Asynchronous announcements in a public channel

Sophia Knight[1], Bastien Maubert[1], and François Schwarzentruber[2]

[1] LORIA - CNRS / Université de Lorraine,
sophia.knight@gmail.com, bastien.maubert@gmail.com
[2] ENS Rennes - IRISA - LORIA
francois.schwarzentruber@ens-rennes.fr

Abstract. We propose a variant of public announcement logic for asynchronous systems. We give a syntax where sending and receiving messages are modeled by different modal operators. 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. Finally, we provide complexity results for the model checking problem.

1 Introduction

Asynchrony has long played a central role in distributed systems, where access to a centralized clock is not always possible, and where communication may not be delivered or received immediately or predictably. Recently, with the proliferation of multi-agent systems (MAS) where independent agents interact, communicate, and make decisions under imperfect information, modelling how knowledge evolves with informative events has also become increasingly important. One of the first and most influential proposals in this direction is public announcement logic (PAL) [11], in which some external omniscient entity publicly makes true announcements to some group of agents. This logic has led to the powerful and much studied dynamic epistemic logic (DEL) [14]. However, both these logics assume synchronicity, even though there has been some discussion on this matter for the latter [4]. In PAL for instance, messages are immediately received by all agents at the same time, as soon as they are sent. As far as we know, little work has been done to address the same problem in asynchronous scenarios.

Our goal in this work is the logical study of scenarios with *asynchronous* announcements. As a first step, similar to PAL, we consider a simple scenario: messages are true at the time of announcement, public (directed to everyone), and we do not model their origin, but rather assume that some external and omniscient entity emits them. Consider the scenario where three autonomous agents, moving through an area, receive messages from a public channel. They do not all read the messages (logical formulas) at the same time, but they do read them in the same order. Figure 1 depicts the architecture of the system: each agent has a private copy of the channel and they read messages in *first in, first out* (FIFO) order, that is, messages are read in the order they are sent.

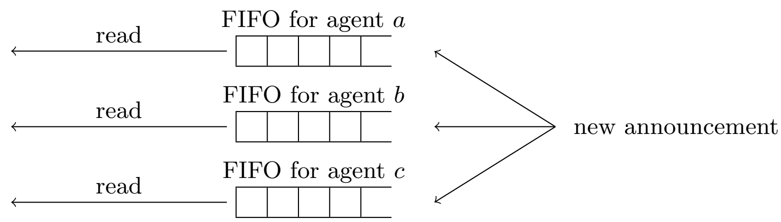


Fig. 1. Agent architecture

In PAL, not only are all messages received at the same time by all agents, but they are also received at the same time they are sent. Therefore, in PAL, 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 the channel that she has not read yet. Thus, in our logic, we provide the following modal operators:

- $K_i\varphi$, where i is an agent and φ a formula. Intuitively, this will mean "agent i knows φ," and as usual, this will be interpreted as "φ is true at every state that agent i considers possible." Below, we discuss the accessibility relation we use to define all the states agent i considers possible at a given state (all states that are *indistinguishable* from the current state for agent i).
- $\langle\psi\rangle\varphi$, which will mean "after the currently true formula ψ is (asynchronously) announced, φ is immediately true."
- $\bigcirc_i\varphi$ which will mean "after agent i *receives* the next announced formula in her queue, φ is immediately true."

Interestingly, the most intuitive semantics for this logic presents a challenging problem of circular definition. We describe this difficulty in more detail below, but the basic issue is that in order to check the truth of some formulas, we must quantify over the set of all indistinguishable states that are consistent. A state is consistent if it is the result of making a true announcement in a state that is itself consistent. So evaluating the truth of a formula requires determining whether a state is consistent, which in turn requires evaluating the truth of formulas. In PAL, a similar problem occurs, as the definition of the update of a model by an announcement and the definition of the truth values are mutually dependent. While in PAL this circularity can be solved simply by resorting to a double induction, things are more complicated here. Indeed, because of asynchrony, an agent does not know what or how many messages other agents have received; therefore, evaluating a knowledge operator in a state requires considering possibly infinitely many indistinguishable states, which makes a double induction impossible. This circularity problem is inherent to the asynchronous setting, and is independent from our choice to consider an external source for the announcements.

We partially tackle this issue by defining two 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. In the latter case, the semantics is defined thanks to an application of the Knaster-Tarski fixed point theorem [12].

