Department of Computer Science,
University of Otago

Technical Report OUCS-2006-06

What becomes true after arbitrary announcements

Authors:

Philippe Balbiani, Andreas Herzig, and Tiago De Lima
Institut de Recherche en Informatique de Toulouse (IRIT), Toulouse, France

Hans van Ditmarsch
Department of Computer Science, University of Otago

Status: Under Submission

Department of Computer Science,
University of Otago, PO Box 56, Dunedin, Otago, New Zealand

http://www.cs.otago.ac.nz/research/techreports.html

What becomes true after arbitrary announcements

PHILIPPE BALBIANI, HANS VAN DITMARSCH[1] , ANDREAS HERZIG, AND TIAGO DE LIMA[2]

ABSTRACT. Public announcement logic is an extension of multi-agent epistemic logic with dynamic operators to model the informational consequences of announcements to the entire group of agents. We propose an extension of public announcement logic with a dynamic modal operator that expresses what is true after *arbitrary* announcements. Intuitively, $[!]\varphi$ expresses that φ is true after an arbitrary announcement ψ. We show completeness for a Hilbert-style axiomatization of this logic, and also provide a labelled tableau-calculus.

Keywords: public announcement logic; dynamic epistemics; tableau calculus.

1 Introduction

One motivation to formalize the dynamics of knowledge is to characterize how truth or knowledge conditions can be realized by new information. From that perspective, it seems unfortunate that in public announcement logic *PAL* [12, 4, 17] it may come to pass that a true formula becomes false because it is announced. The prime example is the new information expressed by the Moore-sentence 'atom p is true and you (agent a) do not know that', formalized by $p \wedge \neg K_a p$ [11, 7], but there are many other examples [16]. After the Moore-sentence is announced, you know that p is true, so $p \wedge \neg K_a p$ is now false. Worse, no additional announcement or sequence of announcements can make it true again. Also, the Moore-sentence cannot become known. But, for example, true facts p can always become known. The issues of what can become true and known are also known as reachability and knowability, respectively, and the 'Fitch-paradox' addresses the problematic question whether what is true can become known. For example, see van Benthem in [15] or, for further references, [1].

Consider an extension of public announcement logic wherein we can express what becomes true, whether known or not, without explicit reference

[1]Hans van Ditmarsch thanks the University of Otago for the support of his research leave in Toulouse and much appreciates the hospitality of Irit.

[2]Supported by the Programme Alßan, the European Union Programme of High Level Scholarships for Latin America, scholarship number E04D041703BR.

to announcements realizing that. Let us work our way upwards from a concrete announcement. When p is true, it becomes known by announcing it. Formally, in public announcement logic, $p \wedge [!p]K_a p$. This is equivalent to

$$\langle !p \rangle K_a p$$

which stands for 'the announcement of p can be made and after that the agent knows p'. More abstractly this means that there is a announcement ψ, namely $\psi = p$, that makes the agent know p, slightly more formal:

$$\exists \psi : \langle !\psi \rangle K_a p$$

We introduce a dynamic modal operator that expresses *exactly* that:

$$\langle ! \rangle K_a p$$

Obviously, the truth of this expression depends on the model: p has to be true. In case p is false, we can achieve $\langle ! \rangle K_a \neg p$ instead. The formula $\langle ! \rangle (K_a p \vee K_a \neg p)$ is valid.

We overlooked a 'detail' of the semantics. The condition '$\langle ! \rangle \varphi$ is true iff there is a ψ such that $\langle !\psi \rangle \varphi$ is true' (for some state of the world in a Kripke model) is not well-defined, because the announced formula ψ may then case be the formula $\langle ! \rangle \varphi$ itself. We therefore need a syntactic restriction on announcements replacing a $\langle ! \rangle$ operator in a formula. We propose that such announcements may not contain $\langle ! \rangle$ operators. With that restriction, the language is well-defined. The corresponding logic is called arbitrary public announcement logic, or in short, *arbitrary announcement logic*. Some other options for the truth condition for [!], resulting in different semantics for [!], are discussed in the concluding section.

We provide a sound and complete axiomatization for arbitrary announcement logic – though unfortunately with an infinitary derivation rule, and also a sound and complete tableau calculus – although, for similar reasons, without a decision procedure to determine derivability. As the calculus uses a novel technique to label sequents with both the current state and the history of announcements leading to that state, we first introduce it for *PAL*, before generalizing it to arbitrary public announcement logic. That restriction is, of course, decidable. The technique of using the history of announcements in the tableau calculus is not unlike the history that is recorded in Lutz' recent optimal reduction of public announcement logic to epistemic logic [9].

In section 2 we define the logical language $\mathcal{L}_{apal}$ and its semantics. In section 3 we provide a Hilbert-style axiomatization of arbitrary announcement logic – in the proof, only crucial details are given in which it differs from a standard axiomatization of public announcement logic without common knowledge [17]. In section 4 we provide a tableau calculus for public announcement logic, and in section 5 we expand this to a tableau calculus for arbitrary public announcement logic. Section 6 outlines alternative semantics for the arbitrary announcement operator [!].

2 Syntax and semantics

Assume a finite set of agents A and a countably infinite set of atoms P.

DEFINITION 1 (Language). The language $\mathcal{L}_{apal}$ of arbitrary public announcement logic is inductively defined as

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid [!\varphi]\varphi \mid [!]\varphi$$

where $a \in A$ and $p \in P$. Additionally, $\mathcal{L}_{pal}$ is the language without inductive construct $[!]\varphi$, $\mathcal{L}_{el}$ the language without as well $[!\varphi]\varphi$, and $\mathcal{L}_{pl}$ the language without as well $K_a\varphi$.

The languages $\mathcal{L}_{pal}$, $\mathcal{L}_{el}$, and $\mathcal{L}_{pl}$ are, of course, those of public announcement logic, epistemic logic, and propositional logic, respectively.

For $K_a\varphi$, read 'agent a knows that φ'. For $[!\varphi]\psi$, read 'after public announcement of φ, ψ is true'. For $[!]\psi$, read 'after every public announcement, ψ is true'. Other propositional and epistemic connectives are defined by usual abbreviations. The dual of K_a is $\widehat{K}_a$, the dual of $[!\varphi]$ is $\langle !\varphi\rangle$, and the dual of $[!]$ is $\langle !\rangle$. For $\widehat{K}_a\varphi$, read 'agent a considers it possible that φ', for $\langle !\varphi\rangle\psi$, read 'the announcement φ can be made and after that ψ is true' and for $\langle !\rangle\psi$, read 'there is an announcement after which ψ.'

DEFINITION 2 (Structures). An *epistemic model* $M = \langle W, \sim, V\rangle$ consists of a *domain* W of (factual) *states* (or 'worlds'), *accessibility* $\sim : A \rightarrow \mathcal{P}(W \times W)$, where each $\sim(a)$ is an equivalence relation, and a *valuation* $V : P \rightarrow \mathcal{P}(W)$. For $w \in W$, (M, w) is an *epistemic state* (also known as a pointed Kripke model).

For $\sim(a)$ we write $\sim_a$, and for $V(p)$ we write V_p. So, accessibility $\sim$ can be seen as a set of equivalence relations $\sim_a$, and V as a set of valuations V_p. Given two states w, w' in the domain, $w \sim_a w'$ means that w is indistinguishable from w' for agent a on the basis of its knowledge.

We adopt the standard rules for omission of parentheses in formulas and we also delete them in representations of structures such as (M, w) whenever convenient and unambiguous. We continue with the semantics.

DEFINITION 3. Assume an epistemic model $M = \langle W, \sim, V\rangle$. The interpretation of an arbitrary $\varphi \in \mathcal{L}_{apal}$ is defined by induction. Note the restriction to the language of *PAL* in the clause for $[!]\varphi$.

