Asynchronous Broadcast Logic

Sophia Knight[a], Bastien Maubert[b], François Schwarzentruber[c]

[a] *Uppsala University, Sweden*
[b] *Università degli Studi di Napoli Federico II*
[c] *ENS Rennes / IRISA, France*

Abstract

We propose a variant of public announcement logic for asynchronous systems. To capture asynchrony, we introduce two different modal operators for sending and receiving messages. The natural approach to defining the semantics leads to a circular definition, but we describe two restricted cases in which we solve this problem. The first case requires the Kripke model representing the initial epistemic situation to be a finite tree, and the second one only allows announcements from the existential fragment. After establishing some validities, we study the model checking problem and the satisfiability problem in cases where the semantics is well-defined, and we provide several complexity results.[1]

Keywords: Epistemic logic, Public announcements, Asynchronous systems

1. Introduction

Asynchrony plays a central role in distributed systems such as robotic rescue teams, smart cities, autonomous vehicles, etc. In such systems, there may be an unpredictable delay between sending and receiving messages, and there is not always access to a centralized clock. Recently, with the proliferation of multi-agent systems (MAS) where independent agents interact, communicate, and make decisions under imperfect information, modeling the evolution of knowledge as informative events occur has become increasingly important for both verification and design. For instance, it is often crucial to know whether an agent has received some information, so it would be highly desirable to be able to analyze messages such as "agent a knows that agent b received message m", *i.e.*, we want to model

$$\textbf{\textit{messages with epistemic content,}} \tag{1}$$

and because we are considering automated systems where agents do not lie, make logical mistakes, or have inaccurate factual information (for instance autonomous vehicles communicating about their position), we also make the classic assumption that

$$\textbf{\textit{messages are true when they are sent.}} \tag{2}$$

[1] We acknowledge support from ERC project EPS 313360.

Public announcement logic (PAL) [Plaza, 2007] is one of the first and most influential proposals for modelling the relationship between knowledge and announcements. In PAL, true announcements are made to a group of agents. This logic later led to the powerful and much studied dynamic epistemic logic (DEL) [van Ditmarsch et al., 2007], which can describe more complex forms of communication, such as semi-private announcements, private announcements, and much more. However, both these logics assume synchronicity: In PAL messages are immediately received by all agents at the same time, as soon as they are sent, and in DEL agents may perceive events differently, but events immediately change the epistemic state of agents as soon as they occur. In asynchronous settings, however, messages are not delivered instantly, and agents may receive them at different points in time, making PAL intrinsically unfit for reasoning about such settings. This fact becomes even more evident when we consider that in PAL, every announcement immediately leads to *common knowledge*, while common knowledge is not achievable in asynchronous systems [Moses and Tuttle, 1988; Halpern and Moses, 1990]. The only work we know of considering how DEL can capture asynchrony is [Dégremont et al., 2011], but in this logic an agent can only consider possible "future" events if they do not change her epistemic state. This is related to the *principle of inertia* [van Lambalgen, 2010; Braüner et al., 2016], which states that in absence of any observation, one assumes that nothing has happened. This assumption is natural in contexts where agents believe that they can observe all events at the time of their occurrence. In asynchronous systems however, agents should know that even when they do not observe anything, or when they do not receive any messages, it is possible that messages are being sent and received by other agents. Therefore the inertia principle does not apply in our setting, and agents should be able to

imagine possible pending messages. (3)

Our aim is to propose a logic in the spirit of PAL for reasoning about (1) epistemic messages in asynchronous systems, (2) that are true at the time of announcement, and where (3) agents can imagine messages that have been sent but not yet received.

Because this is an ambitious endeavour, we make a few assumptions to start as simply as possible:

Broadcasts:	all messages are sent to every agent
External source:	messages are emitted by an external, omniscient source
FIFO:	messages are received in the order they are sent.

The first assumption comes from PAL, and is natural in the context of smart cities for instance, where autonomous vehicles broadcast their current position or direction. The second one is a choice made to simplify the syntax by not having to model the origin of announcements, as in PAL. Announcements from an external, omniscient source can in some cases be used to model messages broadcast by agents within the system, in particular, an omniscient outside agent broadcasting that agent a knows φ is in many situations equivalent to agent a broadcasting φ to the other agents within the system. This captures the fact that in order for an agent within the system to make a true announcement, she should know that the announcement is true before she broadcasts it. Thus upon receiving an announcement made by agent a, another agent learns both the announcement and the fact that a knows it, modelled within the framework where announcements come from an external source by the announcement of $K_a\psi$ (K_a being the classic knowledge operator of epistemic logics, see for instance Fagin et al. [2004]). However, depending on assumptions about the agents' epistemic and reasoning capacities, this way of modelling agents' announcements may not be completely faithful to the real situation. We discuss this issue briefly in the future work section.

Concerning the last assumption, FIFO is a simple but classic scheme of communication in asynchronous systems (see for instance Yu and Gouda [1982]; Brand and Zafiropulo [1983]; Chambart and Schnoebelen [2008]).

Figure 1 depicts the architecture of such a system with three autonomous agents receiving messages from a public channel and reading them when they are ready to process them. To represent the fact that agents read messages in the order they were sent, we provide each one with a private FIFO channel. Each copy receives the same messages from the public channel, in the order in which they are announced, but the moment at which these messages are read differs from one agent to another.

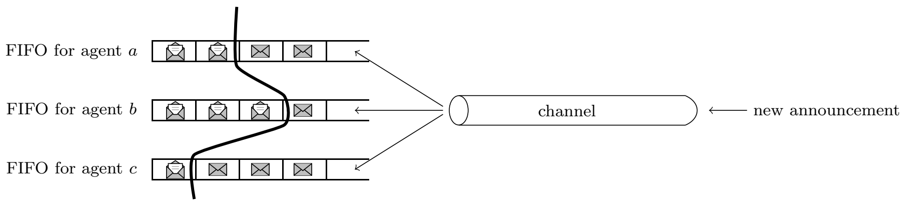


Figure 1: Agent architecture

In PAL, messages are received at the same time they are sent, and thus the announcement operator combines both sending and receiving. In contrast, in our setting, messages are not received immediately and they may be received at different times by different agents. The syntax reflects this aspect by providing both a sending operator, which adds new messages to the public channel, and a receiving operator for each agent, which allows her to read the first message in her FIFO queue that she has not read yet. Thus, in our logic, we provide the following modal constructions:

- $\langle\psi\rangle\varphi$, which means "$\psi$ is currently true, and after its announcement, φ holds";
- $\bigcirc_a\varphi$, which means "after agent a reads the next announcement, φ holds";
- $K_a\varphi$, which means "agent a knows that φ holds".

Interestingly, the natural semantics for this logic presents a challenging problem of circular definition: In order to define the truth of epistemic formulas, we classically quantify over the set of all states that the agent considers possible, where states include the current content of the public channel and pointers to the last message read by each agent. But some states are not *consistent* and must not be considered: intuitively a state is consistent if the announcements it contains were true at the time they were made. Therefore, defining consistent states requires defining the truth of formulas, and vice versa. PAL presents a similar circularity, as the definition of the update of a model by an announcement and the definition of the truth values are mutually dependent, thus this phenomenon is not inherent to the asynchronous setting. However, only asynchrony makes it a problem. Indeed in PAL a simple definition by mutual induction is possible. In an asynchronous setting, however, agents do not know what or how many messages other agents have received; in particular, an agent may consider it possible that some formula has been announced that is bigger than all those that have actually been announced, which makes a mutual induction impossible for lack of a decreasing quantity. This circularity problem is inherent in the asynchronous setting, and is independent from the assumptions of broadcasts, external source and FIFO described above. It only depends on the assumptions that announcements can talk about knowledge (1), that they must be true (2), and that agents can imagine pending messages (3).

We partially solve the issue by defining three restricted cases in which we manage to avoid circularity. The first one requires the Kripke model representing the initial epistemic situation to be a finite tree; the second one only allows announcements from the existential fragment of our logic, and the third one makes the assumption that only a bounded number of announcements can be made during each time unit (a strong form of non-Zeno assumption), and that agents have access to a global clock. In the second case, the semantics is defined thanks to an application of the Knaster-Tarski fixed point theorem [Tarski et al., 1955].

We then discuss some properties of our logics, compare them to PAL, and establish some validities that hold whenever the semantics is well-defined; we also study the model checking problem for our logic and establish the following complexity results:

Restrictions	Complexity of model checking
Propositional announcements	PSPACE-complete
Finite tree initial models	in PSPACE
Announcements from the existential fragment	in EXPTIME, PSPACE-hard

Finally, we study the satisfiability problem in the case of propositional announcements, and we establish that it is NEXPTIME-complete.

The paper is organized as follows. In Section 2, we recall (synchronous) public announcement logic. In Section 3 we present our logic, and discuss the circularity problem that arises from the definition of the semantics, and present an example to illustrate the logic. In Section 4 we exhibit cases where it can be solved. We then present some properties in Section 5 (a comparison with PAL and some validities), we study the model checking problem in Section 6 and the satisfiability problem in Section 7. Finally we discuss related work in Section 8 and future work in Section 9.

This paper is an extended version of [Knight et al., 2015]. Section 7 is completely new material. The other sections are based on the older version of this paper but include more details, improvements and clarifications.

2. Background: Public Announcement Logic

In this section, we recall Public Announcement Logic (PAL) [Plaza, 2007], the classic logic for synchronous public announcements. Let $\mathcal{AP}$ be a countable infinite set of *atomic propositions*, and let Ag be a finite set of *agents*.

Definition 1. *The syntax of PAL is given by the following grammar:*

$$\varphi ::= p \mid (\varphi \wedge \varphi) \mid \neg\varphi \mid K_a\varphi \mid \langle\varphi\rangle_{PAL}\varphi$$

where p ranges over $\mathcal{AP}$ and a ranges over Ag.

(a) $\boxed{u : \neg p} \xleftarrow{\;a\;} \boxed{w : p} \xrightarrow{\;a, b\;} \boxed{v : p}$

$\boxed{w : p} \xrightarrow{\;a, b\;} \boxed{v : p}$ (b)

Figure 2: A Kripke model and an updated Kripke model

The intuitive meaning of the last two operators is the following: $K_a\varphi$ means that agent a knows φ, and $\langle\psi\rangle_{\text{PAL}}\varphi$ means that ψ is true and after ψ has been publicly announced and publicly received by all the agents (meaning that all agents know that each agent received the message), φ holds. We define the following usual abbreviations: $\bot := p \wedge \neg p$, $\top := \neg\bot$, $\varphi \vee \varphi' := \neg(\neg\varphi \wedge \neg\varphi')$ and the dual of the knowledge modality, $\hat{K}_a\varphi := \neg K_a \neg\varphi$, which reads as "agent a considers it possible that φ holds".

The semantics of PAL relies on classic Kripke models and the *possible worlds* semantics, widely used in logics of knowledge [Fagin et al., 2004].

Definition 2. *A* Kripke model *is a tuple* $\mathcal{M} = (W, \{\rightarrow_a\}_{a\in Ag}, \Pi)$*, where:*

- W *is a non-empty finite set of* worlds*,*
- *for each* $a \in Ag$*,* $\rightarrow_a \subseteq W \times W$ *is an* accessibility relation *for agent* a*,*
- $\Pi : W \longrightarrow 2^{\mathcal{AP}}$ *is a* valuation *on worlds.*

A *pointed model* $(\mathcal{M}, w)$ is a model $\mathcal{M}$ with a specified world w. For the sake of generality we allow arbitrary relations and not only equivalence relations as traditional in epistemic logic [Fagin et al., 2004; van Ditmarsch et al., 2007].[2]

Example 1. *Let us consider the Kripke model of Figure 2(a), where w, u and v are worlds, a and b are agents and p is an atomic proposition. The arrows represent the agents' accessibility relations. At world w, agent a considers u and v possible, and agent b considers only world v possible. Now assume that p is announced in $(\mathcal{M}, w)$. This is possible, as p holds in w. As a result of this announcement, all agents know that p holds, and thus the resulting epistemic situation is obtained by removing all worlds where p does not hold,* i.e.*, u. This updated model is represented in Figure 2(b).*

The update of a Kripke model by an announcement and the semantics of PAL are defined by mutual induction:

Definition 3. *The update of a Kripke model $\mathcal{M}$ with an announcement ψ is the Kripke model* $\mathcal{M}^\psi = (W^\psi, \{\rightarrow_a^\psi\}_{a\in Ag}, \Pi^\psi)$ *where:*

- $W^\psi = \{u \in W \mid \mathcal{M}, u \models \psi\}$*;*
- $\rightarrow_a^\psi = \rightarrow_a \cap (W^\psi \times W^\psi)$ *for all agents* a*;*
- Π^ψ *is the function* Π *restricted to* W^ψ*,*

and the truth condition relation $\mathcal{M}, w \models \varphi$ (read as φ is true in $\mathcal{M}, w$) is defined as:

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \textit{if} & p \in \Pi(w) \\
\mathcal{M}, w \models (\varphi_1 \wedge \varphi_2) & \textit{if} & \mathcal{M}, w \models \varphi_1 \textit{ and } \mathcal{M}, w \models \varphi_2 \\
\mathcal{M}, w \models \neg\varphi & \textit{if} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models K_a\varphi & \textit{if} & \textit{for all } u \textit{ such that } w \rightarrow_a u,\ \mathcal{M}, u \models \varphi \\
\mathcal{M}, w \models \langle\psi\rangle_{\text{PAL}}\varphi & \textit{if} & \mathcal{M}, w \models \psi \textit{ and } \mathcal{M}^\psi, w \models \varphi.
\end{array}
$$

[2] This allows for alternative semantics of knowledge such as S4 [Hintikka, 1962], S4.2 [Lenzen, 1978], S4.3 [van der Hoek, 1990], S4.4 [Kutschera, 1976], KD45 for beliefs [Fagin et al., 2004], etc.

Observe that the definition by structural induction on φ is sound: defining the semantics of $\langle\psi\rangle_{\text{PAL}}\varphi$ requires having defined the update of a Kripke model by announcement ψ, which only requires having defined the semantics of ψ, a subformula of φ.

Example 2. *Let $\mathcal{M}$ be the model of Figure 2(a). We have $\mathcal{M}, w \models \langle p\rangle_{\text{PAL}} K_a p$. Indeed, the announcement p is true, that is $\mathcal{M}, w \models p$, and $\mathcal{M}^p, w \models K_a p$. Note that $\mathcal{M}^p$ is the model of Figure 2(b).*

The model-checking problem of public announcement logic is P-complete (the membership in P is established in Benthem [2011] and P-hardness in Schnoebelen [2002][3]) and the satisfiability problem for public announcement logic is PSPACE-complete [Lutz, 2006]. A tableau proof system for PAL is provided in [Balbiani et al., 2010].

3. Asynchronous Broadcast Logic

In this section we present our framework for reasoning about asynchronous epistemic announcements in a public channel. As in Section 2, $\mathcal{AP}$ is a countable infinite set of atomic propositions, and Ag is a finite set of agents. For pedagogical reasons, we first introduce models, then the syntax and finally the semantics of our logic, even though by doing so we need to refer to the language before we formally define it.

3.1. Models

Agents start with an initial state of knowledge of the world, which is modelled by an initial pointed epistemic model, or Kripke model. Then true announcements are made by some external entity, and sent in the public channel. The whole sequence of announcements that have been made up to the present moment is modelled as a sequence of formulas from our logic, whose syntax we introduce later in Section 3.2. Agents read these messages independently, possibly at different times, but in FIFO order. To represent which messages each agent has already read, and thus which ones remain to be read, we simply map each agent to the number of announcements she has read. Such a mapping is called a *cut*.

3.1.1. Initial Kripke model

An *initial model* is given as a Kripke model $\mathcal{M} = (W, \{\rightarrow_a\}_{a\in Ag}, \Pi)$, as defined in Definition 2. It represents the initial knowledge of agents before any announcements are made, and it corresponds to the notion of initial knowledge in [Raynal, 2013], p. 5. In practice, $(\mathcal{M}, w)$ is directly provided by the modeller or inferred from what agents perceive [Balbiani et al., 2013; Gasquet et al., 2015].

3.1.2. Announcements

We consider that, in a given scenario, not every formula may be announced, but rather that there is a certain set of relevant announcements. Furthermore, we allow the number of times an announcement can be made to be bounded. To represent this, we define the notion of *announcement protocols* (ABL is the language defined in Section 3.2).

[3]Epistemic logic is an extension of the fragment of CTL with only the next operators AX and EX, proven to be P-hard.

Definition 4. *An* announcement protocol *is a multiset of formulas in* ABL*, where the multiplicity of an element ψ is either an integer or ∞.*

Example 3. *The reader may imagine a card game where it is only possible to announce 'agent a has a heart card' once and 'agent a does not know whether agent b has a heart card or not' twice. We let the proposition $\heartsuit_a$ mean "agent a has a heart card," and define the announcement protocol to be $\{\{\heartsuit_a\ ,\ \ \hat{K}_a\heartsuit_b \wedge \hat{K}_a\neg\heartsuit_b\ ,\ \ \hat{K}_a\heartsuit_b \wedge \hat{K}_a\neg\heartsuit_b\}\}$.*

Given an announcement protocol $\mathcal{A}$, we denote by $\mathsf{Seq}(\mathcal{A})$ the set of finite sequences $\sigma = [\varphi_1, \ldots, \varphi_k]$ such that the multiset $\{\{\varphi_1, \ldots, \varphi_k\}\}$ is a submultiset of $\mathcal{A}$. We define the *size* of a sequence σ as $|\sigma| := \sum_{i=1}^{k} |\varphi_i|$. For $\sigma, \sigma' \in \mathsf{Seq}(\mathcal{A})$, we write $\sigma \leq \sigma'$ if σ is a prefix of σ'. The sequence $\sigma|_k$ is the prefix of σ of length k. Given a formula φ and a sequence of formulas σ, $\varphi{::}\sigma$ (resp. $\sigma{::}\varphi$) is the sequence obtained by adding φ at the beginning (resp. at the end) of σ.

3.1.3. States

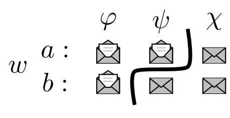
We now define the set of possible states of the models in which the formulas of our logic will be evaluated.

