

A Runs-and-Systems Semantics for Logics of Announcements*

Riccardo Pucella
College of Computer and Information Science
Northeastern University
riccardo@ccs.neu.edu

Mehrnoosh Sadrzadeh
Laboratoire Preuves Programmes et Systèmes
Université Paris Diderot - Paris 7[†]
mehrs@comlab.ox.ac.uk

Abstract

Logics of announcements are logics of knowledge to reason about agents that communicate by broadcasting interpreted messages. These logics are typically given a semantics in terms of updatable Kripke structures, which tend to be abstract. We revisit the semantics of logics of announcements and develop a concrete semantics using runs and systems. The advantage is that we can devise models that capture scenarios without having to express properties of those scenarios within the logic itself. In this concrete setting, we study honesty as well as belief in the presence of announcements that are not broadcast to all agents in a system.

1 Introduction

Logics of knowledge provide the ability to reason about knowledge of agents in situations where they interact. Roughly speaking, logics of knowledge, or epistemic logics, are logics with an operator typically written $K_i\varphi$ used to capture the notion that agent i *knows* fact φ . Such logics have found applications in philosophy, artificial intelligence, economics, game theory, and distributed computing. How knowledge is interpreted varies widely, but a common interpretation based on possible worlds goes back to Hintikka [7], where an agent knows a fact if that fact is true at all the worlds that the agent consider possible alternatives to the actual world. Kripke structures [8] are used to formalize this semantics for knowledge.

Most early uses of logics of knowledge involved reasoning about agents that communicate by exchanging uninterpreted messages. Here, by uninterpreted messages, we mean messages that are not intended to be interpreted as formulas, and therefore do not require a priori knowledge of the semantics of the logic to make sense. For instance, an agent in a system

*An abstract of this work was presented at the 8th Conference on Logic and the Foundations of Game and Decision Theory.

[†]From October 2008, EPSRC Postdoctoral Research Fellow, Computing Laboratory, University of Oxford.

sending a message containing the agent’s name is sending an uninterpreted message. In contrast, there are situations where agents interact by exchanging interpreted messages, that is, messages intended as logical formulas whose meaning depends on the semantics of the logic such as statements about their own knowledge. To reason about such scenarios, *logics of announcements* were developed [14, 5, 2, 1, 16]. Logics of announcements can be seen as a combination of epistemic logic and dynamic logic [6]. Dynamic logic takes as a basic logical operator the operator $[\alpha]\varphi$, interpreted as φ holding after actions α have been performed. Logics of announcements specialize the actions to dynamic logic by permitted agents to *announce* statements, including statements about their own knowledge. Logics of announcements are traditionally given a semantics in terms of updating Kripke structures. Roughly speaking, a formula such as “ φ is true after announcing that agent i knows ψ ” is true at a state of some Kripke structure M if φ is true in the new Kripke structure obtained by updating M with the result of announcing $K_i\psi$. There are several difficulties with this approach to reasoning about announcements. How to update Kripke structures after a particular announcement is not immediately clear, for instance. But from a practical perspective, one difficulty is that traditional models for logics of announcements are quite abstract. It is usually nontrivial to take a given scenario and model it for analysis.

In this paper, we revisit the semantics of logics of announcements, and focus on giving a concrete independently motivated semantics to such logics. The idea is simple: our models, based on the runs-and-systems framework of Fagin et al. [4, 11], describe a concrete scenario to be analyzed in terms of the possible evolutions of a situation, where the possible evolutions include announcements made by the various agents in the scenario. The models are purely descriptive. The logic is used to state properties of the scenario via an intuitive notion of truth. This approach is in contrast to that used in DEL [2], for instance, where modeling a scenario is done completely within the logic, by stating properties of the scenario within the logic itself as axioms, and reasoning about what can be derived from those axioms. As we argue, this sometimes raises the question of how to ensure that the scenario has been accurately captured in the logic. Such a question is less problematic to answer with a concrete semantics, because the model of a scenario is a description that is very close to the way in which the scenario is informally described.