Finally, we 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

The paper is organized as follows. In Section 2, we recall (synchronous) public announcement logic. In Section 3 we present the language and the models for asynchronous public announcement logic we propose here. In Section 4 we present the circularity problem for defining the semantics of the logic, and we exhibit two cases where it can be solved. We then present some validities in Section 5, and we study the model checking problem in Section 6. Finally we discuss related work in Section 7 and future work in Section 8.

2 Background: Public Announcement Logic

In this section, we present background on (synchronous) Public Announcement Logic (PAL) [11]. Let $\mathcal{P}$ be a countable infinite set of *atomic propositions*, and let AGT be a finite set of *agents*.

Definition 1 (Syntax of PAL). *The syntax for PAL is as follows:*

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

where $p \in \mathcal{P}$ and $i \in AGT$.

The intuitive meaning of the last two operators is the following: $K_i\varphi$ means that agent i knows φ, $\langle\psi\rangle_{\mathrm{PAL}}\varphi$ means that ψ is true and after ψ has been publicly announced and publicly received by all the agents, φ holds.

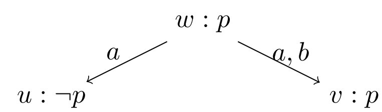
The semantics of PAL relies on classic Kripke models and the *possible worlds* semantics, widely used in logics of knowledge [5].

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

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

Note that we do not require the accessibility relations to be equivalence relations as is traditionally done in epistemic logic [14].

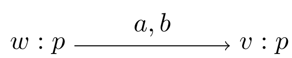
Example 1. Let us consider the following Kripke model, where w, u and v are worlds, a and b are agents and p is a proposition. The arrows represent the agents' accessibility relations. At world w, agent a considers u and v possible, and agent b considers only world v possible.



The semantics is given as follows:

- $\mathcal{M}, w \models p$ if $p \in \Pi(w)$;
- $\mathcal{M}, w \models \varphi_1 \wedge \varphi_2$ if $\mathcal{M}, w \models \varphi_1$ and $\mathcal{M}, w \models \varphi_2$;
- $\mathcal{M}, w \models \neg\varphi$ if $\mathcal{M}, w \not\models \varphi$;
- $\mathcal{M}, w \models K_i\varphi$ if for all u such that $w \rightarrow_i u$, $\mathcal{M}, u \models \varphi$;
- $\mathcal{M}, w \models \langle\psi\rangle_{\text{PAL}}\varphi$ if $\mathcal{M}, w \models \psi$ and $\mathcal{M}^{\psi}, w \models \varphi$ where $\mathcal{M}^{\psi}$ is the restriction of $\mathcal{M}$ to worlds where ψ holds.

Example 2. Let $\mathcal{M}$ be the model of example 1. We have $\mathcal{M}, w \models \langle p\rangle_{\text{PAL}} K_a p$. Indeed, we have $\mathcal{M}^p, w \models K_a p$ where $\mathcal{M}^p$ is



The model checking in public announcement logic is in P and the satisfiability problem in public announcement logic is PSPACE-complete [9]. A tableau proof system for public announcement logic is provided in [3].

3 Language and models

3.1 Language

Again, $\mathcal{P}$ is a countable infinite set of atomic propositions, and AGT is a finite set of agents.

Definition 3 (Syntax). *The syntax for the logic is as follows:*

$$\varphi ::= p \mid (\varphi \wedge \varphi) \mid \neg\varphi \mid K_i\varphi \mid \langle\varphi\rangle\varphi \mid \bigcirc_i \varphi,$$

where $p \in \mathcal{P}$ *and* $i \in AGT$*.*

We use $\mathcal{L}$ to denote the set of all formulas. The intuitive meaning of the last three operators is the following: $K_i\varphi$ means that agent i knows φ, $\langle\psi\rangle\varphi$ means that ψ is true and after ψ has been put on the public channel, φ holds, and $\bigcirc_i\varphi$ means that agent i has a message to read, and after he has read it, φ holds. We classically define $(\varphi \vee \psi) ::= \neg(\neg\varphi \wedge \neg\psi)$, $(\varphi \rightarrow \psi) ::= (\neg\varphi \vee \psi)$, the dual of the knowledge operator: $\hat{K}_i\varphi ::= \neg K_i\neg\varphi$, meaning that agent i considers φ possible, and 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.