Definition 5. *Let $\mathcal{M}$ be an initial model and $\mathcal{A}$ an announcement protocol. We define the set of* possible states $\mathcal{S}_{\mathcal{M},\mathcal{A}}$ *as follows:*

$$\mathcal{S}_{\mathcal{M},\mathcal{A}} = \{(w, \sigma, c) \mid w \in W, \sigma \in \mathsf{Seq}(\mathcal{A}) \text{ and } c : Ag \longrightarrow \{0, ..., |\sigma|\}\}\,.$$

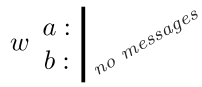
The first element of a state represents the world the system is in. The second element is the list of messages that have already been announced. The last element, c, is called a *cut*, and for each $a \in Ag$, $c(a)$ is the number of announcements of σ that agent a has received so far. Given two cuts c and c', we write $c < c'$ if for all a, $c(a) \leq c'(a)$ and there exists b such that $c(b) < c'(b)$: in other words, $c < c'$ if all agents have received at least as many messages in c' as in c, and at least one agent received strictly more messages in c'. Typical elements of $\mathcal{S}_{\mathcal{M},\mathcal{A}}$ are denoted S, S', etc.

Example 4. *Consider the state $S = (w, [\varphi, \psi, \chi], c)$ where $c(a) = 2$ and $c(b) = 1$. S represents the situation where in initial world w, the sequence $[\varphi, \psi, \chi]$ of formulas has been announced, agent a has received φ and ψ, and agent b has only received φ. Only χ remains in the queue of a and has not been read yet, and only ψ and χ remain in the queue of b. We represent S as follows:*

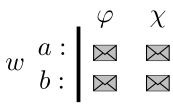


and we may also write $S = (w, [\varphi, \psi, \chi],\ \begin{smallmatrix} a & \mapsto & 2 \\ b & \mapsto & 1 \end{smallmatrix}\ )$.

Example 5. *Consider the state $S = (w, \epsilon, \mathbf{0})$, where ϵ denotes the empty sequence of formulas and $\mathbf{0}$ is the function that assigns 0 to all agents. S represents an initial world w in which no announcement has been made (and therefore no announcement has been received either). It can be represented as follows:*



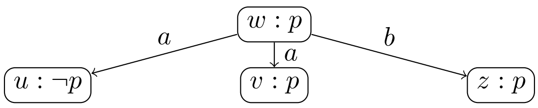
Example 6. *State* $S = (w, [\varphi, \chi], \mathbf{0})$*, which represents the situation where in initial world* w*,* φ *and* χ *have successively been announced, but neither agent* a *or agent* b *received any announcement yet. We depict it as follows:*



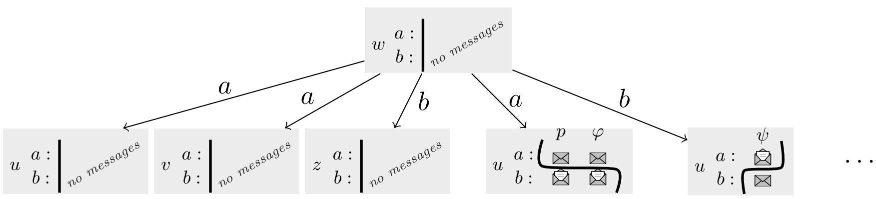
3.1.4. Consistent states

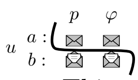
Definition 5 allows for all combinations of worlds, sequences of announcements allowed by the announcement protocol, and cuts. This definition is an over-approximation of the set of states we want to consider: indeed, because announcements must be true, some of the states in $\mathcal{S}$ are *inconsistent.* For example, suppose that w is a world in $\mathcal{M}$ where p does not hold. Because only true announcements can be made, p cannot be announced in world w, and thus the state $(w, [p], \mathbf{0})$ is inconsistent.

Example 7. *Let us consider the following initial model, where* w, u, v *and* z *are worlds,* a *and* b *are agents and* p *is a proposition. The arrows represent the agents' accessibility relations, before any announcements have been made. So at world* w*, agent* a *considers* u *and* v *possible, and agent* b *considers world* z *possible.*



Now assuming that the announcement protocol $\mathcal{A}$ *contains* p*,* φ *and* ψ*, a partial depiction of the asynchronous model* $\mathcal{M} \otimes \mathcal{A}$ *is below. We depict the states* w, u, v*, and* z *where no announcement has been made, as well as copies of* u *where two different sequences of announcements have been made, and received in one state by agent* b *and in a different state by agent* a*. Of course, the entire model* $\mathcal{M} \otimes \mathcal{A}$ *is infinite so we do not depict all the states here.*



State  is not consistent because p has been announced even though p is not true in u. This notion of inconsistency is the source of the circularity problem, as we discuss in Section 3.4. For now, we define the relations that capture which states an agent considers possible before removal of inconsistent ones.

3.1.5. Pre-accessibility relation and asynchronous pre-model

We now define, for each agent, a *pre-accessibility relation* that does not yet take consistency into account, but is only based on the agents' accessibility relations in the initial model, and the messages it has already read.

Definition 6. *The* pre-accessibility relation *for agent a, written R_a, is defined as follows: given $S = (w, \sigma, c)$ and $S' = (w', \sigma', c')$, we have SR_aS' if:*

1. $w \to_a w'$, *and*
2. $\sigma|_{c(a)} = \sigma'|_{c'(a)}$.

The first clause simply says that for S' to be considered possible by a when in S, world w' must be considered possible by a from w. The second clause says that agent a is aware of, and only aware of, messages that she has received: therefore she can only consider possible states where she has received exactly the same messages. Then, because the principle of inertia does not apply to the asynchronous setting, she can imagine any possible sequence of pending announcements, as long as it is compatible with the announcement protocol. Also, as she has no information about what messages the other agents have received, $c'(b)$ can be anything if $b \neq a$: agent a considers it possible that b received more, or fewer, messages than she actually has. Note that the second clause also implies $c(a) = c'(a)$.

Given an initial model $\mathcal{M}$ and an announcement protocol $\mathcal{A}$, we define the *asynchronous pre-model* $\mathcal{M} \otimes \mathcal{A} := (\mathcal{S}_{\mathcal{M},\mathcal{A}}, \{R_a\}_{a \in Ag})$, where $\mathcal{S}_{\mathcal{M},\mathcal{A}}$ is the set of possible states, and for $a \in Ag$, R_a is the pre-accessibility relation for agent a.

3.2. Language

We now introduce the syntax of our logic, which we call *Asynchronous Broadcast Logic*, or ABL for short. Note that we do not use the term "public announcement" in the name of our logic as it has a strong synchronous connotation: public announcements are often thought of as becoming common knowledge the moment they are made.

Definition 7 (Syntax)**.** *The set of* ABL*-formulas is given by the following grammar:*

$$\varphi ::= p \mid (\varphi \wedge \varphi) \mid \neg\varphi \mid K_a\varphi \mid \langle\varphi\rangle\varphi \mid \bigcirc_a \varphi,$$

where p ranges over $\mathcal{AP}$ and a ranges over Ag.

The intuitive meaning of the last three operators is the following: $K_a\varphi$ means that agent a knows φ, $\langle\psi\rangle\varphi$ means that ψ is true and after ψ has been put on the public channel, φ holds, and $\bigcirc_a\varphi$ means that agent a has a pending message, and after she has received and read it, φ holds. We define the dual of the announcement operator: $[\psi]\varphi := \neg\langle\psi\rangle\neg\varphi$, meaning that if ψ is true, then φ holds after its announcement. $|\varphi|$ is the length of φ, and we denote by *propositional formula* a formula that uses no modalities, *i.e.*, containts no occurrences of K_a, $\langle\varphi\rangle$, or $\bigcirc_a$.

In (synchronous) public announcement logic (see Definition 1), the operator $\langle\psi\rangle_{\text{PAL}}$ captures both the broadcast and the reception of an announcement ψ, because in the synchronous setting, sending and reception occur simultaneously. In our asynchronous setting, not only can sending and reception occur at different times, but also different agents may receive the same message at different times. Therefore we capture the broadcast of a formula ψ with operator $\langle\psi\rangle$, while agent a's reception of a broadcasted formula is captured by the operator $\bigcirc_a$.

3.3. Truth conditions

For the rest of the section, we fix an initial model $\mathcal{M}$ and an announcement protocol $\mathcal{A}$. As discussed in Section 3.1.4, some possible states from Definition 5 are inconsistent, because they contain announcements that were not true at the time they were announced. Also, because agents should not consider inconsistent states possible, we described how defining consistency is necessary to define the semantics of the knowledge operator, which in turn is necessary to define the consistency of states that contain epistemic announcements, hence a circularity problem.

We describe in Figure 3 the definition of consistency (represented with symbol $\checkmark$) as well as truth conditions for our logic. This definition is circular, and therefore the semantics as presented here is not well-founded, although it conveys the intended meaning of our operators. In the next section we will describe restricted cases in which we can provide a semantics that is well-defined.

Truth conditions for consistency:

$(w, \epsilon, \mathbf{0}) \models \checkmark$		$(always)$
$(w, \sigma, c) \models \checkmark$	if	there is $c' < c$ s.t. $(w, \sigma, c') \models \checkmark$, or
		$\sigma = \sigma'::\psi, (w, \sigma', c) \in \mathcal{S}_{\mathcal{M},\mathcal{A}}$,
		$(w, \sigma', c) \models \checkmark$ and $(w, \sigma', c) \models \psi$

Truth conditions for formulas:

$(w, \sigma, c) \models p$	if	$p \in \Pi(w)$
$(w, \sigma, c) \models (\varphi_1 \wedge \varphi_2)$	if	$(w, \sigma, c) \models \varphi_1$ and $(w, \sigma, c) \models \varphi_2$
$(w, \sigma, c) \models \neg\varphi$	if	$(w, \sigma, c) \not\models \varphi$
$(w, \sigma, c) \models K_a\varphi$	if	for all S' s.t. $(w, \sigma, c)R_aS'$ and $S' \models \checkmark$, $S' \models \varphi$
$(w, \sigma, c) \models \langle\psi\rangle\varphi$	if	$\sigma::\psi \in \mathsf{Seq}(\mathcal{A})$, $(w, \sigma, c) \models \psi$ and $(w, \sigma::\psi, c) \models \varphi$
$(w, \sigma, c) \models \bigcirc_a\varphi$	if	$c(a) < \lvert\sigma\rvert$ and $(w, \sigma, c^{+a}) \models \varphi$
		where $c^{+a}(b) = \begin{cases} c(b) & \text{if } b \neq a \\ c(b)+1 & \text{if } b = a \end{cases}$

Figure 3: Consistency and semantics

The intuitive meaning of $(w, \sigma, c) \models \checkmark$ is that the state (w, σ, c) is consistent, that is, all announcements in σ were true when they were made. The first clause is obvious: the initial state where no announcement has been made is consistent. The second clause gives two possibilities for a state to be consistent. Either there was an earlier consistent state (w, σ, c') in which some agents received some already announced formulas, increasing the cut from c' to c, or a new, *true* announcement ψ has been made from an earlier consistent state, extending the history from σ' to $\sigma'::\psi$.

For the formulas, the first three clauses are straightforward. The fourth clause says that agent a knows φ if φ holds in all consistent states that she considers possible. The fifth clause says that $\langle\psi\rangle\varphi$ holds in a state S if ψ can be announced (it is true in S), and φ holds in the state obtained by adding ψ to the public channel. The last clause says that $\bigcirc_a\varphi$ holds if agent a has at least one unread announcement in the channel, and φ holds after she reads the first unread message.

3.4. Circularity

By observing the truth conditions for consistency and for formulas in Figure 3, one can see that defining whether a state is consistent requires one to define whether an announcement can be made, and this requires the semantics of our logic to be defined. But to define the semantics of the knowledge operators, we need to define which *consistent* states are considered possible by the agent, which requires us to define which states are consistent, hence the circularity.

Let us consider the following example, where $Ag = \{a\}$. Let the initial model be $\mathcal{M} = (W, \rightarrow_a, \Pi)$ where $W = \{w\}$, $\rightarrow_a = \{(w, w)\}$ and $\Pi(w) = \emptyset$, and let the announcement protocol be $\mathcal{A} = \{\{K_a p\}\}$. According to Figure 3, we have: $(w, [K_a p], \mathbf{0}) \models \checkmark$ iff $(w, \epsilon, \mathbf{0}) \models K_a p$. But, as $(w, \epsilon, \mathbf{0}) R_a (w, [K_a p], \mathbf{0})$, the definition of the truth value of $(w, \epsilon, \mathbf{0}) \models K_a p$ depends on the truth value of $(w, [K_a p], \mathbf{0}) \models \checkmark$. To sum up, the definition of $(w, [K_a p], \mathbf{0}) \models \checkmark$ depends on itself.

The circularity problem depends on assumptions (1), (2), and (3) from the introduction:

- Announcements can be epistemic
- Announcements are true
- Agents can imagine pending messages

If one of these assumptions is dropped, the circularity problem is easily solved: if announcements do not need to be true, then all states are consistent; if announcements are only propositional formulas, consistency of a state (w, σ, c) can be trivially checked by evaluating all propositional formulas in σ in the world w. The last point is only a little bit less obvious: if agents cannot imagine pending annoucements, then the definition of the pre-accessibility relation R_a for agent a (see Definition 6) is that $(w, \sigma, c) R_a (w', \sigma', c')$ if $w \rightarrow_a w'$, $c(a) = c'(a)$ and $\sigma|_{c(a)} = \sigma'$: the only sequence of announcements that she considers possible is the one she has already received. In that case, the length of the sequence of announcements $|\sigma|$ together with the size of the formula to evaluate can be used to define truth conditions for consistency and for formulas by induction. Indeed, evaluating a formula $K_a \varphi$ in a state (w, σ, c) only requires evaluating the consistency of states (w', σ', c') such that $|\sigma'| \leq |\sigma|$, which in turn only requires evaluating formulas $\psi \in \sigma$ in states (w', σ'', c') where σ'' is a strict prefix of σ.

We also note that it is possible to solve the circularity problem by only constraining the last assumption instead of completely dropping it. Indeed, under a bounded *non-Zeno behaviour* assumption (only a bounded finite number of discrete events occur in a finite time), and assuming a global clock that is common knowledge, the imagination of the agents is sufficiently constrained to solve the circularity problem rather easily (see Appendix B).

In relation with the above discussion, we point out that the circularity problem does *not* depend on the following assumptions:

- Announcements are broadcast
- Announcements are made by an external source
- Announcements are received in FIFO order.

In Section 4 we will describe several restricted settings in which we manage to overcome this problem. But first we present a small example to better understand the intuitions behind our logic.

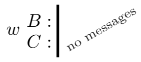
3.5. Example

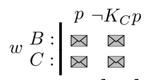
We consider two agents $Ag = \{B, C\}$, where B stands for Bonnie and C for Clyde. Bonnie and Clyde go to rob a bank, and Bonnie stays in the car while Clyde goes to the vault. At noon, Bonnie notices that Clyde left the paper with the secret code to open the vault in the car. She uses her smartphone to broadcast the code on their chat group. But Clyde has also realized that he forgot the paper, and before he receives Bonnie's message he sends a message saying that he does not know the code.

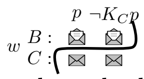
In the following, let p represent the fact that the secret code is 0000, and q the fact that the vault is open. The situation at noon is represented by the following initial pointed Kripke model $(\mathcal{M}, w)$:

$$\begin{tikzcd} u : p, q \arrow[loop below, "{B,C}"] \arrow[r, leftrightarrow, "B"] & w : p, \neg q \arrow[loop below, "{B,C}"] \arrow[r, leftrightarrow, "C"] & v : \neg p, \neg q \arrow[loop below, "{B,C}"] \end{tikzcd}$$

In the initial world w, Bonnie knows p, *i.e.*, she knows the code, but Clyde does not. On the other hand, Clyde knows that q is not true, *i.e.*, he knows that the vault is closed, but Bonnie does not (Clyde could have memorized the code before leaving the car, and thus he might have opened the vault). In fact, we observe that initially Bonnie knows that "the vault is open if, and only if, Clyde knows the code" or, in epistemic logic, $K_B(q \leftrightarrow K_C p)$.

The initial state at noon is 

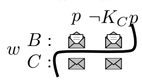
. In our scenario, first p is announced by Bonnie, and then $\neg K_C p$ is announced by Clyde. The actual state becomes 

. Intuitively, since only true announcements are made, we see that $\neg K_C p$ can only be announced before Clyde receives the announcement of p. We would like to verify whether, after Bonnie receives both announcements but Clyde receives neither (the signal inside the bank is weak), that is in state 

, Bonnie knows $\neg q$, *i.e.*, does she know that the vault is not open, and what does she know about Clyde's knowledge. In fact, we can prove that the following holds:

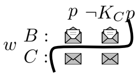
$$(w, \epsilon, \mathbf{0}) \models \langle p \rangle \langle \neg K_C p \rangle \bigcirc_B \bigcirc_B (K_B \neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p) \qquad (4)$$

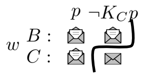
The meaning is that, after both announcements have been made and received by Bonnie but not by Clyde,

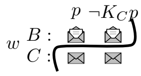
- Bonnie knows that the vault is not open: intuitively, because Clyde told her he didn't have the code, and thus could not have opened it (recall that q represents the situation at noon, *i.e.*, before Bonnie announced p). In state 

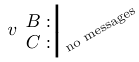
, all *consistent* possible states for B are of the form (w, σ, c) for some σ and c: they share the same world w in which q does not hold. Indeed, Bonnie initially considers world u possible, but states with world u and announcement $\neg K_C p$ are not consistent. Therefore, Bonnie knows $\neg q$.

- Bonnie does not know whether Clyde knows the code (because she does not know whether Clyde received her message).

  In state 

, Bonnie considers state 

as possible, and in this state Clyde knows p. But Bonnie also considers possible state 

, in which Clyde considers state 

possible, and thus does not know p.