$M, w \models p$	iff	$w \in V_p$
$M, w \models \neg\varphi$	iff	$M, w \not\models \varphi$
$M, w \models \varphi \wedge \psi$	iff	$M, w \models \varphi$ and $M, w \models \psi$
$M, w \models K_a\varphi$	iff	for all $u \in W : w \sim_a u$ implies $M, u \models \varphi$
$M, w \models [!\varphi]\psi$	iff	$M, w \models \varphi$ implies $M\vert_\varphi, w \models \psi$
$M, w \models [!]\varphi$	iff	for all $\psi \in \mathcal{L}_{pal} : M, w \models [!\psi]\varphi$

In clause $[!\varphi]\psi$ for public announcement, epistemic model

$M|_\varphi = \langle W', \sim', V' \rangle$ is defined as

$$\begin{array}{lcl} W' & = & \{w' \in W \mid M, w' \models \varphi\} \\ \sim'_a & = & \sim_a \cap (W' \times W') \\ V'_p & = & V_p \cap W' \end{array}$$

Formula φ is valid in model M, notation $M \models \varphi$, iff for all $w \in W$: $M, w \models \varphi$. Formula φ is valid, notation $\models \varphi$, iff for all M (given the parameters A and P): $M \models \varphi$.

The dynamic modal operator $[!\varphi]$ is interpreted as an epistemic state transformer. Announcements are assumed to be truthful, and this is commonly known by all agents. Therefore, the model $M|_\varphi$ is the model M restricted to all the states where φ is true, including access between states. Similarly, the dynamic model operator $[!]$ is interpreted as an epistemic state transformer. Note that we have to restrict the announcements ψ in $[!\psi]\varphi$ to $\mathcal{L}_{pal}$. Without such a (or other, see the final section) restriction the semantics would not be well-defined, as $[!]\varphi$ can itself be announced as well. (The semantics is well-defined in the lexicographic order on $\mathcal{L}_{apal}$ such that a formula in $\mathcal{L}_{pal}$ is less complex than any formula in $\mathcal{L}_{apal}$ containing at least one occurrence of $[!]$.) For the semantics of the dual operators, we have that $M, w \models \langle ! \rangle \psi$ iff there is a $\varphi \in \mathcal{L}_{pal}$ such that $M, w \models \langle !\varphi \rangle \psi$. And we have that $M, w \models \langle !\varphi \rangle \psi$ iff $M, w \models \varphi$ and $M|_\varphi, w \models \psi$. Given a sequence $\vec{\psi} = (\psi_1, \dots, \psi_k)$ of announcements, we write $M|_{\vec{\psi}}$ for $M|_{\psi_1}|_{\dots}|_{\psi_k}$.

EXAMPLE 4. A valid formula of the logic is $\langle ! \rangle (K_a p \vee K_a \neg p)$. To prove this, let (M, w) be arbitrary. Either $M, w \models p$ or $M, w \models \neg p$. In the first case, $M, w \models \langle ! \rangle (K_a p \vee K_a \neg p)$ because $M, w \models \langle !p \rangle (K_a p \vee K_a \neg p)$ – the latter is true because $M, w \models p$ and $M|_p, w \models K_a p$; in the second case, we analogously derive $M, w \models \langle ! \rangle (K_a p \vee K_a \neg p)$ because $M, w \models \langle !\neg p \rangle (K_a p \vee K_a \neg p)$.[1]

We continue by listing some relevant validities.

PROPOSITION 5. *Let,* $\varphi, \psi \in \mathcal{L}_{apal}$ *be arbitrary. Then*

1. $\models [!](\varphi \wedge \psi) \leftrightarrow ([!]\varphi \wedge [!]\psi)$
2. $\models [!]\varphi \rightarrow \varphi$
3. $\models [!]\varphi \rightarrow [!][!]\varphi$
4. $\models \varphi$ *implies* $\models [!]\varphi$
5. $\models K_a[!]\varphi \rightarrow [!]K_a\varphi$ *(but* not *in the other direction)*

---

[1] This example also nicely illustrates the order in which arbitrary objects come to light. The meaning of $\models \langle ! \rangle \varphi$ is (i) '*for all* (M, w) *there is a* ψ *such that* $M, w \models \langle !\psi \rangle \varphi$'. This is really different from (ii) '*there is a* ψ *such that for all* (M, w), $M, w \models \langle !\psi \rangle \varphi$', which might on first sight be appealing to the reader, when extrapolating from the *incorrect* reading 'there is a ψ such that $\models \langle !\psi \rangle \varphi$' of $\models \langle ! \rangle \varphi$. But, for example, there is no formula ψ in the language such that $\langle !\psi \rangle (K_a p \vee K_a \neg p)$ is valid: in other words, (i) may be true, even when (ii) is false.

Proof. See the appendix. ■

The following proposition (for an arbitrary set of agents A) will be helpful to show that in the single-agent case every formula is equivalent to an epistemic $\mathcal{L}_{el}$-formula.

PROPOSITION 6. *Let $\varphi, \varphi_0, \ldots, \varphi_n \in \mathcal{L}_{pl}$, i.e., booleans, and $\psi \in \mathcal{L}_{apal}$.*

1. $\models [!]\varphi \leftrightarrow \varphi$
2. $\models [!]\widehat{K}_a\varphi \leftrightarrow \varphi$
3. $\models [!]K_a\varphi \leftrightarrow K_a\varphi$
4. $\models [!](\varphi \vee \psi) \leftrightarrow (\varphi \vee [!]\psi)$
5. $\models [!](\widehat{K}_a\varphi_0 \vee K_a\varphi_1 \vee \cdots \vee K_a\varphi_n) \leftrightarrow (\varphi_0 \vee K_a(\varphi_0 \vee \varphi_1) \vee \ldots \vee K_a(\varphi_0 \vee \varphi_n))$

Proof. Left to the reader. ■

Let $A = \{a\}$. A formula is in *normal form* when it is a conjunction of disjunctions of the form $\varphi \vee \widehat{K}_a\varphi_0 \vee K_a\varphi_1 \vee \ldots \vee K_a\varphi_n$. Every formula in single-agent epistemic logic (*K45* and therefore also in) *S5* is equivalent to a formula in normal form [10].

PROPOSITION 7. *If there is only one agent, every formula in arbitrary announcement logic is equivalent to a formula in epistemic logic.*

Proof. By induction on the number of occurrences of [!]. Put the epistemic formula in the scope of an innermost [!] in normal form. First, we distribute [!] over the conjunction (proposition 5.1). We now get formulas of the form $[!](\varphi \vee \widehat{K}_a\varphi_0 \vee K_a\varphi_1 \vee \cdots \vee K_a\varphi_n)$. These are reduced by application of proposition 6.4 and 6.5 to formulas of form $\varphi_0 \vee K_a(\varphi_0 \vee \varphi_1) \vee \cdots \vee K(\varphi_0 \vee \varphi_n)$. ■

3 Axiomatization and completeness

In this section we attack the problem of the complete axiomatization of $\mathcal{L}_{apal}$. Let $\sharp$ be a new symbol. Following the line of reasoning suggested by Goldblatt [5] we inductively define the *necessity forms* as follows ($\varphi \in \mathcal{L}_{apal}$):

- $\sharp$ is a necessity form,
- if $\boldsymbol{\psi}$ is a necessity form then $(\varphi \rightarrow \boldsymbol{\psi})$ is a necessity form,
- if $\boldsymbol{\psi}$ is a necessity form then $[!\varphi]\boldsymbol{\psi}$ is a necessity form,
- if $\boldsymbol{\psi}$ is a necessity form then $K_a\boldsymbol{\psi}$ is a necessity form.

all instantiations of propositional tautologies	
$K_a(\varphi \to \psi) \to (K_a\varphi \to K_a\psi)$	distribution of knowledge over implication
$K_a\varphi \to \varphi$	truth
$K_a\varphi \to K_aK_a\varphi$	positive introspection
$\neg K_a\varphi \to K_a\neg K_a\varphi$	negative introspection
$[!\varphi]p \leftrightarrow (\varphi \to p)$	atomic permanence
$[!\varphi]\neg\psi \leftrightarrow (\varphi \to \neg[!\varphi]\psi)$	announcement and negation
$[!\varphi](\psi \wedge \chi) \leftrightarrow ([!\varphi]\psi \wedge [!\varphi]\chi)$	announcement and conjunction
$[!\varphi]K_a\psi \leftrightarrow (\varphi \to K_a[!\varphi]\psi)$	announcement and knowledge
$[!\varphi][!\psi]\chi \leftrightarrow [!(\varphi \wedge [!\varphi]\psi)]\chi$	announcement composition
$[!]\varphi \to [!\psi]\varphi$	arbitrary and specific announcement
From φ and $\varphi \to \psi$, infer ψ	modus ponens
From φ, infer $K_a\varphi$	necessitation of knowledge
From φ, infer $[!\psi]\varphi$	necessitation of announcement
From φ, infer $[!]\varphi$	necessitation of arbitrary announcement
From $\boldsymbol{\varphi}([!\chi]\psi)$ for all $\chi \in \mathcal{L}_{pal}$, infer $\boldsymbol{\varphi}([!]\psi)$	deriving arbitrary announcement / $R([!])$

Table 1. The axiomatization **APAL**

Note that each necessity form has a unique occurrence of $\sharp$. We write $\boldsymbol{\varphi}(\psi)$ to denote a single occurrence of a formula ψ in a necessity form $\boldsymbol{\varphi}(\sharp)$. This notation is, for example, used in the derivation rule $R([!])$ of the axiomatization **APAL**, now to follow.