Note that in (synchronous) public announcement logic (see definition 1), the emission and the reception of a formula ψ is mixed in the operator $\langle\psi\rangle_{\text{PAL}}$, because in this setting, emission and reception occur simultaneously. In the asynchronous version of announcement logic we propose here (see Definition 3), the emission of ψ is is represented by $\langle\psi\rangle$ and the reception of a message by Agent i is represented by the operator $\bigcirc_i$. Note that not only can emission and reception occur at different times, but also different agents may receive the same message at different times.

3.2 Models

The models on which our logic is interpreted represent situations obtained by announcements being made in an initial epistemic model, with agents asynchronously receiving these announcements. We now define initial epistemic models, sequences of announcements, and a third notion that we call *cuts*, representing the announcements each agent has received at the current time.

Initial Kripke model An *initial model* is given as a Kripke model $\mathcal{M} = (W, \{\rightarrow_i\}_{i\in AGT}, \Pi)$, as defined in Definition 2. An initial model represents the initial static situation before any announcements are made.

Sequences of 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 an *announcement protocol* to be a multiset of formulas in our language, 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[3]' twice. We let the proposition $\heartsuit_i$ mean "agent i has a heart card," and define the announcement protocol to be:

$$\{\{\heartsuit_a \ , \quad \hat{K}_a\heartsuit_b \wedge \hat{K}_a\neg\heartsuit_b \ , \quad \hat{K}_a\heartsuit_b \wedge \hat{K}_a\neg\heartsuit_b\}\}.$$

[3] that is 'a considers $\heartsuit_b$ possible *and* considers $\neg\heartsuit_b$ possible'

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 let $|\sigma| = \sum_{i=1}^{k} |\varphi_i|$. 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.

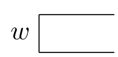
States We can now define the models of our logic. Let $\mathcal{M}$ be an initial model and $\mathcal{A}$ an announcement protocol. We define the *asynchronous model* $\mathcal{M} \otimes \mathcal{A} = (\mathcal{S}, \{R_i\}_{i \in AGT})$, where each R_i is a *pre-accessibility relation*, which we define in Section 3.2, and $\mathcal{S}$ is a set of *states* defined as follows:

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

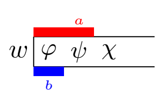
States are also denoted S, S', etc.

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 $i \in AGT$, $c(i)$ is the number of announcements of σ that Agent i has received so far. Given two cuts c and c', we write $c < c'$ if for all i, $c(i) \leq c'(i)$ and there exists j such that $c(j) < c'(j)$.

Example 4. 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 5. Consider the state $S = (w, [\varphi, \psi, \chi], c)$ where $c(a) = 2$ and $c(b) = 1$. S is the state representing that in initial world w, the sequence $[\varphi, \psi, \chi]$ of formulas has been announced and is now in the public channel, 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 $S = (w, [\varphi, \psi, \chi], \begin{smallmatrix} a \mapsto 2 \\ b \mapsto 1 \end{smallmatrix})$.

The definition of $\mathcal{S}$ allows for all combinations of worlds, sequences of announcements, and cuts. This definition is an over-approximation of the set of states we want to consider: indeed, 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. Then, the state $(w, [p], \mathbf{0})$ is intuitively inconsistent because as the formula p is not true in w, it cannot have been announced. This notion of inconsistency is the source of the circularity problem. Indeed, to define 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 related to the current one, which requires us to define which states are consistent, hence the circularity (see Section 4).

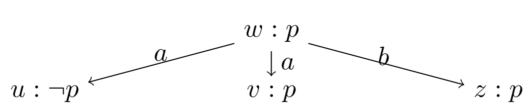
Pre-accessibility relation definition We now define, for each agent, a *pre-accessibility relation* that does not yet take consistency into account, but is only based on the agents' accessibility relations on the initial model and the current cut. This is the first step toward the final definition of the agents' accessibiliy relations, which is presented below.