We note that this example highlights the differences between asynchronous broadcast logic and (synchronous) public announcement logic. Since sending and receiving occur at the same time in PAL, we can informally translate the asynchronous broadcast formula $\langle p\rangle\langle\neg K_C p\rangle \bigcirc_B \bigcirc_B(K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$ to the PAL formula $\langle p\rangle_{\mathrm{PAL}}\langle\neg K_C p\rangle_{\mathrm{PAL}}(K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B\neg K_C p)$. Assuming the same initial Kripke model and state w, we first notice that in PAL any formula of the form $\langle p\rangle_{\mathrm{PAL}}\langle\neg K_C p\rangle_{\mathrm{PAL}}\varphi$ is false because after a proposition p is announced, $K_C p$ holds in any circumstances, so that $\neg K_C p$ cannot be announced. We can try to simulate the state where Bonnie has received both announcements but Clyde received neither by using private announcements, made to Bonnie but not to Clyde. Consider the trivial translation of our ABL formula into a formula with private announcements:

$$\langle p\rangle_B\langle\neg K_C p\rangle_B(K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B\neg K_C p),$$

where $\langle\varphi\rangle_B$ means that φ is sent to B and not to C. In all versions of PAL with any variant of synchronous private or semi-private announcement (*e.g.* Gerbrandy and Groeneveld [1997]; Baltag et al. [1998]; Baltag and Moss [2004]; Baltag et al. [2008]), this formula is still false in $(\mathcal{M}, w)$: because Bonnie receives $\neg K_C p$ immediately when it is sent, Clyde cannot receive p between the announcement of $\neg K_C p$ and its reception by Bonnie, so that Bonnie knows that Clyde does not know p. Thus, this example shows that there is no obvious translation from asynchronous broadcast logic to a variant of DEL, and that asynchronous broadcast logic is indeed quite different from DEL.

The proof of (4) can be found in Appendix A. Note that here we anticipate the fact that we are in one of the cases where we can solve the circularity problem: indeed, all announcements are in the existential fragment of our language (see Section 4.2).

4. Solving the circularity problem

In this section we show how we solve the circularity problem identified in the last section for several restricted cases.

4.1. When the initial model is a finite tree

If we assume in the initial model $\mathcal{M} = (W, \{\rightarrow_a\}_{a\in Ag}, \Pi)$ the relation $\bigcup_a \rightarrow_a$ forms a *finite tree* over W, then the circularity problem can be avoided. In this case, we can define a well-founded order on tuples of the form (w, σ, c, φ), where φ is either a formula in ABL or ✓, the idea being that a tuple (w, σ, c, φ) means '$w, \sigma, c \models \varphi$'.

Definition 8. *The order $\prec$ is defined as follows:*
$(w, \sigma, c, \varphi) \prec (w', \sigma', c', \varphi')$ *if either*

1 *w is a descendent of w' in $\mathcal{M}$,*

2 *or $w = w'$ and $|\sigma| + |\varphi| < |\sigma'| + |\varphi'|$,*

3 *or $w = w'$, $|\sigma| + |\varphi| = |\sigma'| + |\varphi'|$ and $c < c'$,*

where $|\checkmark| = 1$.

It is clear that $\prec$ is a well-founded order, and with this order Figure 3 forms a well-founded inductive definition of consistency and semantics of our language.

We detail the non-trivial cases. For the second clause of Figure 3, observe that by Point 3 of Definition 8, if $c' < c$ then $(w, \sigma, c', \checkmark) \prec (w, \sigma, c, \checkmark)$, and for all w, σ', c and ψ, by Point 2 of Definition 8, we have $(w, \sigma', c, \psi) \prec (w, \sigma'{::}\psi, c, \checkmark)$.

For the clause for $K_a\varphi$ of Figure 3, by Point 1 of Definition 8 we have that for all $\varphi, \sigma, \sigma', c, c'$, if w' is a child of w then $(w', \sigma', c', \varphi) \prec (w, \sigma, c, K_i\varphi)$.

Finally, for the clause for $\langle\psi\rangle\varphi$ of Figure 3, by Point 2 of Definition 8 we have that $(w, \sigma{::}\psi, c, \varphi) \prec (w, \sigma, c, \langle\psi\rangle\varphi)$ for all w, σ, c, φ and ψ (note that $|\langle\psi\rangle\varphi| = 1 + |\psi| + |\varphi|$).

Example 8. *Suppose that we have only one agent a. Let us consider initial model $\mathcal{M}$:*

$$\begin{tikzcd} & w : p \arrow[dl, "a"'] \arrow[dr, "a"] & \\ u : \neg p & & v : p \end{tikzcd}$$

In this model, p holds in the actual world w, but agent a does not know it. Assume that p can be announced at least once ($p \in \mathcal{A}$). We show that, as expected, after p is announced and agent a receives this announcement, agent a knows that p holds. Formally, we prove that, in $\mathcal{M} \otimes \mathcal{A}$, we have $(w, \epsilon, \mathbf{0}) \models \langle p\rangle \bigcirc_a K_a p$. To do so we in fact show that $(w, [p], a \mapsto 1) \models K_a p$, from which it follows that $(w, [p], \mathbf{0}) \models \bigcirc_a K_a p$, hence the desired result. By Definition 6 for pre-accessibility relations, every state S such that $(w, [p], a \mapsto 1) R_a S$ is of the form $S = (w', p{::}\sigma, a \mapsto 1)$, where $w' \in \{u, v\}$ and σ is a sequence of announcements. We must show that every such state either is inconsistent or satisfies p.

First, for $w' = u$. According to the clause for p in Figure 3, we have that $(u, \epsilon, \mathbf{0}) \not\models p$, and by the second clause in Figure 3 it follows that $(u, [p], \mathbf{0}) \not\models \checkmark$, from which it also follows also that $(u, [p], a \mapsto 1) \not\models \checkmark$ and $(u, p{::}\sigma, a \mapsto 1) \not\models \checkmark$, for any σ.

Now, for $w' = v$, by the first clause for p in Figure 3, it follows that for all states of the form $S = (v, p{::}\sigma, a \mapsto 1)$, $S \models p$, so that finally every state related to $(w, [p], a \mapsto 1)$ is either inconsistent or verifies p. Note that we could also prove that S is consistent.

In practice, this setting can be used as an approximation scheme: taking the tree unfolding of models and cutting them at level ℓ amounts to assuming that agents cannot reason about deeper nesting of knowledge. This approach is similar to the well known idea of *bounded rationality* [Jones, 1999], where it is assumed that due to computational limits, agents have only approximate, bounded information about other agents' knowledge, which is represented by allowing only finite-length paths in the Kripke model. We point out, however, that this method of approximation is only appropriate in certain settings. One issue is that it does not allow the accurate representation of transitive accessibility relations, where the leaves of an initial model of any depth ℓ may be reached just by evaluating a formula with one knowledge operator. This setting calls for more work to clarify what the finite tree restriction really captures.

4.2. Announcing existential formulas

Now, we again allow the initial model to be arbitrary. In particular, we may use one of the common models of knowledge, for example an initial model whose underlying frame is KD45 (relations are serial, transitive and Euclidean) or S5 (relations are equivalence relations); see Fagin et al. [2004]. However, we restrict the announcement protocol to the existential fragment of our logic, generated by the following rule:

$$\varphi ::= p \mid \neg p \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \hat{K}_a\varphi \mid \bigcirc_a \varphi \mid \langle \varphi \rangle \varphi$$

where p ranges over $\mathcal{AP}$ and a ranges over Ag. Formulas of the existential fragment are called *existential formulas*. If an announcement protocol contains only existential formulas, we call it an *existential announcement protocol*. For instance, the announcement protocol in Example 3 is existential.

Here we tackle the circularity problem by defining consistency and truth conditions separately. We first define as a fixed point the semantics of existential announcements in $\mathcal{A}$, together with consistency. In a second step we define the semantics of the full logic with existential announcements as described in Figure 3, using the fixed point to evaluate consistency.

We fix an initial model $\mathcal{M} = (W, \{\rightarrow_a\}_{a \in Ag}, \Pi)$ and an existential announcement protocol $\mathcal{A}$. Let B be the set of all pairs (S, φ) such that S is a state of $\mathcal{M} \otimes \mathcal{A}$ and φ is either a formula in $\mathcal{A}$ or $\checkmark$, the symbol for consistency. Observe that $(\mathcal{P}(B), \subseteq)$ forms a complete lattice. We now consider the function $f : \mathcal{P}(B) \rightarrow \mathcal{P}(B)$ defined in Figure 4. Function f takes a set Γ of truth pairs (pairs (S, φ) such that $S \models \varphi$), and extends it with the new truth pairs that can be inferred from Γ by applying each of the rules in Figure 3 once. For instance, if $(w, \sigma, c) \models \varphi$ and $(w, \sigma, c) \models \psi$, then $(w, \sigma, c) \models (\varphi \wedge \psi)$. That is, if (w, σ, c, φ) and (w, σ, c, ψ) are in Γ, then $(w, \sigma, c, (\varphi \wedge \psi))$ is in $f(\Gamma)$, which explains line 3 of Figure 4. Every other line of Figure 4 similarly follows from one of the truth conditions.

Now, as we restrict to existential formulas, it is easy to see that f is monotone, that is, if $\Gamma_1 \subseteq \Gamma_2$ then $f(\Gamma_1) \subseteq f(\Gamma_2)$. By the Knaster-Tarski theorem [Tarski et al., 1955], f has a least fixed point $\Gamma^* := \bigcup_{n \in \mathbb{N}} f^n(\emptyset)$.

We can now define the truth condition for consistency as: $S \models \checkmark$ if $(S, \checkmark) \in \Gamma^*$, and use Figure 3 to define the semantics of the language with existential announcements.

Remark 1. *If announcements of the form $K_a\varphi$ were allowed, we would have to add*

$$\left\{ (w, \sigma, c, K_a\varphi) \mid \begin{array}{l} \textit{for all } (w', \sigma', c') \textit{ such that } (w, \sigma, c) R_a (w', \sigma', c'), \\ \textit{either } (w', \sigma', c', \checkmark) \notin \Gamma \quad \textit{or} \quad (w', \sigma', c', \varphi) \in \Gamma \end{array} \right\}$$

to the definition of f in Figure 4. But then, if $(w, \sigma, c) R_a (w', \sigma', c')$ we would have:

- $(w, \sigma, c, K_a p) \in f(\emptyset)$;
- $(w, \sigma, c, K_a p) \notin f(\{(w', \sigma', c', \checkmark), (w', \sigma', c', \neg p)\})$

It would thus no longer hold that $f(\Gamma_1) \subseteq f(\Gamma_2)$ whenever $\Gamma_1 \subseteq \Gamma_2$. As f is clearly not a decreasing function either, we would not be able to apply the Knaster-Tarski theorem.

$$
\begin{aligned}
f(\Gamma) = \Gamma\ &\cup \big\{ (w,\sigma,c,p) \mid \ p \in \Pi(w) \ \big\} \\
&\cup \big\{ (w,\sigma,c,\neg p) \mid \ p \notin \Pi(w) \ \big\} \\
&\cup \big\{ (w,\sigma,c,(\varphi \wedge \psi)) \mid \ (w,\sigma,c,\varphi) \in \Gamma \ \text{ and } \ (w,\sigma,c,\psi) \in \Gamma \ \big\} \\
&\cup \big\{ (w,\sigma,c,(\varphi \vee \psi)) \mid \ ((w,\sigma,c,\varphi) \in \Gamma \ \text{ or } \ (w,\sigma,c,\psi) \in \Gamma \ \big\} \\
&\cup \left\{ (w,\sigma,c,\hat{K}_a\varphi) \mid \ \begin{array}{l} \text{there exists } (w',\sigma',c') \ \text{ s.t. } \ (w,\sigma,c) R_a (w',\sigma',c'), \\ (w',\sigma',c',\checkmark) \in \Gamma \ \text{ and } \ (w',\sigma',c',\varphi) \in \Gamma \end{array} \right\} \\
&\cup \big\{ (w,\epsilon,\mathbf{0},\checkmark) \mid \ w \in W \ \big\} \\
&\cup \big\{ (w,\sigma,c,\checkmark) \mid \ \text{ there is } c' < c \ \text{ s.t. } \ (w,\sigma,c',\checkmark) \in \Gamma \ \big\} \\
&\cup \left\{ (w,\sigma,c,\checkmark) \mid \ \begin{array}{l} (w,\sigma',c,\checkmark) \in \Gamma \ \text{ and } \ (w,\sigma',c,\psi) \in \Gamma, \\ \text{where } \sigma = \sigma' :: \psi \end{array} \right\} \\
&\cup \big\{ (w,\sigma,c,\bigcirc_a\varphi) \mid \ c(i) < |\sigma| \ \text{ and } \ (w,\sigma,c^{+a},\varphi) \in \Gamma \ \big\} \\
&\cup \big\{ (w,\sigma,c,\langle\psi\rangle\varphi) \mid \ \sigma{::}\psi \in \mathsf{Seq}(\mathcal{A}), (w,\sigma,c,\psi) \in \Gamma \ \text{ and } \ (w,\sigma{::}\psi,c,\varphi) \in \Gamma \ \big\}
\end{aligned}
$$

Figure 4: Function f that applies one step of the truth conditions

Remark 2. *The Knaster-Tarski theorem is often used to define the denotational semantics of programming languages [Winskel, 1993] in the same spirit as what we do here to define consistency.*

5. Semantic properties

In this section we establish some semantic properties of our logic. First we compare it with PAL, explaining why ABL is not a conservative extension of PAL when we have at least two agents. Then we establish some validities of ABL that show how the correctly defined semantics captures the intuitions we have about asynchrony.

For the rest of the section, we assume that we have a class of initial models and a class of announcement protocols for which the circularity problem can be solved and the semantics defined as in Figure 3 (for example, arbitrary initial models and positive announcements), and we discuss some validities of our logic.

5.1. Difference from PAL

We discuss the difference between the semantics of our logic and those of PAL. In PAL, every time an announcement is made, the Kripke model is updated by removing possible worlds where the announcement is not true (see Definition 3). This amounts to using the new information to delete epistemic alternatives that are no longer considered possible: since announcements are true, a world where an announcement is not true is not a possible world. In our case, epistemic alternatives cannot be deleted at the time of the announcement, since announcements are not received immediately by the agents, and in general agents can even have an unbounded number of pending announcements to read. Instead, this pruning is performed directly in the semantics of the knowledge operator, by eliminating all possible states that are not consistent: the pruning is not performed at the moment of the announcement, but is delayed until a knowledge operator is evaluated. Thus, the update operation in PAL and the consistency check in our logic play the same role. This is also reflected in the circularity problem, which stems from a mutual dependence between the definition of the semantics and, in the case of PAL, that of the update, and in the case of our logic, that of consistency.

We also note that if there are at least two agents, our logic is *not* a conservative extension of PAL. An intuitive way of seeing this is that in PAL, an announcement immediately becomes common knowledge, while in our setting asynchrony makes common knowledge unachievable. One may be tempted to define a translation tr from PAL to ABL, where all cases of the inductive definition are trivial, except that of the announcement operator which is

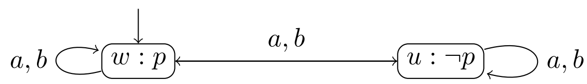
- $tr(\langle\psi\rangle_{\mathrm{PAL}}\varphi) := \langle tr(\psi)\rangle \bigcirc_{a_1} \cdots \bigcirc_{a_n} tr(\varphi)$, where $Ag = \{a_1, \ldots, a_n\}$.

In the single-agent case, the translation tr yields a conservative extension of PAL, but it is not the case when there are at least two agents. One may think that "after the *synchronous* public announcement of ψ, φ holds" is the same thing as "after the *asynchronous* broadcast of ψ and its reception by all agents, φ holds". But we show that this is not the case.

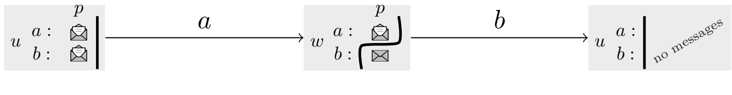
In PAL, if p is announced, then p immediately becomes common knowledge, which we recall, means that all agents know p, they all know that they all know p, they all know that they all know that they all know p, and so on. On the other hand, in asynchronous systems common knowledge cannot be reached [Halpern and Moses, 1990; Moses and Tuttle, 1988], and our logic illustrates this phenomenon. Even finite approximations of common knowledge fail to hold in our logic. For instance, while $[p]_{\mathrm{PAL}}K_aK_bp$ is a validity of PAL, its translation $[p] \bigcirc_a \bigcirc_b K_aK_bp$ is not valid.

Proposition 1. *There exist $\mathcal{M}$, an announcement protocol $\mathcal{A}$ and a consistent state $S \in \mathcal{M} \otimes \mathcal{A}$ such that $\mathcal{M} \otimes \mathcal{A}, S \not\models [p] \bigcirc_a \bigcirc_b K_aK_bp$.*

Proof. The idea is the following: after announcing p, and after all agents have received the message p, a does not know whether agent b has received p or not. Therefore, a does not know that b knows p. Let us consider the following initial model $\mathcal{M}$:



The actual world is w. Since p holds in w, it can be announced. Let $\mathcal{A} = \{\{p\}\}$. We prove that $\mathcal{M} \otimes \mathcal{A}, (w, \epsilon, \mathbf{0}) \not\models [p] \bigcirc_a \bigcirc_b K_aK_bp$. To see this, observe that after p has been announced and received by agent a and agent b (*i.e.*, after evaluation of the first three operators of the formula), we reach state $S = (w, [p], \begin{smallmatrix} a & \mapsto & 1 \\ b & \mapsto & 1 \end{smallmatrix})$. But in $\mathcal{M} \otimes \mathcal{A}$, we have (we only represent a part of $\mathcal{M} \otimes \mathcal{A}$, which is infinite):



Indeed in state $S = (w, [p], \begin{smallmatrix} a & \mapsto & 1 \\ b & \mapsto & 1 \end{smallmatrix})$ agent a considers it possible that agent b did not receive announcement p, and thus she considers state $S' = (w, [p], \begin{smallmatrix} a & \mapsto & 1 \\ b & \mapsto & 0 \end{smallmatrix})$

possible. In S', because b received no announcement, and in the initial model we have $w \rightarrow_b u$, agent b considers it possible that the actual world is u and nothing has been announced, *i.e.* she considers state $S'' = (u, \epsilon, \mathbf{0})$ possible. Because p does not hold in S'', we have that $S \not\models K_a K_b p$, which concludes the proof. ∎

5.2. Validities

We say that a formula φ is *valid* if for every initial model $\mathcal{M}$ and every announcement protocol $\mathcal{A}$ in the classes considered, and for every consistent state $S \in \mathcal{M} \otimes \mathcal{A}$, we have $\mathcal{M} \otimes \mathcal{A}, S \models \varphi$. We write $\models \varphi$ to express that φ is valid. In the following proposition we establish some validities that provide insights into our framework and show how our definitions correctly capture some natural properties that intuitively should hold in the asynchronous framework we consider.