DEFINITION 8. The axiomatization **APAL** is given in Table 1. A formula is a *theorem* if it belongs to the least set of formulas containing all axioms and closed under the rules. If φ is a theorem, we write $\vdash \varphi$.

All axioms and rules are sound. In particular, the rule $R([!])$ is correct with respect to the semantics, i.e. if $\models \boldsymbol{\varphi}([!\chi]\psi)$ for all $\chi \in \mathcal{L}_{pal}$, then $\models \boldsymbol{\varphi}([!]\psi)$. In the above formulation, the rule $R([!])$ is infinitary. The question lies open whether $R([!])$ can be replaced by a finitary inference rule or by finitely many axioms.

EXAMPLE 9. For an example of derivation in **APAL**, using the axiom and rule for $[!]$, we show that the valid formula $[!]\varphi \to [!][!]\varphi$ is also a theorem. Note that in step 4 of the derivation we use that $[!]p \to [!q]\sharp$ is a necessity form and that in step 5 of the derivation we use that $[!]p \to \sharp$ is a necessity form.

1. $\vdash [!(q \wedge [!q]r)]p \leftrightarrow [!q][!r]p$	announcement composition
2. $\vdash [!]p \to [!(q \wedge [!q]r)]p$	arbitrary and specific announcement
3. $\vdash [!]p \to [!q][!r]p$	$1, 2,$ propositionally
4. $\vdash [!]p \to [!q][!]p$	$3, R([!])$
5. $\vdash [!]p \to [!][!]p$	$4, R([!])$

The main effect of rule $R([!])$ is that it makes the canonical model (consisting of all maximal consistent sets of formulas closed under the rule) standard for [!]. Let us see how. In the remainder of this section, most proof details are omitted.

A set x of formulas is called a theory if it satisfies the following conditions:

- x contains the set of all theorems of $\mathcal{L}_{apal}$;
- x is closed under modus ponens and $R([!])$.

Obviously the least theory is the set of all theorems and the largest theory is the set of all formulas. The latter theory is called the trivial theory. A theory x is said to be consistent if $\bot \notin x$. We shall say that a theory x is maximal if for all formulas φ, $\varphi \in x$ or $\neg\varphi \in x$. Let x be a set of formulas. For all formulas φ, let $x + \varphi = \{\psi\colon\ \varphi \rightarrow \psi \in x\}$. For all agents a, let $K_a x = \{\varphi\colon\ K_a\varphi \in x\}$. For all formulas φ, let $[!\varphi]x = \{\psi\colon\ [!\varphi]\psi \in x\}$.

LEMMA 10. *Let x be a theory, φ be a formula, and a be an agent. Then $x + \varphi$, $K_a x$ and $[!\varphi]x$ are theories. Moreover $x + \varphi$ is consistent iff $\neg\varphi \notin x$.*

Proof. The proof is based on the fact that x is closed under modus ponens and $R([!])$. ■

LEMMA 11. *Let x be a consistent theory. There exists a maximal consistent theory y such that $x \subseteq y$.*

Proof. This is the Lindenbaum Lemma for the arbitrary announcement logic. The proof can be done as in [5]. ■

DEFINITION 12 (Canonical model). The canonical model of $\mathcal{L}_{apal}$ is the structure $\mathcal{M}_c = \langle W, \sim, V\rangle$ defined as follows:

- W is the set of all maximal consistent theories;
- For all agents a, $\sim_a$ is the binary (equivalence) relation on W defined by $x \sim_a y$ iff $K_a x = K_a y$;
- For all atoms p, V_p is the subset of W defined by $x \in V_p$ iff $p \in x$.

LEMMA 13 (Truth lemma). *Let φ be a formula in $\mathcal{L}_{apal}$. Then for all maximal consistent theories x and for all finite sequences $\vec{\psi} = (\psi_1, \ldots, \psi_k)$ of formulas in $\mathcal{L}_{apal}$ such that $\psi_1 \in x$, $[!\psi_1]\psi_2 \in x$, …, $[!\psi_1]\ldots[!\psi_{k-1}]\psi_k \in x$:*

$$\mathcal{M}_c|_{\vec{\psi}}, x \models \varphi \text{ iff } [!\psi_1]\ldots[!\psi_k]\varphi \in x$$

Proof. The proof is by induction on φ. ■

As a result we have:

THEOREM 14 (Soundness and completeness). *Let φ be a formula. Then φ is a theorem iff φ is valid.*

Proof. Soundness is immediate, following the observations at the beginning of this section. Completeness follows from Lemmas 10, 11, and 13. ■

4 A tableau method for public announcement logic

We present in this section an adequate proof method for public announcement logic that uses tableaux. Exactly in the same way as all other tableau methods, given a formula φ, it systematically tries to construct a model for it. When it fails, φ is inconsistent and thus its negation is valid.

We use the common tableau representation in which formulas are prefixed by a number that represents possible worlds in the model (similar to [3, Chapter 8]). In addition, formulas are also prefixed by sequences of announcements corresponding to successive model restrictions. Given a finite sequence of formulas $\vec{\psi} = (\psi_1, \ldots, \psi_k)$, for each $0 \leq i \leq k$, the sequence $(\psi_1, \ldots, \psi_i)$ is noted $\vec{\psi}^i$. The vector $\vec{\psi}^0 = \epsilon$ denotes the empty sequence.

DEFINITION 15. A *labelled formula* is a triple $\lambda = (\vec{\psi}^k : n : \varphi)$ where

- $\vec{\psi}^k$ is a finite sequence $(\psi_1, \ldots, \psi_k)$ of formulas in $\mathcal{L}_{pal}$;
- $n \in \mathbb{N}$; and
- $\varphi \in \mathcal{L}_{pal}$.

The part $\vec{\psi}^k : n$ is the *label* of the formula φ. It represents a possible world n in the epistemic model that is successively restricted by the sequence of formulas $\vec{\psi}^k$.

DEFINITION 16. A *skeleton* is a ternary relation $\Sigma \subseteq (A \times \mathbb{N} \times \mathbb{N})$ that represents the accessibility relations. A *branch* is a pair $b = \langle \Lambda, \Sigma \rangle$ where Λ is a set of labelled formulas and Σ is a skeleton.

DEFINITION 17 (Tableau). A *tableau* is a set $T^i = \{b_0^i, b_1^i, \ldots\}$ of branches. A tableau T^{i+1} is *obtained from* a tableau T^i if and only if $T^{i+1} := (T^i \setminus \{b_j^i\}) \cup B$ for some $b_j^i \in T^i$ and some finite set B of branches generated from b_j^i by one of the *tableau rules* defined below.

$\bot$: If $\{(\vec{\psi}^k : n : p), (\vec{\chi}^l : n : \neg p)\} \subseteq \Lambda$, then
$B = \{\langle \Lambda \cup \{(\epsilon : n : \bot)\}, \Sigma \rangle\}$.

$\neg$: If $(\vec{\psi}^k : n : \neg\neg\varphi) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi)\}, \Sigma \rangle\}$.

$\wedge$: If $(\vec{\psi}^k : n : \varphi_1 \wedge \varphi_2) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi_1), (\vec{\psi}^k : n : \varphi_2)\}, \Sigma \rangle\}$.

$\vee$: If $(\vec{\psi}^k : n : \neg(\varphi_1 \wedge \varphi_2)) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \neg\varphi_1)\}, \Sigma \rangle, \langle \Lambda \cup \{(\vec{\psi}^k : n : \neg\varphi_2)\}, \Sigma \rangle\}$.