Definition 4. *The* pre-accessibility relation *for Agent i, written R_i, is defined as follows: given $S = (w, \sigma, c)$ and $S' = (w', \sigma', c')$, we have SR_iS' if:*

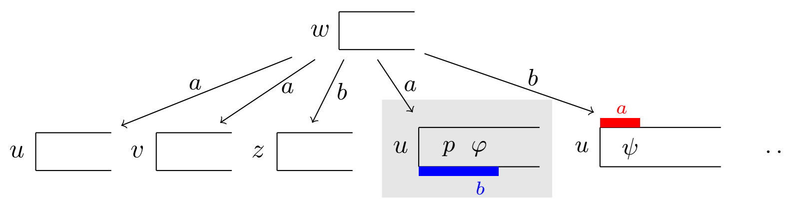
1. $w \rightarrow_i w'$, *and*
2. $c(i) = c'(i)$ *and* $\sigma|_{c(i)} = \sigma'|_{c'(i)}$

The first clause is obvious. The second clause says that Agent i 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. However, the sequence of messages she has not yet received may be longer or shorter in a related state. Also, as she has no information about what messages the other agents have received, we do not put any constraints on $c'(j)$ if $j \neq i$.

Example 6. Here we give an example of an inconsistent state, to show why further refinement of the above-defined models is necessary. 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 announcements have been made, as well as copies of u where two different sequences of announcements have been made, and received in one state by b and in a different state by a. Of course, the entire model $\mathcal{M} \otimes \mathcal{A}$ is infinite so we do not depict all the states here. The grey state shown in the asynchronous model is not consistent because p has been announced although p is not true in u.



4 Semantics: the circularity problem

Figures 2 and 3 show a naive attempt to define the truth conditions, which we explain in Section 4.1. Unfortunately, this leads to a circularity problem which we detail in Section 4.2. We show how to solve this problem for restricted cases in Sections 4.3 and 4.4.

$$
\begin{aligned}
&(w, \epsilon, \mathbf{0}) \models \checkmark \\
&(w, \sigma, c) \models \checkmark \text{ if there is } c' < c \text{ s.t. } (w, \sigma, c') \models \checkmark, \text{ or} \\
&\qquad\qquad\quad \sigma = \sigma'{::}\psi, (w, \sigma', c) \models \checkmark \text{ and} (w, \sigma', c) \models \psi
\end{aligned}
$$

Fig. 2. Truth conditions for consistency

$$
\begin{array}{lll}
(w, \sigma, c) \models p & \text{if} & p \in \Pi(w) \\
(w, \sigma, c) \models \varphi_1 \wedge \varphi_2 & \text{if} & (w, \sigma, c) \models \varphi_1 \text{ and } (w, \sigma, c) \models \varphi_2 \\
(w, \sigma, c) \models \neg\varphi & \text{if} & (w, \sigma, c) \not\models \varphi \\
(w, \sigma, c) \models K_i\varphi & \text{if} & \text{for all } S' \text{ s.t. } (w, \sigma, c) R_i S' \text{ and } S' \models \checkmark,\ S' \models \varphi \\
(w, \sigma, c) \models \langle\psi\rangle\varphi & \text{if} & \sigma{::}\psi \in \mathsf{Seq}(\mathcal{A}),\quad (w, \sigma, c) \models \psi \text{ and } (w, \sigma{::}\psi, c) \models \varphi \\
(w, \sigma, c) \models \bigcirc_i\varphi & \text{if} & c(i) < |\sigma| \text{ and } (w, \sigma, c^{+i}) \models \varphi \\
& & \quad \text{where } c^{+i}(j) = \begin{cases} c(j) & \text{if } j \neq i \\ c(j) + 1 & \text{if j=i} \end{cases}
\end{array}
$$

Fig. 3. Truth conditions for formulas

4.1 Intuitions

The pre-accessibility relation takes into account states that are not actually consistent, because they contain announcements that were not true at the time they were made. The intuitive meaning of $(w, \sigma, c) \models \checkmark$ is that the state (w, σ, c) is consistent, that is, all announcements were true when they were made. Figure 2 is an attempt to define this concept formally. The first clause is obvious: the initial state where no announcements have 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') and then 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, increasing the history from σ' to $\sigma'.\psi$.

In Figure 3, the first three clauses are straightforward. The fourth clause says that Agent i knows φ if φ holds in all consistent states that are indistinguishable to her. 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 $\bigcirc_i\varphi$ holds if Agent i has at least one unread announcement in the channel, and φ holds after she reads the first message.

4.2 Circularity

Let us consider the following example, where $AGT = \{a\}$. Let the initial model be $\mathcal{M} = (W, \rightarrow_a, \Pi)$ where $W = \{w\}$, $\rightarrow_a = \{(w,w)\}$ and $\Pi(w) = \emptyset$. Let the announcement protocol be $\mathcal{A} = \{\{K_a p\}\}$. According to Figure 2, 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.