Our models are quite simple, and they already highlight several subtle issues with reasoning about announcements. For instance, synchrony, that is, whether agents know the time, impacts what knowledge agents gain after hearing an announcement, which is especially relevant when agents only announce statements that they know to be true. Another subtlety that our framework highlights is the role of *strategies* to understand how to model belief. Belief arises naturally when considering announcements that are not broadcast to all agents, but instead are heard by only a subset of agents. Belief, according to some approaches, can be taken to be what an agent knows if she assumes that not having heard an announcement means that that announcement was never made. Making this precise, we argue, can only be done if agents already have an idea about what strategies other agents in the systems are using to choose their actions.

This paper is structured as follows. In §2, we introduce the syntax and semantics of the logic of announcement we use. In §3, we try to make precise the notion of honesty, where agents only announce true statements. In §4, we compare our approach with traditional logics of announcements. In §5, we turn to the question of announcements that are not

broadcasted to all agents, and try to develop reasonable notions of belief in this context. We conclude in §6.

2 A Logic of Announcements

We start by defining a class of structures that correspond somewhat closely to execution of protocols or programs. We use the runs and systems framework of Fagin et al [4], which takes as models sets of execution traces.

An interpreted system is a pair (\mathcal{R}, π) , where \mathcal{R} is a set of runs, each run representing a possible execution of the system, and π is an interpretation for the primitive positions (or atoms). A run is a map from time to global states, where $r(m)$ is the global state in run r at time m . (For simplicity, we take time to range over the natural numbers.) We call a pair (r, m) a point of r . Thus, each point corresponds to a global state of the system.

A global state is of the form (s_e, s_1, \dots, s_n) , where s_e is the local state of the environment, and s_1, \dots, s_n are local states for each agent. If $r(m) = (s_e, s_1, \dots, s_n)$, we write $r_e(m)$ for s_e and $r_i(m)$ for s_i . Intuitively, the local state for an agent records the observations that the agent has made. Here, we consider observations to be announcements that the agent has received. We also record in the local state possible initial observations made by the agent. For instance, in the muddy children puzzle, initial observations for an agent include which other children are dirty. The local state of the environment records information which is not available to the agents. For example, in a coin-tossing scenario, the result of the coin toss and also the sequence of all announcements that have been made can be stored in the local state of environment. The announcements that an agent has received will be a subsequence of this sequence of announcements.

An announcement consists of a message θ (in some language that we describe below, but is irrelevant for now), with an associated announcer a and a group β of recipients. We require that the announcer a be in β . We represent an announcement by the event $\theta!_{\beta}^a$ appearing in the local state of agents receiving the announcement, that is, agents in β . As we will see below, for bookkeeping purposes, we record all announcements in the local state of the environment as well. An announcement is *public* if its associated group of recipients is $\{1, \dots, n\}$, otherwise, it is *private*.

Building announcements out of a language raises subtle issues, including whether we want to distinguish announcements that are logically equivalent. For instance, announcement $(p \wedge q)!_{\beta}$ is often taken to be the same announcement as $(q \wedge p)!_{\beta}$. However, there are situations where “how” the announcement is presented is relevant. After all, an announcement is a message, and concretely, the message has a representation in the system. For instance, an announcement may be represented as a bit-string sent over a network link, and an agent may be able to distinguish two different bit-strings corresponding to two different presentations of the same announcement. If all one cares about is the interpretation of the announcement, as is the case in several announcement logics where announcements are taken to be sets of states, then there is no distinction between announcing $p \wedge q$ and announcing $q \wedge p$. But this view is somewhat limiting. It turns out that the results of this paper do not depend on how we resolve this choice. For simplicity, we take two announcements to be the same when they are structurally equal, with the understanding that we could impose

an arbitrary equivalence relation over announcements to capture announcements we want to consider equivalent (such as logical equivalence).