K: If $(\vec{\psi}^k : n : K_a\varphi) \in \Lambda$ and $(a, n, n') \in \Sigma$, then

$B = \{\langle \Lambda_1, \Sigma\rangle, \dots \langle \Lambda_{k+1}, \Sigma\rangle\}$, where

$$\begin{array}{lll} \Lambda_1 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \neg\psi_1)\} \\ \Lambda_2 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \neg\psi_2)\} \\ \Lambda_3 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), (\vec{\psi}^2 : n' : \neg\psi_3)\} \\ & \vdots & \\ \Lambda_k & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-2} : n' : \psi_{k-1}), (\vec{\psi}^{k-1} : n' : \neg\psi_k)\} \\ \Lambda_{k+1} & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \varphi)\}. \end{array}$$

T: If $(\vec{\psi}^k : n : K_a\varphi) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi)\}, \Sigma\rangle\}$.

4: If $(\vec{\psi}^k : n : K_a\varphi) \in \Lambda$ and $(a, n, n') \in \Sigma$, then
$B = \{\langle \Lambda_1, \Sigma\rangle, \dots \langle \Lambda_{k+1}, \Sigma\rangle\}$, where

$$\begin{array}{lll} \Lambda_1 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \neg\psi_1)\} \\ \Lambda_2 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \neg\psi_2)\} \\ \Lambda_3 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), (\vec{\psi}^2 : n' : \neg\psi_3)\} \\ & \vdots & \\ \Lambda_k & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-2} : n' : \psi_{k-1}), (\vec{\psi}^{k-1} : n' : \neg\psi_k)\} \\ \Lambda_{k+1} & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : K_a\varphi)\}. \end{array}$$

$5_\uparrow$: If $(\vec{\psi}^k : n : K_a\varphi) \in \Lambda$ and $(a, n', n) \in \Sigma$, then
$B = \{\langle \Lambda_1, \Sigma\rangle, \dots \langle \Lambda_{k+1}, \Sigma\rangle\}$, where

$$\begin{array}{lll} \Lambda_1 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \neg\psi_1)\} \\ \Lambda_2 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \neg\psi_2)\} \\ \Lambda_3 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), (\vec{\psi}^2 : n' : \neg\psi_3)\} \\ & \vdots & \\ \Lambda_k & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-2} : n' : \psi_{k-1}), (\vec{\psi}^{k-1} : n' : \neg\psi_k)\} \\ \Lambda_{k+1} & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : K_a\varphi)\}. \end{array}$$

$\widehat{K}$: If $(\vec{\psi}^k : n : \neg K_a\varphi) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^0 : n' : \psi_1), \dots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \neg\varphi)\}, \Sigma \cup \{(a, n, n')\}\rangle\}$ for some n' that does not appear in Λ.

$[!\varphi]$: If $(\vec{\psi}^k : n : [!\varphi_1]\varphi_2) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \neg\varphi_1)\}, \Sigma\rangle, \langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi_1), (\vec{\psi}^k, \varphi_1 : n : \varphi_2)\}, \Sigma\rangle\}$.

$\langle !\varphi\rangle$: If $(\vec{\psi}^k : n : \neg[!\varphi_1]\varphi_2) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi_1), (\vec{\psi}^k, \varphi_1 : n : \neg\varphi_2)\}, \Sigma\rangle\}$.

1.	$\epsilon : 0 : \neg[!p \wedge \neg K_a p]\neg(p \wedge \neg K_a p)$		
2.	$\epsilon : 0 : p \wedge \neg K_a p$	(1)	
3.	$p \wedge \neg K_a p : 0 : \neg\neg(p \wedge \neg K_a p)$	(1)	
4.	$p \wedge \neg K_a p : 0 : p \wedge \neg K_a p$	(3)	
5.	$p \wedge \neg K_a p : 0 : p$	(4)	
6.	$p \wedge \neg K_a p : 0 : \neg K_a p$	(4)	
7.	$\epsilon : 1 : p \wedge \neg K_a p$	(6)	$(a, 0, 1) \in \Sigma$
8.	$p \wedge \neg K_a p : 1 : \neg p$	(6)	
9.	$\epsilon : 1 : p$	(7)	
10.	$\epsilon : 1 : \neg K_a p$	(7)	
11.	$\epsilon : 1 : \bot$	(8, 9)	

Figure 1. Closed tableau for the formula $[!(p \wedge \neg K_a p)]\neg(p \wedge \neg K_a p)$.

Given a formula $\varphi \in \mathcal{L}_{pal}$, the tableau $T^0 := \{b_0^0\} := \{\langle\{(\epsilon : 0 : \varphi)\}, \emptyset\rangle\}$ is the *initial tableau* for φ. A *tableau for* φ is a tableau that can be obtained from the initial tableau for φ by successive applications of tableau rules.

DEFINITION 18. The branch b is *closed* iff $(\epsilon : n : \bot) \in \Lambda$ for some n. The branch b is *open* iff it is not closed. A tableau is *closed* iff all its branches are closed. A tableau is *open* iff it has at least one open branch.

EXAMPLE 19. Consider the formula $[!(p \wedge \neg K_a p)]\neg(p \wedge \neg K_a p)$. Since it is valid in *PAL*, its negation is not satisfiable. Therefore, under the hypothesis that the method is complete, there exists a closed tableau for it, as shown in figure 1.

DEFINITION 20. The branch b is *satisfiable* if and only if there exists an epistemic structure $M = \langle W, \sim, V\rangle$ and a function f from $\mathbb{N}$ to W such that for all $(\vec{\psi}^k : n : \varphi) \in \Lambda$

$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n) \models \psi_1 & \text{and} \\ M|_{\vec{\psi}^1}, f(n) \models \psi_2 & \text{and} \\ \vdots & \\ M|_{\vec{\psi}^{k-1}}, f(n) \models \psi_k & \text{and} \\ M|_{\vec{\psi}^k}, f(n) \models \varphi & \end{array}$$

and for all $(a, n, n') \in \Sigma$, $f(n) \sim_a f(n')$. In this case we say that b is satisfiable via f.

A tableau is *satisfiable* if and only if it contains a satisfiable branch.

PROPOSITION 21. *φ is valid if and only if there exists a closed tableau for $\neg\varphi$.*

Proof. This is a particular case of corollaries 25 and 27 in section 5. ■

It should be possible to prove in a way similar to [3] and [6] that the number of different labels in a branch is at most exponential in the size of the input formula, establishing that our tableau method can be turned into a decision procedure for *PAL*.

5 A tableau method for arbitrary public announcement logic

Now, we generalise the method given in section 4 to arbitrary announcements. We reuse labelled formulas for $\mathcal{L}_{apal}$ as introduced in definition 15.

DEFINITION 22 (Tableau (continuation)). A *tableau* for the formula $\varphi \in \mathcal{L}_{apal}$ is defined as in Definition 17. The tableau rules are the same, plus the following ones.

[!]: If $(\vec{\psi}^k : n : [!]\varphi) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : [!\chi]\varphi)\}, \Sigma\rangle\}$ for any $\chi \in \mathcal{L}_{pal}$.

⟨!⟩: If $(\vec{\psi}^k : n : \neg[!]\varphi) \in \Lambda$, then
$B = \{\langle \Lambda \cup \{(\vec{\psi}^k : n : \neg[!p]\varphi)\}, \Sigma\rangle\}$ for some $p \in P$ that does not occur in Λ.

These rules are similar to Smullyan's tableau rules for closed first-order formulas [13, 8]. They reflect that the operator [!] quantifies over announcements. In tableau rule [!], this operator is eliminated by replacing it by an arbitrary $\mathcal{L}_{pal}$-formula. Tableau rule ⟨!⟩ is more curious though: instead of replacing the operator by an announcement $\langle !\psi\rangle$ of a $\mathcal{L}_{pal}$-formula ψ, we replace it by an announcement of a new propositional letter. The intuitive argument here is the following. Since this new propositional letter does not occur in the branch, we are free to give it an arbitrary interpretation to represent a specific restriction in the model. In this way, we make the calculus simpler because it is not necessary to make a 'good choice' at the moment of the application of rule ⟨!⟩. The example below may help to clarify these intuitions.

EXAMPLE 23. Consider the valid formula $[!]K_a p \rightarrow [!][!]K_a p$. A closed tableau for its negation is shown in figure 2.

PROPOSITION 24. *If φ is satisfiable, then there is no closed tableau for φ.*

Proof. See the appendix. ■

COROLLARY 25 (Soundness). *If there is a closed tableau for $\neg\varphi$, then φ is valid.*

PROPOSITION 26. *If there is no closed tableau for φ, then φ is satisfiable.*

Proof. See the appendix. ■

COROLLARY 27 (Completeness). *If φ is valid, then there is a closed tableau for $\neg\varphi$.*

Let T be a closed tableau. Hence there is no infinite branch in T. Seeing that T is finitely branching, König's infinity lemma for trees implies that T is finite. This leads immediately to the following corollary.