4.3 When the initial model is a finite tree

If we assume the initial model $\mathcal{M} = (W, \{\rightarrow_i\}_{i \in AGT}, \Pi)$ to be such that W is finite and $\bigcup_i \rightarrow_i$ makes a **finite** tree over W, then the circularity problem can be avoided. Indeed, 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 $\mathcal{L}$ or the symbol $\checkmark$, the idea being that a tuple (w, σ, c, φ) means '$w, \sigma, c \models \varphi$'.

Definition 5. *The order $\prec$ is defined as follows:*
$(w, \sigma, c, \varphi) \prec (w', \sigma', c', \varphi')$ *if either*

① *w is in the subtree 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 easy to see that $\prec$ is a well-founded order, and with this order figures 2 and 3 together form a well-founded inductive definition of consistency and semantics of our language.

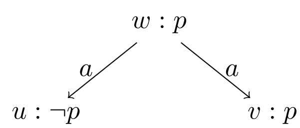
We detail the non-trivial cases. For the second clause of Figure 2, observe that by Point ③ of Definition 5, if $c' < c$ then $(w, \sigma, c', \checkmark) \prec (w, \sigma, c, \checkmark)$, and for all w, σ', c and ψ, by Point ② of Definition 5, we have $(w, \sigma', c, \psi) \prec (w, \sigma' :: \psi, c, \checkmark)$.

For the fourth clause of Figure 3, by Point ① of Definition 5 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 fifth clause of Figure 3, by Point (2) of Definition 5 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 $|\langle\psi\rangle\varphi| = 1 + |\psi| + |\varphi|$).

The following simple example illustrates how our semantics works, and how it indeed captures the intuitions behind the operators.

Example 7. Suppose that we have only one agent a. Let us consider the following initial model $\mathcal{M}$:



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 has received this announcement, it holds that 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 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 4 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 just have to show that each such state either is inconsistent or verifies p.

First, for $w' = u$. According to the first clause of Figure 3, we have that $(u, \epsilon, \mathbf{0}) \not\models p$, and by the second clause of Figure 2 it follows that $(u, [p], \mathbf{0}) \not\models \checkmark$, from which it 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 of Figure 3, we have 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: unfolding models and cutting at level l of the obtained trees amounts to assuming that agents cannot reason about deeper nesting of knowledge. This approach is similar to the well known idea of *bounded rationality*, [7], 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 this representation really captures, and to develop precise results about which formulas we are able to correctly evaluate using this method of approximation.

4.4 Announcing existential formulas

Now, we again allow the initial model to be arbitrary. In particular, we may use an initial model whose underlying frame is $KD45$ (relations are serial, transitive and Euclidean) or $S5$ (relations are equivalence relations) (see [5]). But we restrict the announcement protocol to the existential fragment[4], generated by the following rule:

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \hat{K}_i \varphi \mid \bigcirc_i \varphi \mid \langle \varphi \rangle \varphi$$

where $p \in \mathcal{P}$ and $i \in AGT$. Formulas of the existential fragment are called *existential formulas*. If an announcement protocol contains only formulas of the existential fragment, we call it an *existential announcement protocol*. For instance, Example 3 consists of an existential announcement protocol.

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

Let us fix an initial model $\mathcal{M} = (W, \{\rightarrow_i\}_{i \in AGT}, \Pi)$ 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 the symbol $\checkmark$, which is true only at consistent states. Observe that $(\mathcal{P}(B), \subseteq)$ forms a complete lattice. We now consider the function $f : \mathcal{P}(B) \to \mathcal{P}(B)$ 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 figures 2 and 3 one time. 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 in Figure 4. Every other line of Figure 4 similarly follows from one line 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 [12], 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.

[4] The terminology 'existential fragment' is used in the model checking community [1], because the operators $\hat{K}_i$, $\bigcirc_i$ and $\langle \varphi \rangle$ are existential. For instance, we will require that $(w, \sigma, c) \models \hat{K}_i \varphi$ iff *there exists* S' s.t. $(w, \sigma, c) R_i S'$, $S' \models \checkmark$, and $S' \models \varphi$. When these operators are not in the scope of a negation, only existential quantification needs to be used in the semantic interpretation of formulas.

