
11. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
12. Roderick Bloem, Robert Könighofer, and Martina Seidl. Sat-based synthesis methods for safety specs. In *Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, pages 1–20, 2014.
13. Thomas Bolander and Mikkel Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
14. Laura Bozzelli, Hans P. van Ditmarsch, and Sophie Pinchinat. The complexity of one-agent refinement modal logic. In *JELIA*, pages 120–133, 2012.
15. Laura Bozzelli, Hans P. van Ditmarsch, and Sophie Pinchinat. The complexity of one-agent refinement modal logic. In *IJCAI*, 2013.
16. Ashok K. Chandra and Larry J. Stockmeyer. Alternation. In *Proc. of FOCS'76*, pages 98–108, 1976.
17. Christophe Chareton and Hans van Ditmarsch. Strategic knowledge of the past in quantum cryptography.
18. Tristan Charrier, Andreas Herzig, Emiliano Lorini, Faustine Maffre, and François Schwarzentruber. Building epistemic logic from observations and public announcements. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016.*, pages 268–277, 2016.
19. Tristan Charrier and François Schwarzentruber. Arbitrary public announcement logic with mental programs. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 1471–1479, 2015.

20. Tristan Charrier and François Schwarzentruber. A succinct language for dynamic epistemic logic. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*, pages 123–131, 2017.
21. Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, pages 337–340, 2008.
22. Rolf Drechsler and Bernd Becker. *Binary Decision Diagrams - Theory and Implementation*. Springer, 1998.
23. Ronald Fagin, Yoram Moses, Joseph Y Halpern, and Moshe Y Vardi. *Reasoning about knowledge*. MIT press, 2003.
24. Michael J Fischer and Richard E Ladner. Propositional dynamic logic of regular programs. *Journal of computer and system sciences*, 18(2):194–211, 1979.
25. Tim French and Hans P. van Ditmarsch. Undecidability for arbitrary public announcement logic. In *Advances in Modal Logic*, pages 23–42, 2008.
26. Olivier Gasquet, Valentin Goranko, and François Schwarzentruber. Big brother logic: visual-epistemic reasoning in stationary multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 30(5):793–825, 2016.
27. Andreas Herzig, Emiliano Lorini, and Faustine Maffre. A poor man's epistemic logic based on propositional assignment and higher-order observation. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, pages 156–168, 2015.
28. Andreas Herzig and Faustine Maffre. How to share knowledge by gossiping. In *European Conference on Multi-Agent Systems*, pages 249–263. Springer, 2015.
29. David S. Johnson. A catalog of complexity classes. In *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity (A)*, pages 67–161. Elsevier, 1990.
30. Konstantin Korovin. iprover - an instantiation-based theorem prover for first-order logic (system description). In *Automated Reasoning, 4th International Joint Conference, IJCAR 2008, Sydney, Australia, August 12-15, 2008, Proceedings*, pages 292–298, 2008.
31. Séverin Lemaignan, Raquel Ros, L Mosenlechner, Rachid Alami, and Michael Beetz. Oro, a knowledge management platform for cognitive architectures in robotics. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3548–3553. IEEE, 2010.
32. Séverin Lemaignan, Mathieu Warnier, Akin E. Sisbot, and Rachid Alami. Human-Robot Interaction: Tackling the AI Challenges. *Artificial Intelligence*, 2014.
33. Harry R. Lewis. Complexity results for classes of quantificational formulas. *J. Comput. Syst. Sci.*, 21(3):317–353, 1980.
34. Benedikt Löwe et al. Logic and the simulation of interaction and reasoning: Introductory remarks. 2008.
35. L. Löwenheim. Über möglichkeiten im relativkalkül. *Mathematische Annalen*, 76:447–470, 1915.
36. Joseph S. Miller and Lawrence S. Moss. The undecidability of iterated modal relativization. *Studia Logica*, 79(3):373–407, 2005.
37. Alexandre Niveau and Bruno Zanuttini. Efficient representations for the modal logic S5. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 1223–1229, 2016.
38. Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
39. G. Sutcliffe. The CADE ATP System Competition - CASC. *AI Magazine*, 37(2):99–101, 2016.
40. Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic. In *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, pages 366–378, 2015.
41. Hans van Ditmarsch, Davide Grossi, Andreas Herzig, Wiebe van der Hoek, and Louwe B Kuijer. Parameters for epistemic gossip problems. *Proc. LOFT 2016*, 2016.
42. Hans van Ditmarsch and Barteld Kooi. *One Hundred Prisoners and a Light Bulb*. Springer, 2015.
43. Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer, Dordecht, 2008.
44. Hans P. van Ditmarsch. The russian cards problem. *Studia Logica*, 75(1):31–62, 2003.