We want to interpret knowledge in such systems. We take a standard Hintikka semantics for knowledge, where an agent knows a fact in some global state s if that fact is true at all the global states she considers possible alternatives to state s . To do so, we define, for each agent i , a relation over points of the system capturing which points agent i cannot distinguish. Intuitively, two points are indistinguishable to agent i if i has the same local state at both points. We define $(r, m) \sim_i (r', m')$ if $r_i(m) = r'_i(m')$. Note that this makes \sim_i an equivalence relation. We also define the reflexive transitive closure of \sim_i relative to a subset β of agents, by taking $(r, m) \sim_\beta^* (r', m')$ if there exists a sequence i_1, \dots, i_k of agents in β with $(r, m) \sim_{i_1} (r_1, m_1) \sim_{i_2} \dots \sim_{i_{k-1}} (r_{k-1}, m_{k-1}) \sim_{i_k} (r', m')$.

The above description of systems is extremely general, suitable for defining a broad class of multiagent systems, and in fact more general than we want for defining announcement systems. In particular, we assume a specific structure to the local state of agents and the environment in announcement systems. We assume that the local state of an agent (including the environment) contains an initial local state ι and a list of sets of observations S_j , that is $s_i = \langle \iota, S_1, \dots, S_m \rangle$ where ι is an initial local state and each S_j is a set of observations, for $i \in \{e, 1, \dots, n\}$. We write $\theta!_\beta^a \in r_i(m)$ if $\theta!_\beta^a$ appears in one of the sets of observations in the local state of agent i , including the environment, that is, if $\theta!_\beta^a \in S_j$ for $1 \leq j \leq m$ and $r_i(m) = \langle \iota, S_1, \dots, S_m \rangle$.

The following properties of a run r capture these properties and characterize systems that interest us:

- R1. Facts do not change during the run. Thus, for all primitive propositions p and times $m, m' \geq 0$, we have $\pi(r, m)(p) = \mathbf{true}$ if and only if $\pi(r, m')(p) = \mathbf{true}$.
- R2. Announcements are accumulated in the local state of the environment. Thus, for all times $m \geq 0$, if $r_e(m) = \langle \iota, S_1, \dots, S_m \rangle$ where ι is an initial state and S_1, \dots, S_m are sets of announcements, then $r_e(m+1) = \langle \iota, S_1, \dots, S_m, S_{m+1} \rangle$, where S_{m+1} is a (possibly empty) set of announcements.
- R3. An announcement $\theta!_\beta^a$ appears in the local state of the environment if and only if it appears in the local state of every agent in β . Thus, for all times $m \geq 0$, announcements $\theta!_\beta^a$ and agents $i \in \beta$, we have $\theta!_\beta^a \in r_e(m)$ (as defined above) if and only if $\theta!_\beta^a \in r_i(m)$.
- R4. The local state for agent i at every point is obtained from the local state of the environment at that point, filtered to keep only announcements pertaining to i . Thus, for all times $m \geq 0$ and agents $i \in \beta$, if $r_e(m) = \langle \iota, S_1, \dots, S_m \rangle$ where ι is an initial state and S_1, \dots, S_m are sets of announcements, then $r_i(m) = \langle \iota', S'_1, \dots, S'_m \rangle$ where $S'_j = \{\theta!_\beta^a \in S_j \mid i \in \beta\}$ for all $j \in \{1, \dots, m\}$.

Properties R2 and R3 together ensure that agents have *perfect recall*, that is, they do not forget announcements that they have heard in the past. R3 imposes no constraint on how the local state of agents records announcements, beyond the fact that all pertinent announcements that have been made appear in the local state. In particular, from a round to the next, if no new announcement is heard by the agent, then the local state of the agent

need not change. In other words, an agent need not be aware of time passing if she has not heard any new announcements. Thus, R3 allows for asynchronous behavior [4]. Property R4 forces the representation of agent local states to record the number of rounds and the announcements made at each round. Thus, in a precise sense, if every run has property R4 in a system, then agents know the current time in a system—at a point (r, m) , they will not consider a point with a different time m' possible.