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

Fig. 4. Function applying one step of the truth conditions

Remark 1. If announcements of the form $K_i\varphi$ were allowed, then we would have to add the clause

$$
\left\{(w,\sigma,c,K_i\varphi) \mid \begin{array}{l}\text{for all } (w',\sigma',c') \text{ such that } (w,\sigma,c)R_i(w',\sigma',c'),\\ \text{either } (w',\sigma',c',\checkmark)\notin\Gamma \;\text{ or }\; (w',\sigma',c',\varphi)\in\Gamma\end{array}\right\}
$$

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.

5 Validities

We say that a formula φ is *valid* if for every initial model $\mathcal{M}$ and every announcement protocol $\mathcal{A}$, such that either $\mathcal{M}$ is a finite tree or $\mathcal{A}$ is an existential announcement protocol, and for every consistent state $S\in\mathcal{M}\otimes\mathcal{A}$,[5] we have $\mathcal{M}\otimes\mathcal{A},S\models\varphi$. We write $\models\varphi$ to express that φ is valid. As usual, we use $[\varphi]\psi$ as shorthand for $\neg\langle\varphi\rangle\neg\psi$.

Proposition 1. *We have:*

1. $\models \bigcirc_1\bigcirc_2\varphi \leftrightarrow \bigcirc_2\bigcirc_1\varphi$

[5] Recall definition of $\mathcal{M}\otimes\mathcal{A}$ from 3.2.

2. $\models \bigcirc_1 \top \to (\bigcirc_1 \varphi \leftrightarrow \neg \bigcirc_1 \neg\varphi)$
3. $\models \neg \bigcirc_1 \top \to [\varphi] \bigcirc_1 K_1\varphi$, *where* φ *is a propositional formula* [6]

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

Suppose that we have $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \bigcirc_1 \bigcirc_2 \varphi$. By Figure 3, this means that $c(1) < |\sigma|$ and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c^{+1}) \models \bigcirc_2\varphi$, and the latter implies that $c^{+1}(2) < |\sigma|$ and $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, \left(c^{+1}\right)^{+2}) \models \varphi$. Now, because $\left(c^{+1}\right)^{+2} = \left(c^{+2}\right)^{+1}$, we get that $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c^{+2}) \models \bigcirc_1\varphi$, and therefore $\mathcal{M} \otimes \mathcal{A}, (w, \sigma, c) \models \bigcirc_2 \bigcirc_1 \varphi$. The proof for the other direction is symmetric.

Let us comment on the above validities. Validity 1 says that it is possible to permute the order of agents that receive next messages in their respective queues. Validity 2 says that if an agent has a message to read, then reading the message is a deterministic operation. Validity 3 says that if an agent has no pending messages and some propositional formula[7] is announced, then after reading his next message, the agent will know that formula.

We also establish the following proposition, which essentially says that in the case where all sequences of announcements are possible, if all the $\bigcirc_i$ 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 other 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 it is considered, no matter whether it was actually the announced formula or not.

In the following, either let $\mathcal{M}$ be a finite tree and $\mathcal{A}_U$ the universal announcement protocol containing every formula with infinite cardinality, or let $\mathcal{M}$ be an arbitrary initial model and $\mathcal{A}_U$ the announcement protocol containing every formula in the existential fragment with cardinality infinity.

Proposition 2. *Let* φ *be a formula in* $\mathcal{L}$, *in which every* $\bigcirc_i$ *is under the scope of some* K_j. *For every initial model* $\mathcal{M}$ *and consistent state* $S = (w, \sigma, c) \in \mathcal{M} \otimes \mathcal{A}_U$, *for every* $\psi \in \mathcal{A}_U$, *we have* $\mathcal{M} \otimes \mathcal{A}_U, S \models \langle\psi\rangle\varphi \leftrightarrow \psi \wedge \varphi$.

This result follows immediately from the following lemma:

Lemma 1. *Let* φ *be a formula in* $\mathcal{L}$, *in which every* $\bigcirc_i$ *is under the scope of some* K_j. *For every initial model* $\mathcal{M}$ *and for every consistent state* $S = (w, \sigma, c) \in \mathcal{M} \otimes \mathcal{A}_U$, *for every* $\psi \in \mathcal{A}_U$ *such that* $(w, \sigma{::}\psi, c)$ *is consistent, we have* $\mathcal{M} \otimes \mathcal{A}_U, (w, \sigma{::}\psi, c) \models \varphi$ *iff* $\mathcal{M} \otimes \mathcal{A}_U, (w, \sigma, c) \models \varphi$.