1.	$\epsilon : 0 : [!]K_a p \wedge \neg[!][!]K_a p$		
2.	$\epsilon : 0 : [!]K_a p$	(1)	
3.	$\epsilon : 0 : \neg[!][!]K_a p$	(1)	
4.	$\epsilon : 0 : \neg[!q][!]K_a p$	(3)	
5.	$\epsilon : 0 : q$	(4)	
6.	$q : 0 : \neg[!]K_a p$	(4)	
7.	$q : 0 : \neg[!r]K_a p$	(6)	
8.	$q : 0 : r$	(7)	
9.	$q, r : 0 : \neg K_a p$	(7)	
10.	$\epsilon : 1 : q$	(9)	$(a, 0, 1) \in \Sigma$
11.	$q : 1 : r$	(9)	
12.	$q, r : 1 : \neg p$	(9)	
13.	$\epsilon : 0 : [!q]K_a p$	(2)	

Left branch:

14.	$\epsilon : 0 : \neg q$	(13)
17.	$\epsilon : 0 : \bot$	(5, 14)

Right branch:

15.	$\epsilon : 0 : q$	(13)
16.	$q : 0 : K_a p$	(13)

Right branch, left sub-branch:

18.	$\epsilon : 1 : \neg q$	(16)
21.	$\epsilon : 1 : \bot$	(10, 18)

Right branch, right sub-branch:

19.	$\epsilon : 1 : q$	(16)
20.	$q : 1 : p$	(16)
22.	$\epsilon : 1 : \bot$	(12, 20)

Figure 2. Closed tableau for the formula $[!]K_a p \rightarrow [!][!]K_a p$.

COROLLARY 28. *The set of valid formulas of $\mathcal{L}_{apal}$ is recursively enumerable.*

Proof. To see this, it suffices to consider a Turing machine TM that enumerates all possible pairs (φ, T) made up of a formula φ and a closed tableau T. For each generated pair (φ, T), TM checks whether T is a tableau for $\neg\varphi$. When the checking process is finished, TM generates another pair, performs another round of checking, and so on ad infinitum. ■

6 Discussion and conclusion

We proposed an extension of public announcement logic with an operator [!] that expresses what is true after *arbitrary* announcements. We proved several validities involving that operator, gave a sound and complete infinitary axiomatization, and a labelled tableau calculus, presented both for public announcement logic and for arbitrary public announcement logic. Some further issues in the logic, and in the tableau calculi, might be interesting and fruitful to pursue.

Alternative semantics for arbitrary announcement. Currently, the semantics states that $[!]\varphi$ is true in some epistemic state iff $[!\psi]\varphi$ is true there for all $\psi \in \mathcal{L}_{pal}$, i.e., formulas of public announcement logic, without arbitrary announcement operators. As said, without some such restriction the semantics would not be well-defined. An alternative semantics would be to define that $[!]\varphi$ is true in some epistemic state iff $[!\psi]\varphi$ is true for all ψ lower than $[!]\varphi$ in a different suitable complexity order. What order? A likely condition seems that $[!\psi]\varphi < [!]\varphi$ if $\psi < [!]\varphi$. Such a semantics leads in a, we think, *very* promising direction: the truth of a formula $[!]\varphi$ then depends on the structure of φ (and the current epistemic state) *only*, in other words: there can be no surprising, complex, announcements, realizing φ. In fact, the most likely outcome is that the set of formulas less complex than φ is *finite*, in which case we would have a validity $[!]\varphi \leftrightarrow \bigwedge_{\psi \in cl(\varphi)} [!\psi]\varphi$, where $cl(\varphi)$ is the set of formulas simpler than φ – one may well think of this set as a Fischer-Ladner sort of closure. (The intuitively more appealing form of the validity is $\langle ! \rangle \varphi \leftrightarrow \bigvee_{\psi \in cl(\varphi)} \langle !\psi \rangle \varphi$.) The validity $[!]\varphi \rightarrow [!\psi]\varphi$ (for arbitrary formulas) then easily follows. We would then also have a decision procedure to determine validity, the logic would be greatly simplified, etc[2].

This may appear an unlikely outcome. But, to sharpen our intuitions, consider the formula $\langle ! \rangle K_a p$. It is easy to conceive a model wherein by announcing q the agent learns p, so that we have $\langle !q \rangle K_a p$. In which case there is no relation between the achieved $K_a p$ and the announcement realizing it. On the other hand, in the initial model p must have been true, so announcing p would have achieved the same. Therefore $\langle !p \rangle K_a p$. Now there is a relation between the announcement and the postcondition!

World-based semantics. Another option for the truth condition for $[!]$ would be to quantify over subsets of the set of possible states, instead of quantifying over formulas. This straightforwardly avoids syntactic problems as the above. Formally, we would have that (given domain W of model M)

$$M, w \models [!]\varphi \text{ iff } M|_{W'}, w \models \varphi \text{ for all } W' \subseteq W \text{ such that } w \in W'$$

This corresponds to one of the semantics for propositional quantifiers in modal logic proposed by Fine [2]. In this alternative 'world-based semantics' arbitrary announcement no longer preserves bisimulation.

For a distinguishing example, consider the formula $\varphi = (K_b p \wedge K_b \neg K_a p) \rightarrow [!](K_a p \rightarrow K_b K_a p)$, and a four-state model $M = \langle W, \sim, V \rangle$ such that $W = \{s_1, s_2, t_1, t_2\}$, $\sim_a = \{\langle s_1, t_1 \rangle, \langle s_2, t_2 \rangle\}^*$, $\sim_b = \{\langle s_1, s_2 \rangle\}^*$ (where $*$ is reflexive, symmetrical and transitive closure), and $V(p) = \{s_1, s_2\}$. Then for the world-based semantics $M, s_1 \not\models \varphi$: $M, s_1 \models K_b p \wedge K_b \neg K_a p$, but $M, s_1 \not\models [!](K_a p \rightarrow K_b K_a p)$ because $M|_T, s_1 \not\models K_a p \rightarrow K_b K_a p$, where $T = \{s_1, s_2, t_2\}$. But: for the semantics presented in section 2, $M, s_1 \models \varphi$ and $M', s_1 \models \varphi$.

M is bisimilar to the two-state model $M' = \langle \{s_1, t_1\}, \sim', V' \rangle$ where $\sim'_1 = \{\langle s_1, s_1 \rangle\}$, $\sim'_2 = \{\langle s_1, t_1 \rangle\}^*$, and $V'(p) = \{s_1\}$, and $M', s_1 \not\models \varphi$. Whereas using the semantics of definition 3 φ is true in *both* (M, s_1) and (the bisimilar) (M', s_1). Clearly, also the logics would differ for the world-based semantics.

[2] An alternative outcome is that $cl(\varphi)$ as above is infinite, but that, instead, satisfiability can be determined by a finite closure set. This suggests a completeness proof for such a logic styled on similar completeness proofs for announcement logics with infinitary modal operators [17], such as common knowledge, and with a finite axiomatization different from the infinitary one presented in Section 3. Note that in a (finite) canonical model for $\langle ! \rangle \varphi$ the formula $\langle ! \rangle \varphi$ is equivalent to a boolean combination of formulas of form $\langle !\psi \rangle \varphi$.

A semantics that does not preserve bisimilarity under action execution is obviously undesirable. (It can be easily shown that arbitrary announcement preserves bisimilarity of structures with respect to the set of accessibility relations $\sim_a$ for all agents). A different realization of Fine's idea's would be only to require that all subsets in a bisimulation contraction are definable. This would follow if all singletons in the contraction are definable. The formula characterizing that may well be $\langle ! \rangle (\varphi \rightarrow [!]\varphi)$ (*iii*). This formula expresses that we can make an anouncement after which no announcement has any further informative effect. If no announcement is informative, this means that no actual model restriction can occur. This is only the case in a singleton model. Therefore, (*iii*) expresses that an announcement can be made to restrict arbitrary epistemic states to singletons, or, in other words, that every epistemic state has a characteristic formula. The principle (*iii*) is not valid for either of the two semantics we have discussed here.