Definition 2.1. *An announcement system is a system in which every run satisfies R1–3. An announcement system is synchronous if every run additionally satisfies R4.*

All our work until now has remained completely independent of the language of announcements. We write announcements in an epistemic propositional language. Start with a set Φ_0 of primitive propositions, representing the basic facts that we care about, and form the language \mathcal{L}_n^K of announcements over n agents by closing Φ_0 under \wedge , \neg , and the modal operator K_i . As usual, we take $\varphi \vee \psi$ to be an abbreviation for $\neg(\neg\varphi \wedge \neg\psi)$, and $\varphi \Rightarrow \psi$ an abbreviation for $\neg\varphi \vee \psi$. We use θ to range over formulas in \mathcal{L}_n^K .

We define a simple logic $\mathcal{L}_n^{K,A}$ for reasoning about announcements, which includes \mathcal{L}_n^K as a sublanguage. The logic has essentially only epistemic operators. Again, we start with the set Φ_0 of primitive propositions. We consider two additional kinds of propositions specifically for dealing with announcements: propositions of the form $\text{ann}_j(\theta, \beta)$, read “agent j has just announced θ to agents in β ”, and $\text{announced}(j, \theta, \beta)$, read “agent j announced θ to agents in β at some point in the past.” There is some flexibility in the choice of propositions for the logic, as usual, and making a distinction between current announcements and past announcements turns out to be convenient for some classes of specifications. Formulas of $\mathcal{L}_n^{K,A}$ include $\varphi_1 \wedge \varphi_2$, $\neg\varphi$, $K_i\varphi$ (read “agent i knows φ ”), and $C_\beta\varphi$ (read “it is common knowledge among the agents in β that φ ”).

$$\varphi, \psi ::= p \mid \text{ann}_a(\theta, \beta) \mid \text{announced}(a, \theta, \beta) \mid \varphi \wedge \psi \mid \neg\varphi \mid K_i\varphi \mid C_\beta\varphi$$

Again, we take $\varphi \vee \psi$ to be an abbreviation for $\neg(\neg\varphi \wedge \neg\psi)$, and $\varphi \Rightarrow \psi$ an abbreviation for $\neg\varphi \vee \psi$. We also write $E_G\varphi$ as an abbreviation for $\bigwedge_{i \in G} K_i\varphi$. Since \mathcal{L}_n^K is a sublanguage of $\mathcal{L}_n^{K,A}$, we consider θ as a formula of $\mathcal{L}_n^{K,A}$. Working with a two-layered syntax yields a simple logic with a straightforward semantics, avoiding problems caused by announcements such as $\text{ann}_a(\text{ann}_b(\theta, \gamma), \beta)$ —expressing, for instance, that the statement of an announcement has just been announced—which are difficult to interpret semantically. These nested announcements tend to be uncommon in applications.

We interpret $\mathcal{L}_n^{K,A}$ with respect to announcement systems and say formula φ is true at a point (r, m) of system \mathcal{A} , written $(\mathcal{A}, r, m) \models \varphi$, defined inductively as

$$\begin{aligned} (\mathcal{A}, r, m) \models p &\text{ iff } \pi(r, m)(p) = \mathbf{true} \\ (\mathcal{A}, r, m) \models \text{ann}_a(\theta, \beta) &\text{ iff } \theta!_\beta^a \in r_e(m) \text{ and } (m = 0 \text{ or } \theta!_\beta^a \notin r_e(m-1)) \\ (\mathcal{A}, r, m) \models \text{announced}(a, \theta, \beta) &\text{ iff } \theta!_\beta^a \in r_e(m) \\ (\mathcal{A}, r, m) \models \varphi_1 \wedge \varphi_2 &\text{ iff } (\mathcal{A}, r, m) \models \varphi_1 \text{ and } (\mathcal{A}, r, m) \models \varphi_2 \\ (\mathcal{A}, r, m) \models \neg\varphi &\text{ iff } (\mathcal{A}, r, m) \not\models \varphi \end{aligned}$$

$$(\mathcal{A}, r, m) \models K_i \varphi \text{ iff for all } (r', m') \sim_i (r, m), (\mathcal{A}, r', m') \models \varphi$$

$$(\mathcal{A}, r, m) \models C_\beta \varphi \text{ iff for all } (r', m') \sim_\beta^* (r, m), (\mathcal{A}, r', m') \models \varphi$$