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

Case $\varphi = K_a\varphi'$:

It is enough to observe that: $\{S \mid (w, \sigma{::}\psi, c)R_aS\} = \{S \mid (w, \sigma, c)R_aS\}$.

Case $\varphi = \langle\varphi_1\rangle\varphi_2$:

[6] A propositional formula is any formula without modalities, i.e. no occurrences of K_i, $\langle\varphi\rangle$, or $\bigcirc$.

[7] We restrict to propositional formulas in order to avoid Moore's paradox [14].

By definition of $\mathcal{A}_U$, $\sigma{::}\varphi_1 \in \mathsf{Seq}(\mathcal{A}_U)$. We therefore have $(w, \sigma{::}\psi, c) \models \langle\varphi_1\rangle\varphi_2$ iff $(w, \sigma{::}\psi, c) \models \varphi_1$ and $(w, \sigma{::}\psi{::}\varphi_1, c) \models \varphi_2$. By induction hypothesis, this is equivalent to $(w, \sigma, c) \models \varphi_1$ and $(w, \sigma{::}\psi, c) \models \varphi_2$. Again by induction hypothesis, the latter is equivalent to $(w, \sigma, c) \models \varphi_2$, which is in turn equivalent to $(w, \sigma{::}\varphi_1, c) \models \varphi_2$ (observe that $(w, \sigma{::}\varphi_1, c)$ is consistent as $(w, \sigma, c) \models \varphi_1$). We finally obtain that $(w, \sigma{::}\psi, c) \models \langle\varphi_1\rangle\varphi_2$ iff $(w, \sigma, c) \models \varphi_1$ and $(w, \sigma{::}\varphi_1, c) \models \varphi_2$, that is $(w, \sigma, c) \models \langle\varphi_1\rangle\varphi_2$.

Finally, the case $\varphi = \bigcirc_i\varphi'$ is not possible as $\bigcirc_i$ is not under the scope of any K_j.

In our framework, the behavior of the public channel is common knowledge. For instance, let us consider a situation with two agents, 1 and 2, where Agent 1 has read all the announced messages. Now, assume that p is announced, thus put in the queue, and Agent 1 reads it. Agent 1 now knows p, but she also knows that if Agent 2 has read all the announced messages (and in particular the last one, which is p), then Agent 2 also knows p. In some sense, it means that initially Agent 1 knows that Agent 2 will receive the same messages as herself. This is reflected in the following validity: $\bigcirc_1\bot \to [p!]\bigcirc_1 K_1(p \wedge (\bigcirc_2\bot \to K_2 p))$.

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,[8] $(\mathcal{M}, w)$, 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.

6.1 Propositional announcements

In this section, we suppose that formulas in $\mathcal{A}$ are propositional (which is a particular case of existential announcements). We consider the model checking problem for asynchronous announcement logic where inputs $\mathcal{M}, w, \mathcal{A}, \varphi_0$ are such that $\mathcal{A}$ only contains propositional formulas. This problem is called the *model checking problem for propositional protocols*.

Theorem 1. *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 $P(\psi)$ holds:

[8] A pointed model is a model with a specified state.

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

This establishes soundness and completeness 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 (*☞) 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.

```
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 ∈ R_a(w), σ' ∈ Seq(A), c' on σ' do
        if c'(i) = c(i) and σ'[1..c(i)] = σ[1..c(i)] 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 ◯_i ψ: return c(i) < |σ| and mc (M, A, w, σ, c^{+i}, ψ)
```