Positive formulas. Given a preferred semantics, a suitable direction of research may be the syntactic or semantic characterization of interesting fragments of the logic. Consider the fragment $\varphi ::= p|\neg p|\varphi \vee \varphi|\varphi \wedge \varphi|K_a\varphi|[!\neg\varphi]\varphi|[!]\varphi$ of the *positive* formulas. The positive formulas preserve truth under arbitrary model restriction. Restricted to epistemic logic, this was observed by van Benthem in [14]. Van Ditmarsch and Kooi extended this in [16] to public announcement logic with clause $[!\neg\varphi]\varphi$ – note that the truth of the announcement is a condition of its execution, which, when seen as a disjunction, explains the negation in $[!\neg\varphi]$. Surprisingly, we can expand this fragment with $[!]\varphi$ for arbitrary announcement logic (in the case $[!]\varphi$ of the inductive proof to show truth preservation, assuming the opposite easily leads to a contradiction). It is also easy to see that for every positive formula φ, $\varphi \rightarrow [!]\varphi$ is valid: this expresses truth preservation after arbitrary announcement, i.e., arbitrary *definable* submodel restriction. This principle may possibly also characterize the positive formulas. Note that there is no corresponding principle in public announcement logic that expresses truth preservation, although there we can express the notion of *success* (announced formulas always remain true, as typically required in belief revision) by a principle $[!\varphi]\varphi$. Preserved formulas are always successful: $\varphi \rightarrow [!]\varphi$ implies $\varphi \rightarrow [!\varphi]\varphi$, and $\varphi \rightarrow [!\varphi]\varphi$ iff $[!\varphi]\varphi$.

Further extensions of the language. Along a common line in dynamic epistemics, one might expand the language with additional modal operators, in particular with common knowledge, with assignments (actions that change the truth value of atomic propositions), and with actions that are not public, such as private announcements. Thus, the notion of an arbitrary public announcement, which can be seen as an epistemic action, can be expanded by adding assignments to a more extensive action language for *planning*. For the most expressive form of such a language our original problem almost becomes trivial: in finite models all (satisfiable) formulas are realizable in such a logic, or, more formally, given arbitrary finite (M, w)

and satisfiable φ, there is an epistemic action α such that $(M, w \models \langle\alpha\rangle\varphi$. This follows from a recent unpublished result by van Ditmarsch and Kooi.

What is the modal logic of [!]? We have seen various schematic validities involving *only* the operator [!], such as $[!]\varphi \to \varphi$ and $[!]\varphi \to [!][!]\varphi$ (see proposition 5). Further note that we can 'ground' the notion of epistemic state transition, induced by the interpretation of [!], to a notion of *local* accessibility between states in a model, by the expedient of 'lifting' individual accessibility $w \sim_a u$ to a state transition $(M, w) \sim_a (M, u)$, and then seeing epistemic states as points in the larger model induced by such state transitions. We then can ask ourselves the following two questions: given the logical language $\varphi ::= p|\neg p|\varphi \wedge \varphi|[!]\varphi$: What is that class of models? What is the logic of [!] alone? This logic is at least *S4* because both $[!]\varphi \to \varphi$ and $[!]\varphi \to [!][!]\varphi$ are valid. In other respects, remark that $\langle!\rangle\varphi \wedge \langle!\rangle\psi \to ((\langle!\rangle(\varphi \wedge \langle!\rangle\psi) \vee (\langle!\rangle(\langle!\rangle\varphi \wedge \psi)$ is not valid. (To see this in $\mathcal{L}_{apal}$, replace φ by $K_a p \wedge \neg K_a q$, and ψ by $\neg K_a p \wedge K_a q$.)

We conjecture that the McKinsey formula $[!]\langle!\rangle\varphi \to \langle!\rangle[!]\varphi$ is valid. This schema expresses that a formula can change its truth value only *finitely* often as a result of a sequence of announcements. To see this correspondence, suppose the schema is *invalid*. Then there are (M, w) and (a concrete formula) φ such that $M, w \models [!]\langle!\rangle\varphi$ and $M, w \not\models \langle!\rangle[!]\varphi$, i.e., $M, w \models [!]\langle!\rangle\varphi$ (i) and $M, w \models [!]\langle!\rangle\neg\varphi$ (ii). If these are both true, the formula φ can change its truth value infinitely often. Note that a sequence of announcements is also an announcement, by the validity of $[!\varphi][!\psi]\chi \leftrightarrow [!(\varphi \wedge [!\varphi]\psi)]\chi$. Therefore, a *true* formula $[!\chi_1]\langle!\chi_2\rangle\varphi$ because of (i), gives us the requirement to *falsify* φ after announcement of $\chi_1 \wedge [!\chi_1]\chi_2$ because of (ii), and so on, arbitrarily often. It seems obvious that this cannot be, but we have no proof.

Along a similar line, we conjecture that the schemata $\varphi \to [!](\langle!\rangle\varphi \to \varphi)$ as well as the slightly weaker $\varphi \to ([!]\langle!\rangle\varphi \to [!]\varphi)$ are valid. These formulas express that a formula can change its truth value *only once* (instead of, for 'McKinsey', finitely often). To see this, note that the latter is equivalent to $(\varphi \wedge \langle!\rangle\neg\varphi) \to \langle!\rangle[!]\neg\varphi$, in other words: 'if φ is true and we can make it false, then we can guarantee that it remains false forever.' If the McKinsey formula $[!]\langle!\rangle\varphi \to \langle!\rangle[!]\varphi$ is valid, it should therefore follow that these two weaker principles are also valid.

Appendix

Proof of Proposition 5.

1. Obvious.
2. Assume $M, w \models [!]\varphi$. Then in particular, $M, w \models [\top]\varphi$, i.e. (as $M, w \models \top$), $M, w \models \varphi$.
3. Let M and $w \in M$ be arbitrary. Assume $M, w \models \langle!\rangle\langle!\rangle\neg\varphi$. Then there are χ and χ' such that $M, w \models \langle!\chi\rangle\langle!\chi'\rangle\neg\varphi$. Using the validity (for arbitrary formulas) $[!\varphi][!\varphi']\varphi'' \leftrightarrow [!(\varphi \wedge [!\varphi]\varphi')]\varphi''$, we therefore have $M, w \models \langle!(\chi \wedge [!\chi]\chi')\rangle\neg\varphi$, from which follows $M, w \models \langle!\rangle\neg\varphi$.

4. Let M, w be arbitrary. We have to show that for arbitrary $\psi \in \mathcal{L}_{pal}$: $M, w \models [!\psi]\varphi$. From the assumption $\models \varphi$ follows $\models [!\psi]\varphi$ by necessitation for $[!\psi]$. Therefore also $M, w \models [!\psi]\varphi$.

5. Let (M, w), $\varphi \in \mathcal{L}_{pal}$, and $u \in M|_\varphi$ with $u \sim_a w$ arbitrary. We have to prove that $M|_\psi, u \models \varphi$. Because state u is also in M, from the assumption $M, w \models K_a[!]\varphi$ and $w \sim_a u$ also in M follows $M, u \models [!]\varphi$. As ψ is true in u, $M|_\psi, u \models \varphi$.

■

Proof of Proposition 24. We show that satisfiability of formulae is preserved by the tableau rules defined above. In other words, let T^i be a tableau for a given formula that contains a branch $b = \langle \Lambda, \Sigma \rangle$. We show that if b is satisfiable in the sense of Definition 16, then the set of branches B generated by any tableau rule defined above has also at least one satisfiable branch.

- For rules $\bot$, $\neg$, $\wedge$ and $\vee$: Left to the reader.