Some remarks about this semantics are in order. First, note that $\text{ann}_i(\theta, \beta)$ is true the first time $\theta \uparrow_\beta^i$ is announced; subsequent announcements of $\theta \uparrow_\beta^i$ will make the proposition false. This is not a problem for the examples we will consider in this paper, and indeed, for most scenarios of interest. If repeated announcements are needed, one possibility is to add primitive propositions to the announcements along the lines of “this is the second such announcement”.¹ Second, there is no specific proposition expressing that an agent i has actually heard an announcement. It is already expressible in our logic: $K_i \text{announced}(j, \theta, \beta)$ captures agent i hearing announcement $\theta \uparrow_\beta^j$. We sometimes write $\text{ann}(\theta, \beta)$ for $\bigvee_{i \in \beta} \text{ann}_i(\theta, \beta)$, which can be interpreted as “some agent (necessarily in β) just announced θ to agents in β .”

We say φ is valid in \mathcal{A} if $(\mathcal{A}, r, m) \models \varphi$ for all r and m . A formula φ is valid with respect to a class of announcement systems \mathcal{C} if $\mathcal{A} \models \varphi$ for all \mathcal{A} in \mathcal{C} .

Example 2.2 (The muddy children puzzle). *Consider a scenario with $n + 1$ agents, n of which are children (numbered $1, \dots, n$), one of which is their father (numbered 0). The primitive propositions are $\text{dirty}[1], \dots, \text{dirty}[n]$, interpreted as “child i is dirty”. We can construct a system $\mathcal{A}^{\text{muddy}}$ for this scenario, with a run r^D for every subset of $\{1, \dots, n\}$ that we interpret as the children that are initially dirty. The initial local state of each child is a single bit recording whether that child is dirty or not. The interpretation π of primitive propositions is chosen so that $\text{dirty}[i]$ is true in a run exactly when the bit of agent i in the initial local state is set. In the first round, if at least one child is dirty, the father announces to everyone that there is a dirty child:*

$$\theta_1 \triangleq \left(\bigvee_{i=1}^n \text{dirty}[i] \right) \uparrow_{\{0,1,\dots,n\}}^0$$

For the following $k - 1$ rounds, where k is the number of dirty children in the run, the father announces at round j :

$$\theta_j \triangleq \left(\bigwedge_{i=1}^n \neg K_i \text{dirty}[i] \wedge \neg K_i \neg \text{dirty}[i] \right) \uparrow_{\{0,1,\dots,n\}}^0$$

Note that the run r^\emptyset corresponding to the case where there are no dirty children has no announcement.

Let \mathcal{C}^{ann} be the class of all announcement systems. Let \mathcal{C}^{syn} be the class of all synchronous announcement systems.

We can axiomatize reasoning in announcement systems as follows. The first set of axioms concerns propositional reasoning:

¹Another possibility is to augment announcements in local state of agents with tags to differentiate the same textual announcement made at different times. A cursory look at our results reveals they would not be affected by this change.

Prop. Any instance of a valid formula of propositional logic

MP. From φ and $\varphi \Rightarrow \psi$ infer ψ

The second set of axioms concerns reasoning about knowledge and common knowledge, and is the usual S5 semantics:

K1. $K_i\varphi \wedge K_i(\varphi \Rightarrow \psi) \Rightarrow K_i(\psi)$

K2. $K_i\varphi \Rightarrow \varphi$

K3. $K_i\varphi \Rightarrow K_iK_i\varphi$

K4. $\neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$

K5. From φ infer $K_i\varphi$

The following axioms concern common knowledge:

C1. $C_G\varphi \Rightarrow E_G(\varphi \wedge C_G\varphi)$

C2. From $\varphi \Rightarrow E_G(\psi \wedge \varphi)$ infer $\varphi \Rightarrow C_G\psi$

Finally, the following axioms capture properties of announcements:

A1. $\text{ann}_j(\theta, \beta) \Rightarrow \text{announced}(j, \theta, \beta)$

A2. $\text{announced}(j, \theta, \beta) \Rightarrow K_i\text{announced}(j, \theta, \beta)$, where $i \in \beta$

Axiom **A2** extends to common knowledge immediately.