Proposition 2. *For every $\varphi \in$ ABL and propositional formula ψ, we have that:*

1. $\models \bigcirc_a \bigcirc_b \varphi \leftrightarrow \bigcirc_b \bigcirc_a \varphi$
2. $\models \bigcirc_a \top \rightarrow (\bigcirc_a \varphi \leftrightarrow \neg \bigcirc_a \neg\varphi)$
3. $\models \neg \bigcirc_a \top \rightarrow [\psi] \bigcirc_a K_a \psi$
4. $\models \neg \bigcirc_a \top \rightarrow [\psi] \bigcirc_a K_a(\neg \bigcirc_b \top \rightarrow K_b \psi)$

Proof. We prove the first validity and the other three are left to the reader.

Suppose that we have $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \bigcirc_a \bigcirc_b \varphi$. By Figure 3, this means that $c(1) < |\sigma|$ and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c^{+a}) \models \bigcirc_b \varphi$, and the latter implies that $c^{+a}(b) < |\sigma|$ and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, (c^{+a})^{+b}) \models \varphi$. Now, because $(c^{+a})^{+b} = \left(c^{+b}\right)^{+a}$, we obtain that $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c^{+b}) \models \bigcirc_a \varphi$, and therefore $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \bigcirc_b \bigcirc_a \varphi$. The proof for the other direction is symmetric. ∎

Let us comment on these validities. The first one says that it is possible to permute the order of agents that receive next announcements in their respective queues. The second one says that if an agent has an announcement to read, then reading it is a deterministic operation. The third one says that if an agent has no pending announcement and some propositional formula is announced, then after reading his next pending announcement, the agent will know that formula. Intuitively, this is because the truth value of a propositional formula does not change, and the agents know this. The last validity illustrates the fact that in our framework, the behaviour of the public channel is common knowledge. Indeed it says that if, in a situation where agent a has read all the announced messages, a propositional formula ψ is announced and agent a reads it, then agent a knows that if agent b has read all the announced messages (and in particular the last one, which is ψ), then agent b also knows ψ. In some sense, it means that initially agent a knows that agent b will receive the same messages as herself. In the last two validities we restrict to propositional formulas in order to avoid Moore's paradox [van Ditmarsch et al., 2007].

We also establish the following proposition, which says that if all the $\bigcirc_a$ operators in a formula φ are under the scope of a knowledge operator, then its truth value is left unchanged by the announcement of any formula ψ. Indeed, the knowledge operator considers all possibilities for the content of the agent's channel, so that the possibility that ψ is in the channel is considered, whether ψ was actually announced or not.

In the following, in addition to the assumption that models and announcement protocols are restricted to classes for which the semantics is defined, we consider announcement protocols in which each announcement can be made infinitely many times. We call such protocols *free protocols*.

Proposition 3. *Let φ be a formula in* ABL*, in which every $\bigcirc_a$ is under the scope of some K_b, and let $\mathcal{A}_\infty$ be a free protocol. For every initial model $\mathcal{M}$ and consistent state $S = (w, \sigma, c) \in \mathcal{M} \otimes \mathcal{A}_\infty$, for every $\psi \in \mathcal{A}_\infty$, we have $\mathcal{M} \otimes \mathcal{A}_\infty, S \models \langle\psi\rangle\varphi \leftrightarrow \psi \wedge \varphi$.*

This result follows immediately from the following lemma:

Lemma 1. *Let φ be a formula in* ABL*, in which every $\bigcirc_a$ is under the scope of some K_b, and let $\mathcal{A}_\infty$ be a free protocol. For every initial model $\mathcal{M}$ and for every consistent state $(w, \sigma, c) \in \mathcal{M} \otimes \mathcal{A}_\infty$, for every $\psi \in \mathcal{A}_\infty$ such that $(w, \sigma{::}\psi, c)$ is consistent, we have $\mathcal{M} \otimes \mathcal{A}_\infty, (w, \sigma{::}\psi, c) \models \varphi$ iff $\mathcal{M} \otimes \mathcal{A}_\infty, (w, \sigma, c) \models \varphi$.*

Proof. By induction on φ. The Boolean cases are trivial.

Case $\varphi = K_a\varphi'$: Since (w, σ, c) is a state, $c(a) \leq |\sigma|$. It is then easy to check that $\{S \mid (w, \sigma{::}\psi, c)R_aS\} = \{S \mid (w, \sigma, c)R_aS\}$, and the result follows.

Case $\varphi = \langle\psi'\rangle\varphi'$: If $\psi' \notin \mathcal{A}_\infty$, the formula φ trivially does not hold in both states. Otherwise, because ψ' has infinite multiplicity in $\mathcal{A}_\infty$, $\sigma{::}\psi' \in \mathsf{Seq}(\mathcal{A}_\infty)$. We therefore have $(w, \sigma{::}\psi, c) \models \langle\psi'\rangle\varphi'$ iff $(w, \sigma{::}\psi, c) \models \psi'$ and $(w, \sigma{::}\psi{::}\psi', c) \models \varphi'$.

Assume that $(w, \sigma{::}\psi, c) \models \langle\psi'\rangle\varphi'$, we prove that $(w, \sigma, c) \models \langle\psi'\rangle\varphi'$. We have that $(w, \sigma{::}\psi, c) \models \psi'$ (hence $(w, \sigma{::}\psi{::}\psi', c)$ is consistent) and $(w, \sigma{::}\psi{::}\psi', c) \models \varphi'$. Because ψ' is a subformula of φ, each $\bigcirc_a$ in it is in the scope of some K_b; we can thus apply the induction hypothesis for $(w, \sigma{::}\psi, c) \models \psi'$, obtaining that $(w, \sigma, c) \models \psi'$. By induction hypothesis on $(w, \sigma{::}\psi{::}\psi', c) \models \varphi'$, we get first that $(w, \sigma{::}\psi, c) \models \varphi'$, then $(w, \sigma, c) \models \varphi'$ and finally $(w, \sigma{::}\psi', c) \models \varphi'$ (observe that $(w, \sigma{::}\psi', c)$ is consistent since $(w, \sigma, c) \models \psi'$). We have proved that $(w, \sigma, c) \models \langle\psi'\rangle\varphi'$.

The other direction is treated the same way.

Finally, the case $\varphi = \bigcirc_a\varphi'$ is not possible as $\bigcirc_a$ is not under the scope of any K_b. ∎

6. Model checking

Here we address the model checking problem when $\mathcal{A}$ is a finite multiset, that is, when the support set of $\mathcal{A}$ is finite and the multiplicity of each element is an integer. More precisely, we consider the following decision problem:

- input: an initial pointed model $(\mathcal{M}, w)$, a *finite* multiset of formulas $\mathcal{A}$ (where multiplicities are written in *unary*), a formula φ_0;
- output: yes if $\mathcal{M} \otimes \mathcal{A}, (w, \epsilon, \mathbf{0}) \models \varphi_0$, no otherwise.

In practice, model checking is used to check a scenario described by $\mathcal{A}$ and φ_0 from a given initial situation $(\mathcal{M}, w)$.

6.1. Propositional announcements

In this section, we suppose that formulas in $\mathcal{A}$ are propositional. Note that in this case (which is a particular case of existential announcements) the circularity problem does not exist, as consistency of a state (w, σ, c) can be trivially checked by verifying that all *propositional* formulas in σ hold in world w of the initial model, according to the classic semantics of propositional logic.

We consider the model checking problem for ABL where inputs $(\mathcal{M}, w, \mathcal{A}, \varphi_0)$ are such that $\mathcal{A}$ only contains propositional formulas. We call this problem the *model checking problem for propositional protocols.*

Theorem 2. *The model checking problem for propositional protocols is in* PSPACE.

Proof. Figure 5 presents an algorithm that takes a pointed model $(\mathcal{M}, w)$, a finite multiset $\mathcal{A}$, a sequence $\sigma \in \mathsf{Seq}(\mathcal{A})$, a cut c on σ and a formula φ as an input. To check the consistency of a state (w, σ, c), we call $\mathsf{checkconsistency}(\mathcal{M}, \mathcal{A}, w, \sigma, c)$ which verifies that every (propositional) formula ψ occurring in σ evaluates to true with the valuation $\Pi(w)$.

It is easily proven by induction that, for all ψ, the following property $P(\varphi)$ holds:

$\mathcal{M}, \mathcal{A}, (w, \sigma, c) \models \varphi$ iff $\mathsf{mc}(\mathcal{M}, \mathcal{A}, w, \sigma, c, \varphi)$ returns true.

This establishes the correctness of the algorithm. We now analyze its complexity.

First, observe that because $\mathcal{A}$ is finite and each element has finite multiplicity, we have that $\mathsf{Seq}(\mathcal{A})$ only contains sequences of length linear in $|\mathcal{A}|$ (recall that multiplicities are written in unary). It is therefore easy to see that the consistency check $(*\checkmark)$ is done in polynomial time in the size of the input and thus requires a polynomial amount of space. Now, the number of nested calls of mc is bounded by the size of the formula to check, and each call requires a polynomial amount of memory for storing local variables, so that the algorithm runs in polynomial space. ∎

Theorem 3. *The model-checking problem for propositional protocols is* PSPACE*-hard.*

Proof. See Appendix C. ∎

6.2. Finite tree initial model

In this section, we restrict the set of inputs $\mathcal{M}, \mathcal{A}, w, \varphi_0$ of the model checking problem to those where the initial pointed models $(\mathcal{M}, w)$ are finite trees rooted in w.

Theorem 4. *The model-checking problem when we restrict initial models to finite trees is in* PSPACE.

Proof. We consider the algorithm of Figure 5 again but now the consistency checking $(*\checkmark)$ consists of calling the following procedure:

```
function checkconsistency(M, A, w, σ, c)
  if c = 0
    return true
  else
    for c' < c do
      if mc(M, A, w, σ, c', ✓) then
        return true
    return mc(M, A, w, σ', c, ✓) and mc(M, A, w, σ', c, ψ)    where σ = σ'::ψ
```

```
function mc(M, A, w, σ, c, φ)
  match φ do
    case p: return p ∈ V(w);
    case ✓: return checkconsistency(M, A, w, σ, c)                          (*✓)
    case ¬ψ: return not mc(M, A, w, σ, c, ψ);
    case (ψ1 ∧ ψ2): return mc(M, A, w, σ, c, ψ1) and mc(M, A, w, σ, c, ψ2);
    case K_a ψ :
      for (u, σ', c') such that w →_a u, σ' ∈ Seq(A) and c' is a cut on σ' do
        if σ'[1..c(a)] = σ[1..c'(a)] and mc(M, A, u, σ', c', ✓) then
          if not mc(M, A, u, σ', c', ψ) then
            return false
      return true
    case ⟨ψ⟩χ :
      if σ::ψ ∈ Seq(A) and mc(M, A, w, σ, c, ψ) then
        return mc(M, A, w, σ::ψ, c, χ);
      else
        return false;
    case ◯_a ψ: return c(a) < |σ| and mc(M, A, w, σ, c^{+a}, ψ)
```

Figure 5: Model checking algorithm

Soundness and completeness are proven by induction on inputs using the order $\prec$ defined in Section 4.1.

Concerning the complexity, the argument given in the proof of Theorem 2 no longer holds. In order to bound the number of nested calls of mc, we have to remark that from a call of mc to a sub-call of mc:

(1) either we change the current world w in the initial model for a successor u in the finite tree;
(2) or the quantity $|\sigma| + |\varphi| + \sum_{a \in Ag} c(a)$ is strictly decreasing, where $|\varphi|$ is the length of φ and if $\sigma = [\varphi_1, \ldots, \varphi_k]$ then $|\sigma| = \sum_{i=1}^{k} |\varphi_i|$.

Now, the number of times (1) occurs is bounded by the depth $depth(\mathcal{M}, w)$ of the finite tree $\mathcal{M}, w$. As each φ is either a subformula of the input formula φ_0 or a subformula of a formula in $\mathcal{A}$, $|\varphi| \leq |\varphi_0| + |\mathcal{A}|$ where $|\mathcal{A}| := \sum_{\psi \in \mathcal{A}} |\psi|$, and where each single formula ψ is counted as many times as it occurs in the multiset $\mathcal{A}$. Furthermore, $|\sigma| \leq |\mathcal{A}|$ and $c(a) \leq |\mathcal{A}|$. Thus, the quantity $|\sigma|+|\varphi|+\sum_{a \in Ag} c(a)$ is bounded by $(|Ag|+2)|\mathcal{A}|+|\varphi_0|$. Therefore, the number of nested calls to mc is bounded by $depth(\mathcal{M}, w) \times ((|Ag|+2)|\mathcal{A}| + |\varphi_0|)$. So the algorithm requires a polynomial amount of memory in the size of the input (recall that the multiplicity of $\mathcal{A}$ is encoded in unary). ∎

6.3. Existential announcements

In this subsection, we design an exponential-time algorithm for the model checking problem in the case of existential announcements.

Given an input $\mathcal{M}, \mathcal{A}, w, \varphi_0$, the algorithm first computes the least fixed point Γ^* of the function f defined in Section 4.2. Because the number of possible sequences in $\mathsf{Seq}(\mathcal{A})$ is exponential in $|\mathcal{A}|$, the set B of pairs (S, φ) where $S \in \mathcal{M} \otimes \mathcal{A}$ and $\varphi \in \mathcal{A} \cup \{\checkmark\}$ is of size exponential in the size of the input, and therefore computing the fixed point requires exponential time in the size of the input. This gives us the semantics of consistency for states of $\mathcal{M} \otimes \mathcal{A}$.

Then, to evaluate φ_0, we use the procedure mc of Figure 5 where line $(*\checkmark)$, which checks the consistency of a state (w, σ, c), is replaced by checking whether $(w, \sigma, c, \checkmark) \in \Gamma^*$. The algorithm mc also requires exponential time. To sum up:

Theorem 5. *The model-checking problem for existential announcements is in* EXPTIME.

7. Satisfiability for propositional announcements

In this section, we address the satisfiability problem when $\mathcal{A}$ is a finite multiset of propositional formulas, that is, when the support set of $\mathcal{A}$ is finite, the multiplicity of each element is an integer and formulas in $\mathcal{A}$ are propositional. More precisely, we say that a formula φ_0 is $\mathcal{A}$-satisfiable if there exists an initial pointed model $(\mathcal{M}, w)$ such that $\mathcal{M} \otimes \mathcal{A}, (w, \epsilon, \mathbf{0}) \models \varphi_0$. We consider the following decision problem:

- input: a *finite* multiset of propositional formulas $\mathcal{A}$ (where multiplicities are written in *unary*), a formula φ_0;
- output: yes if φ_0 is $\mathcal{A}$-satisfiable, no otherwise.

In practice, a typical application of the satisfiability problem would be to check that a class of systems described by a formula φ satisfies a property ψ. To do so, one checks whether $\varphi \wedge \neg\psi$ is satisfiable. If it is not, then indeed all φ-systems satisfy ψ. If it is satisfiable, then the algorithm we present here (like all tableau methods) produces a counter-example, *i.e.*, a model $(\mathcal{M}, w)$ such that $\mathcal{M} \otimes \mathcal{A}, (w, \epsilon, \mathbf{0}) \models \varphi \wedge \neg\psi$, or in other words, a φ-system that does not satisfy ψ.

7.1. Tableau method description

Our tableau method manipulates terms that we call *tableau terms*, which are of the following kind:

- $(w \;\; \sigma \;\; c \;\; \varphi)$: w is a world symbol that represents a world of the model $\mathcal{M}$ being constructed, σ is a sequence of formulas in $\mathsf{Seq}(\mathcal{A})$, c is a cut for σ and φ is a sub-formula of φ_0 that should be true in $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c)$.
- $(w \rightarrow_a u)$: w and u are two world symbols such that $w \rightarrow_a u$ in the model $\mathcal{M}$ being constructed.
- $\bot$: Denotes an inconsistency.

A *tableau rule* is represented by a *numerator* $\mathcal{N}$ above a line and a finite list of *denominators* $\mathcal{D}_1, \ldots, \mathcal{D}_k$ below this line, separated by vertical bars, representing non-deterministic choice:

$$\frac{\mathcal{N}}{\mathcal{D}_1 \mid \ldots \mid \mathcal{D}_k}$$

The numerator and the denominators are finite sets of tableau terms.

A *tableau for* input $(\mathcal{A}, \varphi_0)$ is a finite tree with a set of tableau terms at each node, whose root is:

$$\Gamma_0 = \{(\mathrm{w}_0 \ \ \epsilon \ \ \mathbf{0} \ \ \varphi_0)\}.$$

A rule with numerator $\mathcal{N}$ is *applicable* to a node carrying a set Γ if Γ contains an instance of $\mathcal{N}$ for which the rule has not yet been applied. If no rule is applicable, Γ is said to be *saturated*. We call a node n an *end node* if the set of tableau terms Γ it carries is saturated, or if $\bot \in \Gamma$. The tableau is extended the following way:

1. Choose a leaf node n carrying Γ where n is not an end node, and choose a rule applicable to n.
2. For each denominator $\mathcal{D}_i$ of the rule, create one successor node for n carrying the union of Γ with an appropriate instanciation of $\mathcal{D}_i$.

A branch in a tableau is a path from the root to an end node. A branch is *closed* if its end node contains $\bot$, otherwise it is *open*. A tableau is *closed* if all its branches are closed, otherwise it is *open*. A pair $(\mathcal{A}, \varphi_0)$ is said to be *consistent* if no tableau for $(\mathcal{A}, \varphi_0)$ is closed.