- For rule K: If $\lambda = (\vec{\psi}^k : n : K_a\varphi) \in \Lambda$ and $(a, n, n') \in \Sigma$, then B contains all the branches $b_j = \langle \Lambda_j, \Sigma \rangle$ for $1 \leq j \leq k+1$ where
$$\begin{array}{lll} \Lambda_1 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \neg\psi_1)\} \\ \Lambda_2 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \neg\psi_2)\} \\ \Lambda_3 & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), (\vec{\psi}^2 : n' : \neg\psi_3)\} \\ & \vdots & \\ \Lambda_k & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \ldots, (\vec{\psi}^{k-1} : n' : \neg\psi_k)\} \\ \Lambda_{k+1} & = \Lambda \cup & \{(\vec{\psi}^0 : n' : \psi_1), \ldots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \varphi)\}. \end{array}$$
By hypothesis, there exists an epistemic structure $M = \langle W, \sim, V \rangle$ and a function f from $\mathbb{N}$ to W such that $M|_{\vec{\psi}^k}, f(n) \models K_a\varphi$. Then, for all $w \in W|_{\vec{\psi}^k}$, $w \in\sim_a (f(n))$ implies $M|_{\vec{\psi}^k}, w \models \varphi$. Then, one of the following conditions holds.
$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n') \models \neg\psi_1 & \text{or} \\ M|_{\vec{\psi}^1}, f(n') \models \neg\psi_2 & \text{or} \\ M|_{\vec{\psi}^2}, f(n') \models \neg\psi_3 & \text{or} \\ \vdots & \\ M|_{\vec{\psi}^{k-1}}, f(n') \models \neg\psi_k & \text{or} \\ M|_{\vec{\psi}^k}, f(n') \models \varphi. & \end{array}$$
Therefore one of the branches b_j is satisfiable.

- For rules T, 4 and $5_\uparrow$: The argument is similar to that for rule K. We also use the fact that $\sim$ is an equivalence relation.

- For rule $\widehat{K}$: If $(\vec{\psi}^k : n : \neg K_a\varphi) \in \Lambda$, then B contains only one branch $b_1 = \langle \Lambda_1, \Sigma_1 \rangle$ such that

$$\begin{aligned} \Lambda_1 &= \Lambda \cup \{(\vec{\psi}^0 : n' : \psi_1), \ldots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \neg\varphi)\} \\ \Sigma_1 &= \Sigma \cup \{(a, n, n')\} \end{aligned}$$

  for some n' that does not occur in Λ.
  By hypothesis, there exists an epistemic structure $M = \langle W, \sim, V \rangle$ and a function f from $\mathbb{N}$ to W such that

$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n) \models \psi_1 & \text{and} \\ \vdots & \\ M|_{\vec{\psi}^{k-1}}, f(n) \models \psi_k & \text{and} \\ M|_{\vec{\psi}^k}, f(n) \models \neg K_a\varphi & \end{array}$$

  Then, there exists $w \in W|_{\vec{\psi}^k}$ such that $w \in \sim_a (f(n))$ and $M|_{\vec{\psi}^k}, w \models \neg\varphi$. We thus consider the function f' from $\mathbb{N}$ to W defined as follows.

$$\begin{array}{lll} f'(n) & := f(n) & \text{for all } n \text{ that occurs in } \Lambda \\ f'(n') & := w & \end{array}$$

  Therefore, b_1 is satisfiable.

- For rule $[!\varphi]$: If $(\vec{\psi}^k : n : [!\varphi_1]\varphi_2) \in \Lambda$, then B contains the branches $b_1 = \langle \Lambda \cup \{(\vec{\psi}^k : n : \neg\varphi_1)\}, \Sigma \rangle$ and $b_2 = \langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi_1), (\vec{\psi}^k, \varphi_1 : n : \varphi_2)\}, \Sigma \rangle$. Seeing that $M|_{\vec{\psi}^k}, f(n) \models [!\varphi_1]\varphi_2$ iff $M|_{\vec{\psi}^k}, f(n) \models \neg\varphi_1$ or $M|_{\vec{\psi}^k}|_{\varphi_1}, f(n) \models \varphi_2$, thus b_1 is satisfiable or b_2 is satisfiable.

- For rule $\langle !\varphi \rangle$: If $(\vec{\psi}^k : n : \neg[!\varphi_1]\varphi_2) \in \Lambda$, then B contains only one branch $b_1 = \langle \Lambda \cup \{(\vec{\psi}^k : n : \varphi_1), (\vec{\psi}^k, \varphi_1 : n : \neg\varphi_2)\}, \Sigma \rangle$. Seeing that $M|_{\vec{\psi}^k}, f(n) \models \neg[!\varphi_1]\varphi_2$ iff $M|_{\vec{\psi}^k}, f(n) \models \varphi_1$ and $M|_{\vec{\psi}^k}|_{\varphi_1}, f(n) \models \neg\varphi_2$, thus b_1 is satisfiable.

- For rule $[!]$: If $(\vec{\psi}^k : n : [!]\varphi) \in \Lambda$, then B contains only one branch $b_1 = \langle \Lambda \cup \{(\vec{\psi}^k : n : [!\chi]\varphi)\}, \Sigma \rangle$ for some $\chi \in \mathcal{L}_{pal}$. Seeing that $M|_{\vec{\psi}^k}, f(n) \models [!]\varphi$ iff $M|_{\vec{\psi}^k}, f(n) \models [!\chi]\varphi$ for all $\chi \in \mathcal{L}_{pal}$, thus b_1 is satisfiable.

- For rule $\langle ! \rangle$: If $(\vec{\psi}^k : n : \neg[!]\varphi) \in \Lambda$, then B contains only one brance $b_1 = \langle \Lambda \cup \{(\vec{\psi}^k : n : \neg[!p]\varphi)\}, \Sigma \rangle$ for some $p \in P$ that does not occur in Λ. Since $M|_{\vec{\psi}^k}, f(n) \models \neg[!]\varphi$, then there exists a formula $\chi \in \mathcal{L}_{pal}$ such that $M|_{\vec{\psi}^k}, f(n) \models \neg[!\chi]\varphi$. Let $M' = \langle W, \sim, V' \rangle$ be the epistemic structure defined as follows.

  - $V'(p') := V(p')$ if p' is different fromp p; and
  - $V'(p) := \{w \in W : M|_{\vec{\psi}^k}, w \models \chi\}$.

  Now we have that $M|_{\vec{\psi}^k}, f(n) \models \neg[!p]\varphi$. Therefore, b_1 is satisfiable.

■

Proof of Proposition 26. We prove completeness of the tableau method proceeding by induction on the structure of formulae of $\mathcal{L}_{apal}$. For that reason, we need some definitions.

DEFINITION 29. The tableau T^i is *saturated* if and only if all applicable tableau rules have been applied at least once. More precisely, we say that T^i is saturated if and only if T^i is saturated under the tableau rules.

This implies, for example, that if $(\Lambda, \Sigma) \in T^i$ and $(\vec{\psi} : n : K_a\varphi) \in \Lambda$ then $(\vec{\psi} : n : \varphi) \in \Lambda$.

DEFINITION 30. Let $\lambda = (\vec{\psi}^k : n : \varphi)$ be a labelled formula. The *weight* of λ, $\|\lambda\|$, is the number of occurrences of the symbol $[!]$ in $\vec{\psi}^k$ and φ. The *length* of λ, $|\lambda|$, is inductively defined as follows.

- $|p| := 0$;
- $|\neg\varphi| := |\varphi| + 1$;
- $|\varphi_1 \wedge \varphi_2| := |\varphi_1| + |\varphi_2| + 1$;
- $|K_a\varphi| := |\varphi| + 1$;
- $|[!\varphi_1]\varphi_2| := |\varphi_1| + |\varphi_2| + 2$;
- $|[!]\varphi| := |\varphi| + 2$;
- $|(\vec{\psi}^k : n : \varphi)| := |\psi_1| + \ldots + |\psi_k| + k + |\varphi|$.

And the *size* of λ is the ordered pair $size(\lambda) := (\|\lambda\|, |\lambda|)$.

Let $size(\lambda_1) = (x_1, y_1)$ and $size(\lambda_2) = (x_2, y_2)$. Then size of λ_1 is *less than* size of λ_2, in symbols $size(\lambda_1) < size(\lambda_2)$, if and only if

- $x_1 < x_2$; or
- $x_1 = x_2$ and $y_1 < y_2$.

It is clear that the relation '$<$' is a well-founded order.

We prove that if a saturated tableau for a given formula φ is open, then φ is satisfiable. We prove this by using the open saturated tableau for the formula φ to construct an epistemic structure that satisfies it.

Suppose that T^ω is an open saturated tableau for φ. Then, it contains at least one open branch $b = \langle \Lambda, \Sigma \rangle$. We use this branch to construct an epistemic structure $M = \langle W, \sim, V \rangle$ as follows.

- $W := \{n \in \mathbb{N} : n \text{ occurs in } \Lambda\}$;
- $\sim_a :=$ symmetric, reflexive and transitive closure of $\{(n, n') : (a, n, n') \in \Sigma\}$;
- $V(p) := \{n : (\vec{\psi}^k : n : p) \in \Lambda \text{ for some } \vec{\psi}^k\}$; and

We also define $f(n) := n$ for all n occurring in Λ.