Fig. 5. Model checking algorithm

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

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 3. *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 (∗☞) 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
```

Soundness and completeness are proven by induction on inputs using the order $\prec$ defined in Section 4.3.

Concerning the complexity, the argument given in the proof of Theorem 1 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 $|\sigma| + |\varphi| + \sum_{i \in AGT} c(i)$ is strictly decreasing, where $|\varphi|$ is the length of φ and if $\sigma = [\varphi_1, \dots, \varphi_k]$ 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 $c(i) \leq |\mathcal{A}|$. Thus, the quantity $|\sigma| + |\varphi| + \sum_{i \in AGT} c(i)$ is bounded by $(|AGT| + 2)|\mathcal{A}| + |\varphi_0|$. Therefore, the number of nested calls to mc is bounded by $depth(\mathcal{M}, w) \times ((|AGT| + 2)|\mathcal{A}| + |\varphi_0|)$. So the algorithm requires 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 4.4. 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 exponential size 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, in which checking the consistency of a state (w, σ, c), (∗☞), is done by checking whether $(w, \sigma, c, \checkmark) \in \Gamma^*$. The algorithm mc also requires exponential time. To sum up:

Theorem 4. *The model checking problem where the announcements are existential is in EXPTIME.*

7 Related work

As far as we know, there has not been much work on the relationship between knowledge, announcements and asynchronicity. In [4], asynchronicity in dynamic epistemic logic is studied, but with the idea of synchronicity being that all agents observe a universal clock, whereas our notion of synchronicity is that all agents receive messages at the same time, immediately when they are sent. A logic dealing with knowledge and asynchronicity is also developed in [10], but in this setting, messages do not have logical content: they are atomic propositions, and it is impossible, for example, to make an announcement about knowledge or about the effect of another announcement.

Arbitrary public announcement logic (APAL) [2] has some similarity 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 the knowledge operator considers any possible future sequence of announcements that follows the protocol, which is a similar 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 ([6], [13]).

The Knaster-Tarski theorem is often used to define the denotational semantics of programming languages [15] in the same spirit as our definition of consistency when announcements are existential. We also note that our definition of asynchronous models $\mathcal{M} \otimes \mathcal{A}$, especially the notion of cuts, is in the spirit of [8].

8 Future work

This work constitutes a first attempt to provide an epistemic logic for reasoning about asynchronous announcements. In the future, first, we would like to overcome the circularity problem, and hence define the semantics, for the most general case (removing the finite tree and existential conditions), and provide model checking algorithms in these cases. One approach to this problem may be using coinduction to define the set of consistent states. Once we have defined the semantics for the general case, we plan to provide a complete axiomatization.

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 saying that "After Agent a broadcasts that φ holds, 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, we would like to model not only asynchronous broadcast 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 [14].

Finally, it would be interesting to add temporal operators to our language, in order to express such things as "After p is announced and Agent 1 receives it, *eventually* she will know that Agent 2 knows p" (assuming that agents are forced to eventually read announcements).

References

1. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
2. Philippe Balbiani, Alexandru Baltag, Hans P. van Ditmarsch, Andreas Herzig, T. Hoshi, and Tiago De Lima. What can we achieve by arbitrary announcements?: A dynamic take on Fitch's knowability. In *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2007), Brussels, Belgium, June 25-27, 2007*, pages 42–51, 2007.
3. Philippe Balbiani, Hans van Ditmarsch, Andreas Herzig, and Tiago De Lima. Tableaux for public announcement logic. *J. Log. Comput.*, 20(1):55–76, 2010.
4. Cédric Dégremont, Benedikt Löwe, and Andreas Witzel. The synchronicity of dynamic epistemic logic. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2011), Groningen, The Netherlands, July 12-14, 2011*, pages 145–152, 2011.
5. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about knowledge*. The MIT Press, 2004.
6. Tim French and Hans P. van Ditmarsch. Undecidability for arbitrary public announcement logic. In *Advances in Modal Logic 7, papers from the seventh conference on "Advances in Modal Logic," held in Nancy, France, 9-12 September 2008*, pages 23–42, 2008.
7. Bryan D. Jones. Bounded rationality. In *Annual Review of Political Science*, pages 2–297, 1999.
8. Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, July 1978.
9. Carsten Lutz. Complexity and succinctness of public announcement logic. In *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, May 8-12, 2006*, pages 137–143, 2006.
10. Prakash Panangaden and Kim Taylor. Concurrent common knowledge: Defining agreement for asynchronous systems. *Distributed Computing*, 6(2):73–93, 1992.
11. Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
12. Alfred Tarski et al. A lattice-theoretical fixpoint theorem and its applications. *Pacific journal of Mathematics*, 5(2):285–309, 1955.
13. Hans van Ditmarsch, Tim French, and James Hales. Positive announcements (under submission).
14. Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Pieter Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.
15. Glynn Winskel. *The formal semantics of programming languages - an introduction*. Foundation of computing series. MIT Press, 1993.