The tableau rules are described in Figure 6, in which we write $(\sigma, c) \sim_a (\sigma', c')$ for $\sigma|_{c(a)} = \sigma'|_{c'(a)}$.

Remark 3. *Rule ch for choosing valuations is necessary for checking consistency of states in rules $K_a\varphi$ and ✓. For this reason, rule ch is always applied in priority before rules $K_a\varphi$ and ✓. In a node carrying Γ and saturated for rule ch, if* w *is a world symbol in Γ, we say that* σ is true in w *if the valuation ν, defined by $\nu(p) = 1$ if* $(\mathrm{w} \ \epsilon \ \mathbf{0} \ p) \in \Gamma$ *and $\nu(p) = 0$ if* $(\mathrm{w} \ \epsilon \ \mathbf{0} \ \neg p) \in \Gamma$, *satisfies every formula in σ (recall that in this section announcements are propositional).*

7.2. Tableau method soundness and completeness

In this section we prove that the tableau method is sound and complete. Note that we will establish that every tableau is finite in the complexity analysis of the tableau method (see Theorem 6).

Proposition 4. *If $(\mathcal{A}, \varphi_0)$ is consistent, then φ_0 is $\mathcal{A}$-satisfiable.*

Proof. Suppose that $(\mathcal{A}, \varphi_0)$ is consistent, and consider a tableau t for $(\mathcal{A}, \varphi_0)$. By assumption, this tableau is open, which means that it has an open branch. Consider one such open branch, and let Γ be the set of tableau terms carried by its end node. We define the model $\mathcal{M} = (W, \{\rightarrow_a\}_{a \in Ag}, \Pi)$, where

- $W = \{\mathrm{w} \mid (\mathrm{w} \ \ \sigma \ \ c \ \ \varphi) \in \Gamma$ for some σ, c and $\varphi\}$,
- $\rightarrow_a = \{(w, u) \mid (\mathrm{w} \ \rightarrow_a \ \mathrm{u}) \in \Gamma\}$ and
- for each $w \in W$, $\Pi(\mathrm{w}) = \{p \mid (\mathrm{w} \ \epsilon \ \mathbf{0} \ p) \in \Gamma\}$.

We prove that for all $(\mathrm{w} \ \ \sigma \ \ c \ \ \varphi) \in \Gamma$, it holds that $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \varphi$. Because $(\mathrm{w}_0 \ \epsilon \ \mathbf{0} \ \varphi_0)$ is in $\Gamma_0 \subseteq \Gamma$, it follows that φ_0 is $\mathcal{A}$-satisfiable.

If $\varphi = p$, by saturation of rule p we have $(\mathrm{w} \ \ \epsilon \ \ \mathbf{0} \ \ p) \in \Gamma$, thus $p \in \Pi(\mathrm{w})$ by construction of $\mathcal{M}$, and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models p$.

If $\varphi = \neg p$, by saturation of rule $\neg p$ we have $(\mathrm{w} \ \ \epsilon \ \ \mathbf{0} \ \ \neg p) \in \Gamma$. We cannot have $(\mathrm{w} \ \epsilon \ \mathbf{0} \ p) \in \Gamma$, otherwise the branch would be closed by saturation of rule $\bot$. Therefore $p \notin \Pi(\mathrm{w})$, and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \neg p$.

$$\frac{(w\ \ \sigma\ \ c\ \ \varphi)}{(w\ \ \epsilon\ \ \mathbf{0}\ \ p) \mid (w\ \ \epsilon\ \ \mathbf{0}\ \ \neg p)}\ ch \quad \text{for all atomic propositions } p \text{ appearing in } \varphi_0 \text{ and } \mathcal{A}$$

$$\frac{(w\ \ \sigma\ \ c\ \ (\varphi \wedge \psi))}{(w\ \ \sigma\ \ c\ \ \varphi)\ (w\ \ \sigma\ \ c\ \ \psi)}\ \wedge \qquad \frac{(w\ \ \sigma\ \ c\ \ \neg(\varphi \wedge \psi))}{(w\ \ \sigma\ \ c\ \ \neg\varphi) \mid (w\ \ \sigma\ \ c\ \ \neg\psi)}\ \neg\wedge$$

$$\frac{(w\ \ \sigma\ \ c\ \ p)}{(w\ \ \epsilon\ \ \mathbf{0}\ \ p)}\ \leftarrow p \qquad \frac{(w\ \ \sigma\ \ c\ \ \neg p)}{(w\ \ \epsilon\ \ \mathbf{0}\ \ \neg p)}\ \leftarrow\neg p \qquad \frac{(w\ \ \sigma\ \ c\ \ \neg\neg\varphi)}{(w\ \ \sigma\ \ c\ \ \varphi)}\ \neg\neg$$

where $p \in \mathcal{AP}$

$$\frac{(w\ \ \sigma\ \ c\ \ \bigcirc_a \varphi)}{(w\ \ \sigma\ \ c^{+a}\ \ \varphi)}\ \bigcirc_a \quad \text{if } c(a) < |\sigma| \qquad \frac{(w\ \ \sigma\ \ c\ \ \bigcirc_a \varphi)}{\bot}\ \bigcirc_a \quad \text{if } c(a) = |\sigma|$$

$$\frac{(w\ \ \sigma\ \ c\ \ \neg\bigcirc_a \varphi)}{(w\ \ \sigma\ \ c^{+a}\ \ \neg\varphi)}\ \neg\bigcirc_a \quad \text{if } c(a) < |\sigma|$$

$$\frac{(w\ \ \sigma\ \ c\ \ \langle\psi\rangle\varphi)}{(w\ \ \sigma\ \ c\ \ \psi)(w\ \ \sigma{::}\psi\ \ c\ \ \varphi)}\ \langle\psi\rangle \quad \text{if } \sigma{::}\psi \in \mathsf{Seq}(\mathcal{A}) \qquad \frac{(w\ \ \sigma\ \ c\ \ \langle\psi\rangle\varphi)}{\bot}\ \langle\psi\rangle \quad \text{if } \sigma{::}\psi \notin \mathsf{Seq}(\mathcal{A})$$

$$\frac{(w\ \ \sigma\ \ c\ \ \neg\langle\psi\rangle\varphi)}{(w\ \ \sigma\ \ c\ \ \neg\psi) \mid (w\ \ \sigma\ \ c\ \ \psi)(w\ \ \sigma :: \psi\ \ c\ \ \neg\varphi)}\ \neg\langle\psi\rangle \quad \text{if } \sigma{::}\psi \in \mathsf{Seq}(\mathcal{A})$$

$$\frac{(w\ \ \sigma\ \ c\ \ K_a\varphi)(w \rightarrow_a u)}{(u\ \ \sigma'\ \ c'\ \ \varphi)}\ K_a\varphi \quad \text{for all } (\sigma', c') \sim_a (\sigma, c) \text{ and } \sigma' \text{ true in } u \text{ (see Remark 3)}$$

$$\frac{(w\ \ \sigma\ \ c\ \ \neg K_a\varphi)}{(u\ \ \sigma'_1\ \ c'_1\ \ \neg\varphi)(w \rightarrow_a u) \mid \cdots \mid (u\ \ \sigma'_n\ \ c'_n\ \ \neg\varphi)(w \rightarrow_a u)}\ \neg K_a\varphi \quad \text{where } (\sigma'_i, c'_i) \sim_a (\sigma, c) \text{ and } u \text{ is fresh}$$

$$\frac{(w\ \ \sigma\ \ c\ \ \varphi)}{\bot}\ \checkmark \quad \text{if } \sigma \text{ is not true in } w \qquad \frac{(w\ \ \epsilon\ \ \mathbf{0}\ \ p)(w\ \ \epsilon\ \ \mathbf{0}\ \ \neg p)}{\bot}\ \bot \quad \text{for } p \in \mathcal{AP}$$

Figure 6: Tableau rules

For boolean connectives, the result follows by saturation of the appropriate tableau rule, plus application of the induction hypothesis.

If $\varphi = \bigcirc_a \varphi'$, we have that $c(a) < |\sigma|$, otherwise Γ would contain $\bot$ by saturation of rule $\bigcirc_a$ and the branch would be closed. Therefore, again by saturation of rule $\bigcirc_a$, Γ contains $(w\ \ \sigma\ \ c^{+a}\ \ \varphi')$. By induction hypothesis we get that $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c^{+a}) \models \varphi'$, and thus $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \bigcirc_a \varphi'$.

If $\varphi = \neg \bigcirc_a \varphi'$ we apply similar reasoning, except for the case $c(a) = |\sigma|$ in which φ trivially holds.

If $\varphi = \langle \psi \rangle \varphi'$, then $\sigma{::}\psi \in \mathsf{Seq}(\mathcal{A})$, otherwise the branch would be closed. It follows by saturation of rule $\langle \psi \rangle$ that $(w\ \ \sigma\ \ c\ \ \psi)$ and $(w\ \ \sigma{::}\psi\ \ c\ \ \varphi')$ are in Γ, and we conclude by applying the induction hypothesis.

If $\varphi = \neg\langle \psi \rangle \varphi'$, in case $\sigma{::}\psi$ is not in $\mathsf{Seq}(\mathcal{A})$, φ trivially holds. Otherwise, by saturation of rule $\neg\langle \psi \rangle$, either $(w\ \ \sigma\ \ c\ \ \neg\psi)$ is in Γ, or both $(w\ \ \sigma\ \ c\ \ \psi)$ and $(w\ \ \sigma{::}\psi\ \ c\ \ \neg\varphi')$ are in Γ. In both cases, we conclude by induction hypothesis.

If $\varphi = K_a \varphi'$, let (u, σ', c') be such that $(w, \sigma, c) R_a (u, \sigma', c')$ and $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \checkmark$, *i.e.* σ' is true in u (see Remark 3). We have that $w \rightarrow_a u$, so by construction of $\mathcal{M}$, $(w \rightarrow_a u) \in \Gamma$. Also, since $(w, \sigma, c) R_a (u, \sigma', c')$ we have that $(\sigma, c) \sim_a (\sigma', c')$. By saturation of rule $K_a \varphi$ we thus have that $(u\ \ \sigma'\ \ c'\ \ \varphi') \in \Gamma$, and by induction hypothesis $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \varphi'$, which concludes.

If $\varphi = \neg K_a \varphi'$, by saturation of rule $\neg K_a \varphi$ there exist $(\sigma', c') \sim_a (\sigma, c)$ and a world symbol u such that Γ contains $(u\ \ \sigma'\ \ c'\ \ \neg\varphi)$ and $(w \rightarrow_a u)$. It follows that $(w, \sigma, c) \rightarrow_a (u, \sigma', c')$. We also have that $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \checkmark$ (or in other words, σ' is true in u), otherwise the branch would be closed by rule $\checkmark$. Finally, by induction hypothesis, $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \neg\varphi'$, and thus $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \neg K_a \varphi'$. ∎

Proposition 5. *If φ_0 is $\mathcal{A}$-satisfiable, then $(\mathcal{A}, \varphi_0)$ is consistent.*

Proof. Suppose that there is a pointed model $(\mathcal{M}_0, w_0)$ such that $\mathcal{M}_0 \otimes \mathcal{A}, (w_0, \epsilon, \mathbf{0}) \models \varphi_0$. We must prove that every tableau for $(\mathcal{A}, \varphi_0)$ has an open branch.

We let W_Γ denote the set of world symbols appearing in a set of tableau terms Γ. Such a set Γ is said to be *interpretable* if, first, it does not contain $\bot$ and, second, there is an initial model $\mathcal{M} = (W, \{\rightarrow_a\}_{a \in Ag}, \Pi)$ and a mapping $f : W_\Gamma \rightarrow W$ such that:

- for each $(w \rightarrow_a u) \in \Gamma$, $f(w) \rightarrow_a f(u)$ and
- for each $(w\ \ \sigma\ \ c\ \ \varphi) \in \Gamma$, $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \checkmark$ and $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \varphi$.

We write $\mathcal{M}, f \models \Gamma$ if these two conditions are met.

Observe that $\Gamma_0 = \{(w_0\ \ \epsilon\ \ \mathbf{0}\ \ \varphi_0)\}$ does not contain $\bot$, and by assumption there is a pointed model $(\mathcal{M}_0, w_0)$ such that $\mathcal{M}_0 \otimes \mathcal{A}, (w_0, \epsilon, \mathbf{0}) \models \varphi_0$. So $\mathcal{M}_0, [w_0 \mapsto w_0] \models \Gamma_0$, and Γ_0 is interpretable. We now prove that when a tableau rule is applied in a node that carries an interpretable set of tableau terms and is not an end node, then one of its successors carries an interpretable set. This implies that every tableau for $(\mathcal{A}, \varphi_0)$ has a branch whose end node carries an interpretable set; in particular, this set does not contain $\bot$, so the branch is open, which concludes.

In the following, Γ is the interpretable set of tableau terms in which the rule is applied, and $\mathcal{M} = (W, \{\rightarrow_a\}_{a \in Ag}, \Pi)$ and $f : W_\Gamma \rightarrow W$ are such that $\mathcal{M}, f \models \Gamma$.

We do not treat the case of rules for propositional logic as it is straightforward.

Rule ch for atomic proposition p, on numerator $\{(w \ \ \sigma \ \ c \ \ \varphi)\}$: If $p \in \Pi(f(w))$, then $\mathcal{M} \otimes \mathcal{A}, (f(w), \epsilon, \mathbf{0}) \models p$, and thus $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \epsilon \ \ \mathbf{0} \ \ p)\}$; otherwise $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \epsilon \ \ \mathbf{0} \ \ \neg p)\}$. So one of the successors is interpretable.

Rule $\bigcirc_a\varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ \bigcirc_a \varphi)\}$: by assumption, $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \bigcirc_a\varphi$. Thus, according to the semantics, we necessarily have that $c(a) < |\sigma|$. So the only successor in the tableau carries the set $\Gamma \cup \{(w \ \ \sigma \ \ c^{+a} \ \ \varphi)\}$. Since $(w \ \ \sigma \ \ c \ \ \bigcirc_a \varphi) \in \Gamma$ and $\mathcal{M}, f \models \Gamma$, $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \checkmark$, and thus also $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c^{+a}) \models \checkmark$. Besides, because $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \bigcirc_a\varphi$, we have that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c^{+a}) \models \varphi$. It follows that $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \sigma \ \ c^{+a} \ \ \varphi)\}$, and the successor is interpretable.

Rule $\neg \bigcirc_a \varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ \neg \bigcirc_a \varphi)\}$: the application of this rule requires that $c(a) < |\sigma|$ hold. So the fact that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \neg \bigcirc_a \varphi$ holds implies $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c^{+a}) \models \neg\varphi$. The consistency aspect is treated like for rule $\bigcirc_a\varphi$, and we obtain that $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \sigma \ \ c^{+a} \ \ \neg\varphi)\}$, hence the successor is interpretable.

Rule $\langle\psi\rangle\varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ \langle\psi\rangle\varphi)\}$: We have that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \langle\psi\rangle\varphi$, so $\sigma{::}\psi \in \mathsf{Seq}(\mathcal{A})$, which implies that it is the first version of the rule that is applied. We also have that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \psi$ and $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma{::}\psi, c) \models \varphi$. From the former and the fact that $\mathcal{M}, f \models \Gamma \ni (w \ \ \sigma \ \ c \ \ \langle\psi\rangle\varphi)$, we obtain that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma{::}\psi, c) \models \checkmark$. It follows that $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \sigma \ \ c \ \ \psi), (w \ \ \sigma{::}\psi \ \ c \ \ \varphi)\}$, and thus the only possible successor is interpretable.

Rule $\neg\langle\psi\rangle\varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ \neg\langle\psi\rangle\varphi)\}$: First, because $\{(w \ \ \sigma \ \ c \ \ \neg\langle\psi\rangle\varphi)\} \in \Gamma$, we have $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \checkmark$. Also, the application of this rule requires that $\sigma{::}\psi \in \mathsf{Seq}(\mathcal{A})$. So the fact that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \neg\langle\psi\rangle\varphi$ holds implies that either $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \neg\psi$ or $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma{::}\psi, c) \models \neg\varphi$. If $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \neg\psi$, we obtain that $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \sigma \ \ c \ \ \neg\psi)\}$, and the first successor is interpretable. Otherwise we have both $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma{::}\psi, c) \models \neg\varphi$ and $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \psi$. The latter implies that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma{::}\psi, c) \models \checkmark$; we obtain that $\mathcal{M}, f \models \Gamma \cup \{(w \ \ \sigma{::}\psi \ \ c \ \ \neg\varphi)\}$, and the second successor is interpretable.

Rule $K_a\varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ K_a\varphi), (w \to_a u)\}$, for some $(\sigma', c') \sim_a (\sigma, c)$ and σ' true in u: First, since rule ch has the priority over rule $K_a\varphi$, we know that Γ is saturated for rule ch. Also, since $(w \to_a u) \in \Gamma$ and tableau terms of this form can only be introduced by rule $\neg K_a\varphi$ together with a tableau term of the form $(u \ \ \sigma'' \ \ c'' \ \ \varphi')$, then there is one such tableau term in Γ. By saturation of rule ch, it follows that for each p appearing in φ_0 and $\mathcal{A}$, either $(u \ \ \epsilon \ \ \mathbf{0} \ \ p)$ or $(u \ \ \epsilon \ \ \mathbf{0} \ \ \neg p)$ is in Γ. This defines a valuation ν for u that, by assumption, makes σ' true (see Remark 3). Because $\mathcal{M}, f \models \Gamma$, we have that $\Pi(f(u))$ agrees with ν on all atomic propositions in $\mathcal{A}$. Since by assumption ν satisfies all formulas in σ', so does $\Pi(f(u))$, and therefore $\mathcal{M} \otimes \mathcal{A}, (f(u), \sigma', c') \models \checkmark$. Now, since $\mathcal{M}, f \models \Gamma$ and $(w \to_a u) \in \Gamma$, we have that $f(w) \to_a f(u)$, and because $(w \ \ \sigma \ \ c \ \ K_a\varphi) \in \Gamma$, it holds that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models K_a\varphi$. Since $f(w) \to_a f(u)$ and $(\sigma, c) \sim_a (\sigma', c')$, we have that $(f(w), \sigma, c) R_a (f(u), \sigma', c')$. As we have seen that $\mathcal{M} \otimes \mathcal{A}, (f(u), \sigma', c') \models \checkmark$, we finally have that $\mathcal{M} \otimes \mathcal{A}, (f(u), \sigma', c') \models \varphi$, thus $\mathcal{M}, f \models \Gamma \cup \{(u \ \ \sigma' \ \ c' \ \ \varphi)\}$, and the successor is interpretable.