Clearly, W is a non-empty set, $\sim_a$ is an equivalence relation, $V(p)$ assigns a subset of W to each proposition that appears on the tableau. If $(a, n, n') \in \Sigma$, then $f(n') \in \sim_a (f(n))$. Thus, we now show, by induction on the size of λ, that for all labelled formulae $\lambda = (\vec{\psi}^k : n : \varphi) \in \Lambda$, we have $\mathcal{P}(\lambda)$ that is to say

$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n) \models \psi_1 & \text{and} \\ \vdots & \\ M|_{\vec{\psi}^{k-1}}, f(n) \models \psi_k & \text{and} \\ M|_{\vec{\psi}^k}, f(n) \models \varphi & \end{array}$$

- The base case, where φ is an atom and cases where $\varphi = \neg\neg\varphi_1$, $\varphi = \varphi_1 \wedge \varphi_2$ and $\varphi = \neg(\varphi_1 \wedge \varphi_2)$ are left to the reader.
- For $\varphi = K_a\varphi_1$: Then, by the fact that rules K, 4, $5_\uparrow$ and T have been applied as many times as possible, for all n' such that $(n, n') \in \sim_a$, Λ contains at least one of the following sets of labelled formulae.

$$\begin{array}{l} \{(\vec{\psi}^0 : n' : \neg\psi_1)\}; \\ \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \neg\psi_2)\}; \\ \{(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), (\vec{\psi}^2 : n' : \neg\psi_3)\}; \\ \vdots \\ \{(\vec{\psi}^0, n' : \psi_1), \ldots, (\vec{\psi}^{k-1} : n' : \neg\psi_k)\}; \\ \{(\vec{\psi}^0, n' : \psi_1), \ldots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \varphi_1)\}. \end{array}$$

  The size of each of these labelled formulae is less than the size of λ. By Induction Hypothesis, for all n' such that $(n, n') \in \sim_a$,

$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n') \models \neg\psi_1 & \text{or} \\ \vdots & \\ M|_{\vec{\psi^{k-1}}}, f(n') \models \neg\psi_k & \text{or} \\ M|_{\vec{\psi}^k}, f(n') \models \varphi_1 & \end{array}$$

  Then $M|_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

- For $\varphi = \neg K_a\varphi_1$: Then, by the fact that rule $\widehat{K}_a$ has been applied, Σ contains (a, n, n') and Λ contains

$$(\vec{\psi}^0 : n' : \psi_1), (\vec{\psi}^1 : n' : \psi_2), \ldots, (\vec{\psi}^{k-1} : n' : \psi_k), (\vec{\psi}^k : n' : \neg\varphi_1).$$

  The size of each of these labelled formulae is less than the size of λ. By Induction Hypothesis, for some $n' \in \sim_a (n)$

$$\begin{array}{ll} M|_{\vec{\psi}^0}, f(n') \models \psi_1 & \text{and} \\ \vdots & \\ M|_{\vec{\psi}^{k-1}}, f(n') \models \psi_k & \text{and} \\ M|_{\vec{\psi}^k}, f(n') \models \neg\varphi_1 & \end{array}$$

Then $M_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

- For $\varphi = [!\varphi_1]\varphi_2$: Then, by the fact that rule $[!\varphi]$ has been applied, Λ contains

  $(\vec{\psi}^k : n : \neg\varphi_1)$
  or both
  $(\vec{\psi}^k : n : \varphi_1)$ and $(\vec{\psi}^k, \varphi_1 : n : \varphi_2)$.

  The size of each of these labelled formulae is less than the size of λ. By Induction Hypothesis, $M|_{\vec{\psi}^k}, f(n) \models \neg\varphi_1$, or both $M|_{\vec{\psi}^k}, f(n) \models \varphi_1$ and $M|_{\vec{\psi}^k}|_{\varphi_1}, f(n) \models \varphi_2$. Then $M|_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

- For $\varphi = \neg[!\varphi_1]\varphi_2$: Then, by the fact that rule $\langle !\varphi \rangle$ has been applied, Λ contains

  $$(\vec{\psi}^k : n : \varphi_1) \text{ and } (\vec{\psi}^k, \varphi_1 : n : \neg\varphi_2).$$

  The size of each of these labelled formulae is less than the size of λ. By Induction Hypothesis, $M|_{\vec{\psi}^k}, f(n) \models \varphi_1$ and $M|_{\vec{\psi}^k}|_{\varphi_1}, f(n) \models \neg\varphi_2$. Therefore, $M|_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

- For $\varphi = [!]\varphi_1$: Then, by the fact that rule $[!]$ has been applied as many times as possible, Λ contains

  $$(\vec{\psi}^k : n : [!\chi]\varphi_1) \text{ for all } \chi \in \mathcal{L}_{pal}.$$

  As $\chi \in \mathcal{L}_{pal}$, the size of each of these labelled formulae is $(x-1, z)$ for some $z \in \mathbb{N}$. Then, the size of each of them is less than the size of λ. By Induction Hypothesis, $M|_{\vec{\psi}^k}, f(n) \models [!\chi]\varphi_1$ for all $\chi \in \mathcal{L}_{pal}$. Then $M|_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

- For $\varphi = \neg[!]\varphi_1$: Then, by the fact that rule $\langle ! \rangle$ has been applied, Λ contains

  $$(\vec{\psi}^k : n : \neg[!p]\varphi_1) \text{ for some } p \in P.$$

  The size of this labelled formula is less than the size of λ. By Induction Hypothesis, $M|_{\vec{\psi}^k}, f(n) \models \neg[!p]\varphi_1$ for some $p \in P$. Then $M|_{\vec{\psi}^k}, f(n) \models \varphi$. Therefore, $\mathcal{P}(\lambda)$ holds.

■

BIBLIOGRAPHY

[1] B. Brogaard and J. Salerno. Fitch's paradox of knowability. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Summer 2004.

[2] K. Fine. Propositional quantifiers in modal logic. *Theoria*, 3:336–346, 1970.

[3] M. Fitting. *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Company, 1983.

[4] J.D. Gerbrandy and W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6:147–169, 1997.

[5] R. Goldblatt. *Axiomatising the Logic of Computer Programming.* Springer-Verlag, 1982.

[6] R. Goré. Tableau methods for modal and tempral logics. In M. D'Agostino, D.M. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods.* Kluwer Academic Publishers, 1999.

[7] J. Hintikka. *Knowledge and Belief.* Cornell University Press, Ithaca, NY, 1962.

[8] R. Letz. First-order tableau methods. In M. D'Agostino, D.M. Gabbay, R. Hähnle, and J. Posegga, editors, *Handbook of Tableau Methods.* Kluwer Academic Publishers, 1999.

[9] C. Lutz. Complexity and succinctness of public announcement logic. To appear in the proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 06), 2006.

[10] J.-J.Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science.* Cambridge Tracts in Theoretical Computer Science 41. Cambridge University Press, Cambridge, 1995.

[11] G.E. Moore. A reply to my critics. In P.A. Schilpp, editor, *The Philosophy of G.E. Moore*, pages 535–677. Northwestern University, Evanston IL, 1942. The Library of Living Philosophers (volume 4).

[12] J.A. Plaza. Logics of public communications. In M.L. Emrich, M.S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216, 1989.

[13] R. M. Smullyan. *First-Order Logic.* Springer-Verlag, 1968.

[14] J.F.A.K. van Benthem. One is a lonely number: on the logic of communication. Technical Report PP-2002-27, ILLC, University of Amsterdam, 2002.

[15] J.F.A.K. van Benthem. Actions that make us know. Manuscript, 2005.

[16] H.P. van Ditmarsch and B.P. Kooi. The secret of my success. *Synthese*, 2006. To appear. Available as technical report OUCS-2004-10.

[17] H.P. van Ditmarsch, W. van der Hoek, and B.P. Kooi. Dynamic epistemic logic. Manuscript, 2006.

Philippe Balbiani, Andreas Herzig, and Tiago De Lima

Institut de recherche en informatique de Toulouse, Toulouse, France.

`{herzig,Tiago.Santosdelima}@irit.fr`

Hans van Ditmarsch

Computer Science, University of Otago, PO Box 56, Dunedin, New Zealand.

`hans@cs.otago.ac.nz`