Proposition 2.3. $\text{announced}(j, \theta, \beta) \Rightarrow C_\beta\text{announced}(j, \theta, \beta)$ is provable from the previous axioms.

Proof. A straightforward derivation using **C2** and **A2**. \square

Let \mathbf{AX}^{ann} be the axiomatization $\{\mathbf{Prop}, \mathbf{MP}, \mathbf{K1-5}, \mathbf{C1-2}, \mathbf{A1-2}\}$. Recall that an axiomatization is *sound* relative to a class of models \mathcal{M} if whenever a formula \mathcal{L} is provable from the axioms, then that formula is valid in \mathcal{M} .

Theorem 2.4. \mathbf{AX}^{ann} is a sound axiomatization for $\mathcal{L}_n^{K,A}$ relative to \mathcal{C}^{ann} .

Proof. To prove soundness, it suffices to prove that every axiom in \mathbf{AX}^{ann} is valid, and that every inference rule in \mathbf{AX}^{ann} preserves validity. Axioms and inference rules **Prop**, **MP**, **K1-5**, and **C1-2** are straightforward.

Axiom **A1** is clearly valid: if $(\mathcal{A}, r, m) \models \text{ann}_j(\theta, \beta)$, then $\theta!_\beta^j \in r_e(m)$ and either $m = 0$ or $\theta!_\beta^a \notin r_e(m-1)$; because $\theta!_\beta^j \in r_e(m)$, then $(\mathcal{A}, r, m) \models \text{announced}(j, \theta, \beta)$.

Axiom **A2** is also clearly valid: if $(\mathcal{A}, r, m) \models \text{announced}(j, \theta, \beta)$, then $\theta!_\beta^j \in r_e(m)$; because $\theta!_\beta^j \in r_e(m)$ and $i \in \beta$, by property R3 of announcement systems, we have $\theta!_\beta^j \in r_i(m)$. Let $(r', m') \sim_i (r, m)$. By definition of \sim_i , $\theta!_\beta^j \in r'_i(m')$, and again by R3, $\theta!_\beta^j \in r'_e(m')$, so $(\mathcal{A}, r', m') \models \text{announced}(j, \theta, \beta)$. Therefore, we have $(\mathcal{A}, r, m) \models K_i\text{announced}(j, \theta, \beta)$. \square

Consider the following seemingly reasonable alternative to **A2**:

$$\mathbf{A2}'. \text{ ann}_j(\theta, \beta) \Rightarrow K_i \text{ann}_j(\theta, \beta)$$

Axiom **A2'** is not in fact sound for announcement systems. Roughly, this is because agents cannot know exactly when an announcement is made, since they may not be aware of time passing. If we restrict ourselves to synchronous announcement systems, however, **A2'** is now sound. Let \mathbf{AX}^{syn} be the axiomatization obtained by adding **A2'** to \mathbf{AX}^{ann} .

Theorem 2.5. \mathbf{AX}^{syn} is a sound axiomatization for $\mathcal{L}_n^{K,A}$ relative to \mathcal{C}^{syn} .