Rule $\neg K_a\varphi$ on numerator $\{(w \ \ \sigma \ \ c \ \ \neg K_a\varphi)\}$: since $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \neg K_a\varphi$, there exist $u \in W$, σ' and c' such that $(f(w), \sigma, c) R_a (u, \sigma', c')$, $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \checkmark$ and $\mathcal{M} \otimes \mathcal{A}, (u, \sigma', c') \models \neg\varphi$. Recall that $(f(w), \sigma, c) R_a (u, \sigma', c')$ means that $f(w) \to_a u$ and $(\sigma, c) \sim_a (\sigma', c')$. Clearly, $\mathcal{M}, f[u \mapsto u] \models \{(u \ \ \sigma' \ \ c' \ \ \neg\varphi)(w \to_a u)\}$, and because u is fresh, $f[u \mapsto u]$ coincides with f on all world symbols appearing in Γ, so that $\mathcal{M}, f[u \mapsto u] \models \Gamma$. Finally, the denominator corresponding to σ', c' is interpretable (there are only finitely many possible σ' and c', see proof of Theorem 6).

Rule $\checkmark$ on numerator $\{(w \;\; \sigma \;\; c \;\; \varphi)\}$: because Γ is interpretable, this rule cannot be applied. Indeed, assume it is applied. Because rule ch is applied in priority, Γ is saturated for rule ch. With reasoning similar to that followed for rule $K_a\varphi$, we obtain that the valuation ν defined by Γ for w coincides with $\Pi(f(w))$ on all atomic propositions appearing in φ_0 and $\mathcal{A}$, and thus they agree on all formulas in σ. Yet on the one hand, since $(w \;\; \sigma \;\; c \;\; \varphi) \in \Gamma$ and $\mathcal{M}, f \models \Gamma$, we have that $\mathcal{M} \otimes \mathcal{A}, (f(w), \sigma, c) \models \checkmark$ and thus $\Pi(f(w))$ satisfies all formulas in σ. On the other hand, because the rule $\checkmark$ is applied, ν does not satisfy all formulas in σ, and we have a contradiction.

Rule $\perp$ on numerator $\{(w \;\; \epsilon \;\; \mathbf{0} \;\; p), (w \;\; \epsilon \;\; \mathbf{0} \;\; \neg p)\}$: because $\mathcal{M}, f \models \Gamma$ this cannot happen, as otherwise we would have both $p \in \Pi(f(w))$ and $p \notin \Pi(f(w))$. ∎

Theorem 6. *The satisfiability problem for finite propositional protocols is in* NEXPTIME.

Proof. Let $\mathcal{A}$ be a propositional and finite protocol and φ_0 be the formula to check. The algorithm to check whether φ_0 is $\mathcal{A}$-satisfiable consists of non-deterministically applying tableau rules of Figure 6 from the initial tableau $\{(w \;\; \epsilon \;\; \mathbf{0} \;\; \varphi_0)\}$.

Each world symbol w except w_0 is created by rule $\neg K_a$ with a formula φ_w, and the number of times this rule is applied to terms with w as world symbol is linear in φ_w. These world symbols can be ordered in a tree structure (a world symbol created by applying rule $\neg K_a$ in a tableau term with world symbol w is a child of w), and the modal depth of φ_w formulas is strictly decreasing in the tree. So the number of created world symbols w is exponential in the size of φ_0.

In addition, recall that the number of possible sequences of announcements σ is exponential in the size of $\mathcal{A}$, and the number of possible cuts c is $|\mathcal{A}|^{|Ag|}$. Therefore, the number of different tableau terms $(w \; \sigma \; c \; \psi)$ is exponential in $|\varphi_0| + |\mathcal{A}|$.

At each step, the algorithm is executing a rule that adds at least one term. As the number of terms is exponential, the number of rule applications is exponential, and thus the running time of the (non-deterministic) algorithm is exponential. So the satisfiability problem when the protocol is finite and propositional is in NEXPTIME. ∎

We now establish the matching lower bound.

Theorem 7. *The satisfiability problem for finite propositional protocols with at least two agents is* NEXPTIME*-hard.*

Proof. See Appendix D. ∎

8. Related work

We review several research areas related to different aspects of the present work.

8.1. Existing logics for asynchrony

As far as we know, there has not been much work on the relationship between knowledge, announcements and asynchrony. In [Dégremont et al., 2011], asynchrony in dynamic epistemic logic is studied, with the notion of asynchrony being that an agent cannot tell whether an event has occurred if her epistemic state is unchanged. This notion of asynchrony is different from the one we consider in this work: indeed in Dégremont et al. [2011], different agents can have a different idea of how many events have occurred so far, because some events might be completely unnoticed by some agents. So in one setting asynchrony is due to events being completely unobserved, while in the other (the one considered in this work) it is due to a delay between the occurrence of an event (the announcement) and its observation (the reception).

A logic dealing with knowledge and asynchrony is also developed in [Panangaden and Taylor, 1992], but in this setting, messages do not have logical content: for example, the logic does not allow for announcements about knowledge or about the effect of other announcements. [Fagin et al., 1992] is concerned with knowledge in multi-agent, dynamic systems which may be asynchronous, but does not explicitly model communication, and in particular the effects of asynchronous sending and receiving of true announcements, which is the focus of our work.

Recently, van Ditmarsch developed a logic of asynchronous announcements in [van Ditmarsch, 2017]. The major difference between our framework and that one is our third basic principle, that agents are able to imagine all possible pending messages. In van Ditmarsch's work, agents in fact do not consider any pending or future announcements possible; they only consider a message possible after they have received it. So in our work, an agent a has three sources of uncertainty: first, uncertainty about the state arising from the underlying Kripke model; second "past uncertainty," that is, uncertainty about which of the messages that a has received have already been received by other agents; and third, "future uncertainty," uncertainty about what messages are pending in the channel but unread by a, or which messages may be broadcast in the future. In van Ditmarsch's work, agents only have the first two sources of uncertainty: uncertainty arising from the underlying Kripke model, and "past uncertainty." This means that agents may not consider the current state possible, and may even have false knowledge. For example, if agents a and b initially do not know true proposition p, and then p is broadcast, in van Ditmarsch's framework, if a has received broadcast p and b has not, b considers it impossible that a knows p, even though a does indeed know p. Symbolically, $K_a p \wedge K_b \neg K_a p$. In our logic this is not the case: even when b has not received the broadcast of p, b considers it possible that p has been broadcast and received by a, so the formula $K_a p \wedge K_b \neg K_a p$ can never hold in our models. In general, $K_a \varphi \longrightarrow \varphi$ in our logic, while this is not the case in van Ditmarsch's logic.

8.2. Semi-private announcements and dynamic epistemic logic

On first glance, asynchronous broadcast logic has some similarities with *semi-private announcement logic*, [Gerbrandy and Groeneveld, 1997; Baltag et al., 1998; Baltag and Moss, 2004; Baltag et al., 2008]. Logics with semi-private announcements follow the same basic idea as public announcement logic, but rather than announcements being received by the entire group of agents, each message is announced to a subset of agents, while the rest of the agents know the message was announced to that group, but do not know what the content of the message was. In the general setting, group A receives message m and updates their knowledge accordingly, while the agents not in group A know that A received either m or its negation, $\neg m$, and update their knowledge accordingly. The identity of the group receiving each message is common knowledge for everyone. On the surface, this logic has some similarities with asynchronous broadcast logic: at any time, a certain group of agents has received each message, while others have not. However, like other variants of public announcement logic, logics of semi-private announcements are synchronous: a message is sent and received simultaneously, and thus common knowledge is achieved immediately by the group of agents receiving the message. Furthermore, even the group of agents who do not receive the message have synchronous information, since they immediately know that the other agents have received some message. Overall, in this setting, the agents have less uncertainty about one another's knowledge than in the asynchronous setting. In fact, the issues of semi-private messages and asynchrony are orthogonal; one could imagine an asynchronous logic of semi-private announcements, where each member of group A eventually receives announcement m, and the rest of the agents eventually receive the information that group A has been asynchronously sent either m or $\neg m$.

8.3. Arbitrary public announcement logic

Arbitrary public announcement logic (APAL) [Balbiani et al., 2007] has some similarities to our approach. In this logic, one can ask whether some formula holds after *any possible announcement*; this is not possible in our logic, but because agents can imagine pending messages, our knowledge operator considers any possible future sequence of announcements that follows the protocol, which is a related idea. Interestingly, the satisfiability problem for APAL is undecidable, but decidability can be achieved by considering a constraint similar to our restriction to existential announcements [French and van Ditmarsch, 2008; van Ditmarsch et al.].

8.4. Distributed systems

The systems we consider are closely related to the notion of *total order broadcast* in distributed systems [Raynal, 2013, p. 154]:

1. if a message is received, then it means that it has been broadcast;
2. no message is received twice;
3. if an agent received φ before φ', they all receive φ before φ' ;
4. φ causally precedes φ' implies that no agent receives φ' before φ;
5. if a message is broadcast, all agents will eventually receive it.

The first point holds in our system since a message (a formula) is only received if it is in the queue, which is the list of broadcast messages. The second point holds because a message is received when an agent's cut is increased to include that message from the queue, which only occurs once for each message. The third point holds because we have FIFO channels, and thus agents all receive messages in the same order, the order in which they are announced. The fourth point follows from the fact that in our systems we only consider a state $(w, \sigma{::}\psi, c)$ consistent if $(w, \sigma, c) \models \psi$, and because messages are received in order. The fifth point is not directly modelled in our systems since we only consider finite histories, but it is a kind of liveness constraint that we will probably be led to consider when we extend the logic with temporal operators (see next section).

More recently, [Griesmayer and Lomuscio, 2013] studies the model checking of distributed systems with respect to epistemic specifications. Although this work is in a synchronous setting, it is quite close to our approach in spirit, and shows that epistemic issues in distributed systems have practical implications, and a logical approach to these concerns can be fruitful.

Finally, we note that our definition of asynchronous models $\mathcal{M} \otimes \mathcal{A}$, especially the notion of cuts, is in the spirit of [Lamport, 1978].

9. Future work

This work is a first attempt to develop an epistemic logic for reasoning about asynchronous announcements. In the future, we would like to overcome the circularity problem, and define the semantics for the most general case (removing the finite tree and existential conditions). Using coinduction to define the set of consistent states may be one approach to this problem. Once we have defined the semantics for the general case, if possible, we hope to provide a complete axiomatization and a general model-checking algorithm. We also plan to implement the model-checking algorithms. Actually, we believe that the model checking of our logic could be reduced to recently proposed succinct languages for Dynamic epistemic logic [Charrier and Schwarzentruber, 2015, 2017]. Therefore, we could use symbolic techniques as presented in [van Benthem et al., 2015].

Second, we would like to model more general situations of asynchronous communication. We plan to consider the case where messages are not read in FIFO order, but are received and read in arbitrary order. We also plan to model the origin of the messages, allowing formulas such as "After agent a broadcasts φ, ψ holds". In our current setting, when the external broadcaster makes a new announcement, the only effect is to queue it in the channel without affecting anyone's epistemic state. However, in the case where the agents themselves make the announcements, agent a making an announcement should impact her knowledge: after the announcement she should know, for instance, that the channel is not empty. She should also know that after another agent checks their channel, that agent will know that ψ has been announced.

Third, it would be interesting to add temporal operators to our language, in order to express properties like "After p is announced and agent a receives it, *eventually* she will know that agent b knows p" (assuming that agents are forced to read announcements eventually).

Finally, we would like to model not only asynchronous broadcasts on a public channel but also private asynchronous communications between agents in the system. In essence, this amounts to defining a complete asynchronous version of dynamic epistemic logic [van Ditmarsch et al., 2007].

Acknowledgements. We would like to thank Hans van Ditmarsch who hosted the three authors in Nancy and who was the initiator of studying asynchrony in DEL.

References

Aucher, G., Schwarzentruber, F., 2013. On the complexity of dynamic epistemic logic. TARK '13.

Balbiani, P., Baltag, A., van Ditmarsch, H.P., Herzig, A., Hoshi, T., Lima, T.D., 2007. What can we achieve by arbitrary announcements?: A dynamic take on Fitch's knowability, in: TARK '07.

Balbiani, P., van Ditmarsch, H., Herzig, A., Lima, T.D., 2010. Tableaux for public announcement logic. J. Log. Comput. 20.

Balbiani, P., Gasquet, O., Schwarzentruber, F., 2013. Agents that look at one another. Logic Journal of the IGPL 21, 438–467. URL: http://dx.doi.org/10.1093/jigpal/jzs052, doi:10.1093/jigpal/jzs052.

Baltag, A., Ditmarsch, H.P.V., Moss, L.S., 2008. Epistemic logic and information update, in: In P. Adriaans, Elsevier Science Publishers.

Baltag, A., Moss, L.S., 2004. Logics for epistemic programs. Synthese 139, 165–224.

Baltag, A., Moss, L.S., Solecki, S., 1998. The logic of public announcements and common knowledge and private suspicions, in: Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98), Evanston, IL, USA, July 22-24, 1998, pp. 43–56.

van Benthem, J., van Eijck, J., Gattinger, M., Su, K., 2015. Symbolic model checking for dynamic epistemic logic, in: Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings, pp. 366–378.

Benthem, J.V., 2011. Logical dynamics of information and interaction. Cambridge University Press.

Brand, D., Zafiropulo, P., 1983. On communicating finite-state machines. Journal of the ACM (JACM) 30, 323–342.

Braüner, T., Blackburn, P., Polyanskaya, I., 2016. Second-order false-belief tasks: Analysis and formalization, in: Logic, Language, Information, and Computation - 23rd International Workshop, WoLLIC 2016, Puebla, Mexico, August 16-19th, 2016. Proceedings, pp. 125–144.

Chambart, P., Schnoebelen, P., 2008. Mixing lossy and perfect fifo channels, in: International Conference on Concurrency Theory, Springer. pp. 340–355.

Charrier, T., Schwarzentruber, F., 2015. 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, pp. 1471–1479. URL: http://dl.acm.org/citation.cfm?id=2773340.

Charrier, T., Schwarzentruber, F., 2017. Arbitrary public announcement logic with mental programs, in: Proceedings of the 2017 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2017 (to appear).

Corradini, F., Berardini, M.R.D., Vogler, W., 2003. Relating fairness and timing in process algebras, in: CONCUR '03.

Dégremont, C., Löwe, B., Witzel, A., 2011. The synchronicity of dynamic epistemic logic, in: TARK '11.

van Ditmarsch, H., 2017. Asynchronous announcements. CoRR abs/1705.03392. URL: http://arxiv.org/abs/1705.03392.

van Ditmarsch, H., French, T., Hales, J. Positive announcements (under submission).

van Ditmarsch, H., van der Hoek, W., Kooi, B.P., 2007. Dynamic epistemic logic. volume 337.

van Emde Boas, P., 1997. The convenience of tilings. Lecture Notes in Pure and Applied Mathematics.

Fagin, R., Halpern, J., Moses, Y., Vardi, M., 2004. Reasoning about knowledge. The MIT Press.

Fagin, R., Halpern, J.Y., Vardi, M.Y., 1992. What can machines know?: On the properties of knowledge in distributed systems. J. ACM.

French, T., van Ditmarsch, H.P., 2008. Undecidability for arbitrary public announcement logic, in: AiML '08.

Gasquet, O., Goranko, V., Schwarzentruber, F., 2015. Big brother logic: visual-epistemic reasoning in stationary multi-agent systems. Autonomous Agents and Multi-Agent Systems , 1–33URL: http://dx.doi.org/10.1007/s10458-015-9306-4, doi:10.1007/s10458-015-9306-4.

Gerbrandy, J., Groeneveld, W., 1997. Reasoning about information change. Journal of Logic, Language and Information 6, 147–169. doi:10.1023/A:1008222603071.

Griesmayer, A., Lomuscio, A., 2013. Model checking distributed systems against temporal-epistemic specifications, in: Formal Techniques for Distributed Systems. Springer, pp. 130–145.

Halpern, J.Y., Moses, Y., 1990. Knowledge and common knowledge in a distributed environment. J. ACM 37.

Hintikka, J., 1962. Knowledge and belief: an introduction to the logic of the two notions. volume 4. Cornell University Press Ithaca.

van der Hoek, W., 1990. Systems for knowledge and beliefs, in: Logics in AI, European Workshop, JELIA '90, Amsterdam, The Netherlands, September 10-14, 1990, Proceedings, pp. 267–281. URL: http://dx.doi.org/10.1007/BFb0018447, doi:10.1007/BFb0018447.

Jones, B.D., 1999. Bounded rationality, in: Annual Review of Political Science.

Knight, S., Maubert, B., Schwarzentruber, F., 2015. Asynchronous announcements in a public channel, in: ICTAC '15.

Kutschera, F., 1976. Einführung in die intensionale semantik.

van Lambalgen, M., 2010. Logical form as a determinant of cognitive processes, in: Logic, Language, Information and Computation, 17th International Workshop, WoLLIC 2010, Brasilia, Brazil, July 6-9, 2010. Proceedings, pp. 59–83.

Lamport, L., 1978. Time, clocks, and the ordering of events in a distributed system. Comm. ACM.

Lenzen, W., 1978. Recent work in epistemic logic.

Lutz, C., 2006. Complexity and succinctness of public announcement logic, in: AAMAS '06.

Moses, Y., Tuttle, M.R., 1988. Programming simultaneous actions using common knowledge. Algorithmica 3, 121–169. URL: http://dx.doi.org/10.1007/BF01762112, doi:10.1007/BF01762112.

Panangaden, P., Taylor, K., 1992. Concurrent common knowledge: Defining agreement for asynchronous

systems. Distributed Computing.
Plaza, J., 2007. Logics of public communications. Synthese 158.
Raynal, M., 2013. Distributed Algorithms for Message-Passing Systems. Springer.
Schnoebelen, P., 2002. The complexity of temporal logic model checking. Advances in modal logic 4, 35.
Sipser, M.. Introduction to the theory of computation. PWS Publishing Company.
Tarski, A., et al., 1955. A lattice-theoretical fixpoint theorem and its applications. Pac. jrn. of Math.
Winskel, G., 1993. The formal semantics of programming languages - an introduction. MIT Press.
Yu, Y.T., Gouda, M., 1982. Deadlock detection for a class of communicating finite state machines. IEEE Transactions on Communications 30, 2514–2518.
Zhang, J., Johansson, K.H., Lygeros, J., Sastry, S., 2001. Zeno hybrid systems. International journal of robust and nonlinear control 11, 435–451.

Appendix A.

Here we prove that, in the example of Section 3.5, we indeed have that

$$(w,\epsilon,\mathbf{0}) \models \langle p\rangle\langle\neg K_C p\rangle \bigcirc_B \bigcirc_B (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$$

1. $(w,\epsilon,\mathbf{0}) \models \langle p\rangle\langle\neg K_C p\rangle \bigcirc_B \bigcirc_B (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$ iff $(w,\epsilon,\mathbf{0}) \models p$, which is clearly true, and $(w,p,\mathbf{0}) \models \langle\neg K_C p\rangle \bigcirc_B \bigcirc_B (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$.
2. $(w,p,\mathbf{0}) \models \langle\neg K_C p\rangle \bigcirc_B \bigcirc_B (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$ iff $(w,p,\mathbf{0}) \models \neg K_C p$ and $(w,p{::}\neg K_C p,\mathbf{0}) \models \bigcirc_B \bigcirc_B (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B \neg K_C p)$.
   (a) $(w,p,\mathbf{0}) \models \neg K_C p$ iff $(w,p,\mathbf{0}) \not\models K_C p$.
   (b) $(w,p,\mathbf{0}) \not\models K_C p$ iff there exists S' s.t. $(w,p,\mathbf{0}) R_C S'$, $S' \models \checkmark$ and $S' \not\models p$. We notice that $(v,\epsilon,\mathbf{0})$ meets the requirements for S', so we conclude that $(w,p,\mathbf{0}) \models \neg K_C p$.
3. $(w,p{::}\neg K_C p,\mathbf{0}) \models \bigcirc_B\bigcirc_B(K_B\neg q\wedge\neg K_B K_C p\wedge\neg K_B\neg K_C p)$ iff $(w,p{::}\neg K_C p,\substack{B\mapsto 1\\ C\mapsto 0}) \models \bigcirc_B(K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B\neg K_C p)$.
4. $(w,p{::}\neg K_C p,\substack{B\mapsto 1\\ C\mapsto 0}) \models \bigcirc_B(K_B\neg q\wedge\neg K_B K_C p\wedge\neg K_B\neg K_C p)$ iff $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B\neg K_C p)$.
5. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models (K_B\neg q \wedge \neg K_B K_C p \wedge \neg K_B\neg K_C p)$ iff $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models K_B\neg q$ and $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models \neg K_B K_C p$ and $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models \neg K_B\neg K_C p$.
6. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models K_B\neg q$ iff for all S' s.t. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B S'$, if $S' \models \checkmark$ then $S' \models \neg q$. We see that if $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B (w,\sigma,c)$ then $w = t$, because if the initial state were s, $\neg K_C p$ would never be announcable. So indeed $S' \models \neg q$, and $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models K_B\neg q$.
7. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models \neg K_B K_C p$ iff $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \not\models K_B K_C p$.
   (a) $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \not\models K_B K_C p$ iff $\exists S'$ s.t. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B S'$ and $S' \models \checkmark$ and $S' \not\models K_C p$.
   (b) $S' \not\models K_C p$ iff $\exists S''$ s.t. $S' R_C S''$ and $S'' \models \checkmark$ and $S'' \not\models p$. We can choose $S' = (w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0})$ and $S'' = u,\epsilon,\mathbf{0}$ and we have that $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B S'$, $S' \models \checkmark$, $S' R_C S''$, $S'' \models \checkmark$ and $S'' \not\models p$.
8. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models \neg K_B\neg K_C p$ iff $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \not\models K_B\neg K_C p$.
9. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \not\models K_B\neg K_C p$ iff $\exists S'$ s.t. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B S'$ and $S' \models \checkmark$ and $S' \not\models \neg K_C p$, i.e. $S' \models K_C p$.
10. Thus, $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \not\models K_B\neg K_C p$ iff $\exists S'$ s.t. $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) R_B S'$ and $\forall S''$ if $S' R_C S''$ and $S'' \models \checkmark$, then $S'' \models p$. We can choose $S' = (w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 1})$ and then we see that for any consistent S'', if $S' R S''$ then $S'' \models p$. This shows that $(w,p{::}\neg K_C p,\substack{B\mapsto 2\\ C\mapsto 0}) \models \neg K_B\neg K_C p$.

Appendix B.

We consider the notion of (non-)Zeno behaviours, from the field of timed and hybrid systems. We describe how, modulo the adoption of a form of asynchrony weaker than the one considered in this work, this notion of non-Zeno behaviour could allow us to solve the circularity problem for the semantics of the full language, for arbitrary announcements and initial models.

In timed and hybrid systems, a behaviour of a system is a *Zeno behaviour* if countably infinitely many discrete events occur in a finite time [Zhang et al., 2001; Corradini et al., 2003]. This is of course impossible in real systems, but such behaviours can occur in models of systems due to abstraction, and many works either study how to detect such behaviours, eliminate them, or directly consider only non-Zeno models, *i.e.*, models that do not present Zeno behaviours.

Similarly, let us here assume that our systems are non-Zeno: only a finite number of discrete events (announcements/reception of formulas) occur in a finite time interval. In fact, we make the stronger assumption that the number of messages sent during one unit of time is bounded. Without loss of generality, we suppose that the number of messages sent during one unit of time is at most one (otherwise, change the time unit). We also suppose that reading a formula takes one unit of time. These assumptions are somewhat idealistic, since the time necessary to send or read a message may be influenced by many factors, such as the length of the message. However, it may be achievable in some circumstances, for example by waiting after sending or receiving a message, in order to use a uniform amount of time.

Note that in the rest of the paper we never mentioned *time* in our systems. Here we need to for the notion of non-Zeno systems to make sense. We thus assume a global clock, and in addition we make the rather strong assumption that all agents have access to this clock, and that this is common knowledge.

Fagin et al. [1992, p. 333] wrote:

> Is the system synchronous? That is, is there a "global clock" that every process can "see," so that every process "knows the time"?

With the assumption that agents have access to a global clock, our systems are not asynchronous according to this definition. However we argue that this definition does not apply here, and that even with the global clock assumption our framework remains asynchronous in spirit. The first reason is that communication remains asynchronous: the delay between sending of an announcement and reception by each agent is unbounded. The second reason is that even though agents have access to a global clock and thus know the time, they cannot talk about it and synchronize. However, knowing the time and the fact that at most one announcement is made per time unit allows agents to refine their pre-accessibility relation by removing all possible states that contain too many announcements. This is enough to solve the problem of circular definition, as we detail now.

First, we introduce the time of the global clock in the states of the models, so that formulas are now evaluated on states of the form (w, σ, c, t) where (w, σ, c) is as before and t is the time represented as a positive integer. Note that because we assumed that sending of an announcement takes one time unit, we always have that $|\sigma| \leq t$.

We define the satisfaction relation $(w, \sigma, c, t) \models \varphi$ by induction as follows:

$$\begin{array}{lll}
(w,\sigma,c,t) \models p & \text{if} & p \in \Pi(w) \\
(w,\sigma,c,t) \models (\varphi_1 \wedge \varphi_2) & \text{if} & (w,\sigma,c,t) \models \varphi_1 \text{ and } (w,\sigma,c,t) \models \varphi_2 \\
(w,\sigma,c,t) \models \neg\varphi & \text{if} & (w,\sigma,c,t) \not\models \varphi \\
(w,\sigma,c,t) \models K_a\varphi & \text{if} & \text{for all } S' \text{ s.t. } (w,\sigma,c)R_a(w',\sigma',c'), |\sigma'| \leq t \\
 & & \text{ and } (w',\sigma',c',t) \models \checkmark,\ (w',\sigma',c',t) \models \varphi \\
(w,\sigma,c,t) \models \langle\psi\rangle\varphi & \text{if} & \sigma{::}\psi \in \mathsf{Seq}(\mathcal{A}),\ \ (w,\sigma,c,t) \models \psi \text{ and } (w,\sigma{::}\psi,c,t{+}1) \models \varphi \\
(w,\sigma,c,t) \models \bigcirc_a\varphi & \text{if} & c(a) < |\sigma| \text{ and } (w,\sigma,c^{+a},t+1) \models \varphi \\
 & & \text{ where } c^{+a}(b) = \begin{cases} c(b) & \text{if } b \neq a \\ c(b)+1 & \text{if } b = a \end{cases} \\
(w,\epsilon,\mathbf{0},t) \models \checkmark & & \\
(w,\sigma,c,t) \models \checkmark & \text{if} & \text{either there is } c' < c \text{ such that } (w,\sigma,c',t) \models \checkmark \\
 & & \text{or } \sigma = \sigma'{::}\psi \text{ and there is } t' < t \text{ such that} \\
 & & (w,\sigma',c) \in \mathcal{S},\ (w,\sigma',c,t') \models \checkmark \\
 & & \text{and } (w,\sigma',c,t') \models \psi
\end{array}$$

The definition is by induction on the lexicographical order on $(t, |\varphi|)$. Observe that in the last clause, where $(w, \epsilon, \mathbf{0}, t) \models \checkmark$ requires $(w, \sigma', c, t') \models \psi$ to be defined, we have $t' < t$. Also, in the clause for the knowledge operator, we restrict the pre-accessibility relation to those states that do not contain more messages than what can have been announced since the beginning. These two observations suffice to see that the induction is well-founded.

So in a sense, our strong non-Zeno assumption together with the common knowledge of a global clock tames the effect of the agents' power to imagine pending messages. We already described in Section 3.4 how removing this assumption on agents' power to imagine solves the circularity problem. In this section we have shown that it is enough to forbid them to imagine too much.

Finally we show with an example that even with common knowledge of a global clock our framework remains asynchronous.

Example 9. *In synchronous public announcement logic, common knowledge*[4] *of formula p is achieved when p is announced. For example, in a two-agent system the following formula is always true: $\langle p\rangle_{PAL} C_{a,b}p$. In our systems, message sending and reception are separate, and there is no common knowledge operator, but if systems with common knowledge of a global clock were equivalent to synchronous systems, we would expect $\langle p\rangle \bigcirc_a \bigcirc_b (K_aK_bp \wedge K_bK_ap)$ to hold always, since $C_{a,b}p \longrightarrow K_aK_bp \wedge K_bK_ap$. However, it is easy to see that this does not always hold. Consider for instance a system with two states, u and v, where p holds at u and not at v, and u and v are equivalent for agents a and b. It is straightforward to see that $(u, \epsilon, \mathbf{0}, 0) \models \neg\langle p\rangle \bigcirc_a \bigcirc_b (K_aK_bp \wedge K_bK_ap)$. Furthermore, it can be shown that for any sequence of formulas $\varphi_1, ..., \varphi_k$, there exists n such that $\langle\varphi_1\rangle \ldots \langle\varphi_k\rangle(\bigcirc_a\bigcirc_b)^k(\neg(K_aK_b)^np \wedge \neg(K_bK_a)^np)$, which strongly suggests that common knowledge is not attainable in these systems.*

[4] In an S5 system, common knowledge of p is the formalization of "everybody knows p, everybody knows that everybody knows p, and so on." In particular, $C_{\{a,b\}}p = p \wedge K_ap \wedge K_bp \wedge K_aK_bp \wedge K_bK_ap \wedge \ldots$

Appendix C.

Theorem 3. *The model checking problem for propositional protocols is* PSPACE*-hard.*

Proof. We give a polynomial-time reduction from the quantified boolean formula (QBF) satisfiability problem [Sipser] to the model checking problem for propositional protocols.

Reduction definition. Let $\exists p_1 \forall p_2 \ldots \forall p_{2n} \chi(p_1, \ldots, p_{2n})$ be a quantified boolean formula where n is an integer. We define an instance $(\mathcal{M}, w_0, \mathcal{A}, \varphi_0)$ of the model checking problem for propositional protocols.

First we consider fresh atomic propositions $p_i^\top$ and $p_i^\perp$ for $i \in \{1, \ldots, 2n\}$, whose intuitive meanings are respectively 'p_i is true' and 'p_i is false'.

1. We define the model $\mathcal{M} = (W, \{\rightarrow_a\}_{a \in Ag}, \Pi)$ such that:
   - $W = \{w_{p_1^\top}, \ldots, w_{p_{2n}^\top}, w_{p_1^\perp}, \ldots, w_{p_{2n}^\perp}\}$;
   - for all $a \in Ag$, $\rightarrow_a = W \times W$;
   - $\Pi(w_\alpha) = \{\alpha\}$.
2. The world w_0 is $w_{p_1^\perp}$ (but it could be any other world in W).
3. The announcement protocol $\mathcal{A}$ is $\{\neg p_1^\top, \ldots, \neg p_{2n}^\top, \neg p_1^\perp, \ldots, \neg p_{2n}^\perp\}$.
   Now we define the following abbreviations:
   - $\mathsf{isdef}_a(p_i) := (\hat{K}_a p_i^\top \wedge K_a \neg p_i^\perp) \vee (\hat{K}_a p_i^\perp \wedge K_a \neg p_i^\top)$, to be read "$p_i$ is defined";
   - $\mathsf{istrue}_a(p_i) := (\hat{K}_a p_i^\top)$, to be read "$p_i$ is true".
4. The formula φ_0 is ψ_1 where the sequence $(\psi_\ell)_{\ell:=1..2n+1}$ is defined by induction:
   - Base case: $\psi_{2n+1} := \chi(\mathsf{istrue}_a(p_1), \ldots, \mathsf{istrue}_a(p_{2n}))$;
   - Inductive case: for all $\ell \in \{1, \ldots, 2n\}$,
     - $\psi_\ell := \hat{K}_a(\bigwedge_{j=1}^{\ell} \mathsf{isdef}_b(p_j) \wedge \bigwedge_{j=\ell+1}^{2n} \neg \mathsf{isdef}_b(p_j) \wedge \psi_{\ell+1})$ if ℓ is odd;
     - $\psi_\ell := K_b((\bigwedge_{j=1}^{\ell} \mathsf{isdef}_a(p_j) \wedge \bigwedge_{j=\ell+1}^{2n} \neg \mathsf{isdef}_a(p_j)) \rightarrow \psi_{\ell+1})$ if ℓ is even.

We claim that $\exists p_1 \forall p_2 \ldots \forall p_{2n} \chi(p_1, \ldots, p_{2n})$ is true if, and only if $\mathcal{M} \otimes \mathcal{A}, (w_0, \epsilon, \mathbf{0}) \models \psi_1$.

Reduction correctness. We prove by recurrence on ℓ the following property $P(\ell)$, for all $\ell \in \{1, \ldots, 2n+1\}$:

For all valuations ν, for all $w \in W$ such that $(w, \sigma_{\nu,\ell}, c_{\nu,\ell})$ is consistent,

$$\nu \models_{\text{QBF}} Q_\ell p_\ell \ldots \forall p_{2n} \chi(p_1, \ldots, p_{2n}) \text{ iff } \mathcal{M} \otimes \mathcal{A}, (w, \sigma_{\nu,\ell}, c_{\nu,\ell}) \models \psi_\ell$$

where

- $Q_\ell = \forall$ if ℓ is even and $Q_\ell = \exists$ if ℓ is odd;
- $\sigma_{\nu,\ell}$ is any sequence σ of $\mathsf{Seq}(\mathcal{A})$ such that the prefix $\sigma|_\ell$ is $[\neg p_1^{\neg\nu(p_1)}, \ldots, \neg p_{\ell-1}^{\neg\nu(p_{\ell-1})}]$;
- $c_{\nu,1} = \mathbf{0}$;
- if $\ell > 1$, $c_{\nu,\ell} = \begin{smallmatrix} a & \mapsto \ell-2 \\ b & \mapsto \ell-1 \end{smallmatrix}$ if ℓ is even and $c_{\nu,\ell} = \begin{smallmatrix} a & \mapsto \ell-1 \\ b & \mapsto \ell-2 \end{smallmatrix}$ if ℓ is odd.

The following picture shows a branch of states reached in the asynchronous model $\mathcal{M} \otimes \mathcal{A}$ when we evaluate formula ψ_1:

$$
\begin{array}{ll}
w_1, \epsilon, \mathbf{0} & \psi_1 \\
\downarrow a & \\
w_2, [\neg p_1^?],\ \begin{array}{l} a \mapsto 0 \\ b \mapsto 1 \end{array} & \psi_2 \\
\downarrow b & \\
w_3, [\neg p_1^?; \neg p_2^?;],\ \begin{array}{l} a \mapsto 2 \\ b \mapsto 1 \end{array} & \psi_3 \\
\downarrow a & \\
\vdots & \\
\downarrow a & \\
w_{2n}, [\neg p_1^?; \neg p_2^?; \dots; \neg p_{2n-1}^?],\ \begin{array}{l} a \mapsto 2n-2 \\ b \mapsto 2n-1 \end{array} & \psi_{2n} \\
\downarrow b & \\
w_{2n+1}, [\neg p_1^?; \neg p_2^?; \dots; \neg p_{2n-1}^? \neg p_{2n}^?],\ \begin{array}{l} a \mapsto 2n \\ b \mapsto 2n-1 \end{array} & \psi_{2n+1}
\end{array}
$$

where ? stands for either $\top$ or $\bot$.

$\boxed{P(2n+1)}$ One can check that for all i, $\nu \models p_i$ iff $w, \sigma_{\nu,2n+1}, c_{\nu,2n+1} \models \mathsf{istrue}_a(p_i)$. Using this, one can prove the base case by induction on χ:

$$\nu \models_{\mathrm{QBF}} \chi(p_1, \dots, p_{2n}) \text{ iff } w, \sigma_{\nu,2n+1}, c_{\nu,2n+1} \models \chi(\mathsf{istrue}_a(p_1), \dots, \mathsf{istrue}_a(p_{2n})).$$

$\boxed{P(\ell+1) \Rightarrow P(\ell)}$ Suppose that $P(\ell+1)$ holds and let us prove that $P(\ell)$ holds. We consider the case when ℓ is odd (the case when ℓ is even is similar). Let ν be a valuation and w a world such that $(w, \sigma_{\nu,\ell}, c_{\nu,\ell})$ is consistent.

$\nu \models_{\mathrm{QBF}} \exists p_\ell \dots \forall p_{2n} \chi(p_1, \dots, p_{2n})$

iff there exists $v \in \{0,1\}$ s.t. $\nu[p_\ell := v] \models_{\mathrm{QBF}} \forall p_{\ell+1} \dots \forall p_{2n} \chi(p_1, \dots, p_{2n})$

iff there exists $v \in \{0,1\}$ s.t. for all $u \in W$, if $(u, \sigma_{\nu[p_\ell:=v],\ell+1}, c_{\nu[p_\ell:=v],\ell+1})$ is consistent then $\mathcal{M} \otimes \mathcal{A}, (u, \sigma_{\nu[p_\ell:=v],\ell+1}, c_{\nu[p_\ell:=v],\ell+1}) \models \psi_{\ell+1}$ (by $P(\ell+1)$)

iff there exists $v \in \{0,1\}$ and a world $u \in W$ s.t. $(u, \sigma_{\nu[p_\ell:=v],\ell+1}, c_{\nu[p_\ell:=v],\ell+1})$ is consistent and $\mathcal{M} \otimes \mathcal{A}, (u, \sigma_{\nu[p_\ell:=v],\ell+1}, c_{\nu[p_\ell:=v],\ell+1}) \models \psi_{\ell+1}$
(because the choice of the world does not matter as long as it satisfies the announcements in $\sigma_{\nu[p_\ell:=v],\ell+1}$, and there always are at least $2n$ such worlds.)

iff $\mathcal{M} \otimes \mathcal{A}, (w, \sigma_{\nu,\ell}, c_{\nu,\ell}) \models \psi_\ell$
(because $\bigwedge_{j=1}^{\ell} \mathsf{isdef}_b(p_j)) \wedge \bigwedge_{j=\ell+1}^{2n} \neg\mathsf{isdef}_b(p_j)$ in ψ_ℓ restricts the states that agent a considers possible to those in which agent b has received either $p_\ell^\top$ or $p_\ell^\bot$.)

$\boxed{\text{Conclusion}}$ By $P(1)$, $\exists p_1 \forall p_2 \dots \forall p_{2n} \chi(p_1, \dots, p_{2n})$ is true iff $\mathcal{M} \otimes \mathcal{A}, w_{p_1^\bot}, \epsilon, \mathbf{0} \models \psi_1$. ∎

Appendix D.

Theorem 7. *The satisfiability problem when the protocol is finite and propositional and if the number of agents is greater than 2 is* NEXPTIME*-hard.*

Proof. The proof follows the same idea as the proof of NEXPTIME-hardness of the satisfiability problem in dynamic epistemic logic [Aucher and Schwarzentruber, 2013]: we prove that the satisfiability problem when the protocol is finite and propositional is NEXPTIME-hard by reducing a NEXPTIME-hard tiling problem [van Emde Boas, 1997] to it. Let C be a countable and infinite set of colors. A *tile type* t is a 4-tuple of colors, denoted $t = (left(t), right(t), up(t), down(t)) \in C^4$. We consider the following *tiling problem*:

> **Input**: a finite set T of tile types, $t_0 \in T$ and a natural number k written in its binary form.
> **Output**: yes iff there exists a function f from $\{0, \dots, k-1\}^2$ to T satisfying:
> (t_0) $f(0,0) = t_0$;
> (v) for all $x \in \{0, ..., k-1\}$ and $y \in \{0, ..., k-2\}$: $up(f(x,y)) = down(f(x, y{+}1))$;
> (h) for all $x \in \{0, ..., k-2\}$ and $y \in \{0, ..., k-1\}$: $right(f(x,y)) = left(f(x{+}1, y))$.

In other words, the problem is to decide whether we can tile a $k \times k$ grid with the tile types of T, t_0 being placed onto $(0,0)$ (Figure 7a shows a 4×4 tiling).

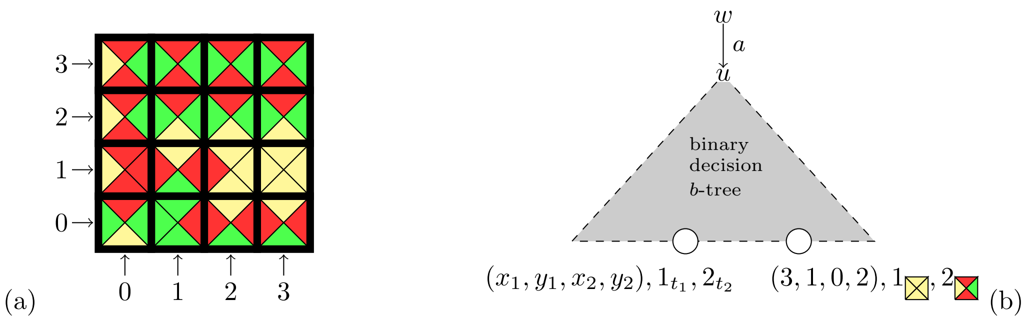


Figure 7: A 4×4 tiling ($k = 4$) and an idea of the initial Kripke model

Let us consider an instance (T, t_0, k) of the tiling problem, and without loss of generality, assume that $k = 2^n$. We define the instance of our satisfiability problem $tr(T, t_0, k) = \langle \mathcal{A}, \varphi \rangle$ where $\mathcal{A} = \{\mathsf{B}_0, \dots, \mathsf{B}_{4n-1}, \mathsf{b}_0, \dots, \mathsf{b}_{4n-1}\}$, where each B_i and b_i is an atomic proposition, and φ is the conjunction of formulas of Figure 8. Observe that this reduction is computable in polynomial time in the size of (T, t_0, k). We prove that (T, t_0, k) is a positive instance of the tiling problem iff φ is $\mathcal{A}$-satisfiable.

General idea. Formula φ enforces an encoding of *two identical* $2^n \times 2^n$-tilings into a single tree (see Figure 7b). Each leaf of the tree represents both a position (x_1, y_1) in the first tiling and a position (x_2, y_2) in the second one. Encoding two copies allows us to compare a tile with the ones around it locally, in leaves coding for adjacent positions, and thus without having to compare different leaves of the tree. This greatly simplifies the task of verifying whether a tree represents a valid tiling.

$$\bigwedge_{j<4n} (\mathsf{B}_j \wedge \mathsf{b}_j) \tag{1}$$

$$\begin{aligned}\hat{K}_a \bigwedge_{\ell<4n} K_b^\ell \quad \Big(& \bigwedge_{j\geq\ell} (\mathsf{B}_j \wedge \mathsf{b}_j) \wedge \hat{K}_b(\mathsf{B}_\ell \wedge \neg \mathsf{b}_\ell) \wedge \hat{K}_b(\neg\mathsf{B}_\ell \wedge \mathsf{b}_\ell) \wedge K_b((\mathsf{B}_\ell \wedge \neg\mathsf{b}_\ell) \vee (\mathsf{b}_\ell \wedge \neg \mathsf{B}_\ell)) \wedge \\ & \bigwedge_{i<\ell} ((\mathsf{B}_i \to K_b\mathsf{B}_i) \wedge (\neg\mathsf{B}_i \to K_b\neg\mathsf{B}_i) \wedge (\mathsf{b}_i \to K_b\mathsf{b}_i) \wedge (\neg\mathsf{b}_i \to K_b\neg\mathsf{b}_i))\Big)\end{aligned} \tag{2}$$

$$K_a K_b^{4n}\Big(\bigvee_{t\in T} 1_t \wedge \bigvee_{t\in T} 2_t \wedge \bigwedge \big\{(1_t \to \neg 1_{t'}) \wedge (2_t \to \neg 2_{t'}) \mid t, t' \in T, t \neq t'\big\}\Big) \tag{3}$$

$$K_a K_b^{4n}\Big((x_1 = x_2) \wedge (y_1 = y_2) \to \bigwedge_{t\in T} (1_t \leftrightarrow 2_t)\Big) \tag{4}$$

$$K_a(\bigwedge_{i=1}^{2n} K_b^i(K_b\mathsf{B}_i \vee K_b\mathsf{b}_i) \to \bigvee_{t\in T} K_b^{4n} 1_t) \tag{5}$$

$$K_a(\bigwedge_{i=2n+1}^{4n} K_b^i(K_b\mathsf{B}_i \vee K_b\mathsf{b}_i) \to \bigvee_{t\in T} K_b^{4n} 2_t) \tag{6}$$

$$K_a K_b^{4n}\,(((x_1 = 0) \wedge (y_1 = 0)) \to t_0) \tag{7}$$

$$K_a K_b^{4n}\Big((x_1{=}x_2) \wedge (y_2{=}y_1{+}1) \to \bigwedge_{t\in T}\Big(1_t \to \bigvee_{t'\in T, down(t')=up(t)} 2_{t'}\Big)\Big) \tag{8}$$

$$K_a K_b^{4n}\Big((x_2{=}x_1{+}1) \wedge (y_1{=}y_2) \to \bigwedge_{t\in T}\Big(1_t \to \bigvee_{t'\in T, left(t')=right(t)} 2_{t'}\Big)\Big) \tag{9}$$

Figure 8: Clauses in φ

The tile types of the first tiling are represented by atomic propositions 1_t and the tile types of the second tiling are represented by atomic propositions $2_{t'}$, where t and t' range over T. They hold at a leaf of the tree whose coordinates correspond to (x_1, y_1) and (x_2, y_2) when the tile type of the first tiling at coordinate (x_1, y_1) is t and the tile type of the second tiling at coordinate (x_2, y_2) is t'.

We enforce the consistency of the binary tree: for instance, all $(x_1, y_1, *, *)$-leaves should be tagged with the same proposition 1_t. To this aim, we need to select all $(x_1, y_1, *, *)$-leaves; this is performed by an "arbitrary" announcement of coordinates in the first tiling. This announcement is imagined by agent a, reason why the tree starts in an a-child of the initial world w (the same technique applies for selecting $(*, *, x_2, y_2)$-leaves).

Encoding coordinates. The coordinates (x_1, y_1) and (x_2, y_2) of the two tilings are represented by a valuation over atomic propositions $\mathsf{B}_0, \ldots, \mathsf{B}_{4n-1}, \mathsf{b}_0, \ldots, \mathsf{b}_{4n-1}$. More precisely, the set $X_1 = \{\mathsf{B}_0, ..., \mathsf{B}_{n-1}, \mathsf{b}_0, ..., \mathsf{b}_{n-1}\}$ contains the atomic propositions encoding the binary representation of the integer x_1 as follows:

- B_i means that the ith bit of x_1 is 1; b_i means that the ith bit of x_1 is 0;
- if B_i and b_i are both true it means that the ith bit is not set yet;
- valuations where B_i and b_i are both false are never considered.

Similarly, $Y_1 = \{\mathsf{B}_n, ..., \mathsf{B}_{2n-1}, \mathsf{b}_n, ..., \mathsf{b}_{2n-1}\}$, $X_2 = \{\mathsf{B}_{2n}, ..., \mathsf{B}_{3n-1}, \mathsf{b}_{2n}, ..., \mathsf{b}_{3n-1}\}$ and $Y_2 = \{\mathsf{B}_{3n}, ..., \mathsf{B}_{4n-1}, \mathsf{b}_{3n}, ..., \mathsf{b}_{4n-1}\}$ contain the atomic propositions encoding binary representations of integers y_1, x_2 and y_2, respectively. For instance, for $n = 4$, the coordinates $(x_1, y_1) = (4, 3)$ and $(x_2, y_2) = (12, 2)$ are represented at a leaf of the tree by the valuation (we recall that in binary notation, 4 is represented by $\overline{0100}$, 3 is represented by $\overline{0011}$, 12 is represented by $\overline{1100}$ and 2 is represented by $\overline{0010}$):

$$\underbrace{\neg\mathsf{B}_0, \mathsf{b}_0\ ,\ \neg\mathsf{b}_1, \mathsf{B}_1\ ,\ \neg\mathsf{B}_2, \mathsf{b}_2\ ,\ \neg\mathsf{B}_3, \mathsf{b}_3}_{4}\ \underbrace{\neg\mathsf{B}_4, \mathsf{b}_4\ ,\ \neg\mathsf{B}_5, \mathsf{b}_5\ ,\ \neg\mathsf{b}_6, \mathsf{B}_6\ ,\ \neg\mathsf{b}_7, \mathsf{B}_7}_{3}$$

$$\underbrace{\neg\mathsf{b}_8, \mathsf{B}_8\ ,\ \neg\mathsf{b}_9, \mathsf{B}_9\ ,\ \neg\mathsf{B}_{10}, \mathsf{b}_{10}\ ,\ \neg\mathsf{B}_{11}, \mathsf{b}_{11}}_{12}\ \underbrace{\neg\mathsf{B}_{12}, \mathsf{b}_{12}\ ,\ \neg\mathsf{B}_{13}, \mathsf{b}_{13}\ ,\ \neg\mathsf{b}_{14}, \mathsf{B}_{14}\ ,\ \neg\mathsf{B}_{15}, \mathsf{b}_{15}}_{2}$$

In order to ensure constraints (v) and (h) in the definition of a tiling, we need to compare tiles that are adjacent in a tiling. Boolean formulas encode the properties $x_1{=}x_2$, $x_2{=}x_1{+}1$, $y_1{=}y_2$ or $y_2{=}y_1{+}1$. For instance:

$$(x_1{=}x_2) \triangleq \bigwedge_{i<n} (\mathsf{B}_i {\leftrightarrow} \mathsf{B}_{i+2n}) \wedge (\mathsf{b}_i {\leftrightarrow} \mathsf{b}_{i+2n})$$

$$(x_2{=}x_1{+}1) \triangleq \bigvee_{i<n} \left(\bigwedge_{j<i} (\mathsf{B}_j {\leftrightarrow} \mathsf{B}_{j+2n}) \wedge (\mathsf{b}_j {\leftrightarrow} \mathsf{b}_{j+2n}) \wedge \mathsf{b}_i \wedge \mathsf{B}_{i+2n} \wedge \bigwedge_{i<j<n} (\mathsf{B}_{j+2n} \wedge \mathsf{b}_j) \right)$$

Announcements. With $\mathcal{A}$, we can announce bit values of coordinates in the first or second tiling and Formula 1 ensures that all formulas in $\mathcal{A}$ are true and hence can be successfully announced.

Tree structure. Formula 2 ensures that there exists an a-successor u such that the epistemic model pointed at u is bisimilar up to modal depth $4n$ to a binary tree (with b-relation between nodes) whose leaves' valuations represent all possible pairs of positions $(x_1, y_1, x_2, y_2) \in \{0, \dots, 2^n - 1\}^4$. Subformula $\bigwedge_{j>\ell}(\mathsf{B}_j \wedge \mathsf{b}_j)$ means that at level ℓ, the j-bits for $j > \ell$ are not set yet. Informally, a leaf corresponds to a pair of one cell in the first tiling and one cell in the second tiling. In Formula 2, modality $\hat{K}_a$ imposes the existence of a new world which is the root of the tree (the root is not the initial world directly because we use agent a to imagine possible announcements). This modality $\hat{K}_a$ also considers states in which b has received announcements; but as we require the state imagined by agent a to be related to states that verify $\mathsf{B}_\ell \wedge \neg\mathsf{b}_\ell$ and states that verify $\neg\mathsf{B}_\ell \wedge \mathsf{b}_\ell$ for all ℓ, we rule out the possibility that agent b has received any announcement.

Encoding two (unconstrained) tilings. Formulas 3 encodes that, *at each leaf of the tree*, there is exactly one tile type for the first tiling and exactly one tile type for the second tiling. Formula 4 encodes the fact that when these two pairs of coordinates coincide, that is when $x_1{=}x_2$ and $y_1{=}y_2$, then the tile type of the first tiling and the tile type of the second tiling are identical.

It may be the case that in the tree, two different leaves with the *same* valuation have different tile types. Therefore, we also have to constrain the tree so that the leaves denoting the same position in the first tiling (resp. second tiling) contain the same tile type for the first tiling (resp. second tiling). This is expressed by formulas 5 and 6.

In Formula 5, modality K_a universally picks a sequence of announcements. The guard $\bigwedge_{i=1}^{2n} K_b^i(K_b\mathsf{B}_i \vee K_b\mathsf{b}_i)$ ensures that all bits of (x_1, y_1) have been announced: at each step $i \leq 2n$ either B_i or b_i has been announced and read by b (checked by the fact that either b knows B_i or b knows b_i). Maybe more has been announced: for instance B_{2n+1}. In particular, b considers sequences of announcements where only coordinates (x_1, y_1) have been announced (and no more). It selects the branches where valuations on the branch respect the announcement:

- either bits are not yet defined (and then it respects the announcements);
- or a bit of (x_1, y_1) is set and it should respect the announcement.

All leaves in selected branches correspond to the announced value of (x_1, y_1). Then, the formula $\bigvee_{t\in T} K_b^{4n} 1_t$ checks that these leaves are of the same tile type t. Likewise with Formula 6 for the second tiling.

So, with formulas 3-6, we encode in the tree two identical (unconstrained) tilings in a single tree. It remains to enforce that this tiling is valid.

Encoding constraints (t_0)*,* (v) *and* (h)*.* They are expressed respectively by formulas 7-9. As we said at the beginning of the proof, the latter two constraints motivate the encoding of *two* tilings. Comparing adjacent positions would not be possible with our epistemic language if the tree encoded a single tiling.

One can then check that there exists a tiling for the instance (T, t_0, k) of the tiling problem iff formula φ is $\mathcal{A}$-satisfiable. ∎