Proof. We show the validity of **A2'**. If $(\mathcal{A}, r, m) \models \text{ann}_j(\theta, \beta)$, then $\theta!_\beta^j \in r_e(m)$ and either $m = 0$ or $\theta!_\beta^j \notin r_e(m-1)$. Take any $(r', m') \sim_i (r, m)$. By definition of \sim_i , $\theta!_\beta^j \in r'_e(m')$. Because \mathcal{A} is synchronous, we have $m' = m$. Consider two cases. If $m = 0$, then $m' = 0$, so $(\mathcal{A}, r', m') \models \text{ann}_j(\theta, \beta)$. Otherwise, $\theta!_\beta^j \notin r_e(m-1)$. By property R4, $\theta!_\beta^j \notin r_i(m-1)$. By definition of \sim_i , $\theta!_\beta^j \notin r'_i(m-1)$. By property R3 and $m' = m$, $\theta!_\beta^j \notin r'_e(m'-1)$, and thus $(\mathcal{A}, r', m') \models \text{ann}_j(\theta, \beta)$. Since (r', m') was chosen arbitrarily, we have $(\mathcal{A}, r, m) \models K_i \text{ann}_j(\theta, \beta)$. \square

Note that **A2** is in fact immediately derivable from **A1** and **A2'** via **K1** and **K5**. We leave this easy verification to the reader.

2.1 Public Announcement Systems

A special class of announcement systems studied in the literature are public announcement systems. For instance, the announcement system in Example 2.2 is a public announcement system.

Definition 2.6. An announcement system \mathcal{A} is public if for every announcement $\theta!_\beta^a$ in \mathcal{A} , we have $\beta = \{1, \dots, n\}$.

Let \mathcal{C}^{pub} be the class of all public announcement systems.

The following axiom distinguishes public announcement systems:

$$\mathbf{A3}. \neg \text{announced}(j, \theta, \beta), \text{ for } \beta \text{ a proper subset of } \{1, \dots, n\}$$

As usual, axiom **A3** is really an axiom schema; sample instances of **A3** in $\mathcal{L}_3^{K,A}$ include $\neg \text{announced}(1, p, \{1, 2\})$ and $\neg \text{announced}(2, p \vee K_2 q, \{2, 3\})$. Let \mathbf{AX}^{pub} be the axiomatization $\mathbf{AX}^{ann} \cup \{\mathbf{A3}\}$.

Theorem 2.7. \mathbf{AX}^{pub} is a sound axiomatization for $\mathcal{L}_n^{K,A}$ relative to \mathcal{C}^{pub} .

Proof. Public announcement systems are announcement systems, so by Theorem 2.4, all axioms and inference rules in \mathbf{AX}^{ann} are sound. Axiom **A3** is trivially valid, because there is no announcement $\theta!_\beta^a$ with $\beta \neq \{1, \dots, n\}$ in any public announcement system. \square

As before, we can obtain a sound axiomatization for synchronous public announcement systems by adding **A2'** to \mathbf{AX}^{pub} .

3 Honest Announcements

Up until now, we have been unable to reason about the *content* of announcements. Intuitively, this inability comes from there being no restrictions on what can be announced. If we do restrict what can be announced, then we can reason to some extent about the content of announcements. A common restriction is to have agents be honest. Intuitively, an announcement is *honest* if its content is true. One interpretation of honesty, then, is to simply require that whenever an announcement $\theta!_{\beta}^a$ is made, θ holds. But this seems too weak. In particular, it permits agents to be honest “by accident”, so to speak. We tend to think of honesty as a mental attitude of an agent saying true things. Not only is the agent saying true things, but the agent knows that the things she is saying are true. This interpretation of honesty is used in the security literature, where it is often weakened to require agents to simply believe that what they say is true [3]. We will have more to say about belief in §5.

If we take this last definition of honesty, then we can define an *honest announcement system* to be an announcement system in which agents only announce what they know is true. For instance, the announcement system in Example 2.2 is honest.

We can characterize honest announcement systems semantically. Doing so is somewhat subtle because we have to take time into account. In particular, an honest announcement of $\theta!_{\beta}^a$ requires that agent a knows θ . But when is that knowledge supposed to hold?

Intuitively, knowledge that the announcement is true should hold just before the announcement is made. Because we take announcements to be instantaneous relative to the granularity of time in our models—an announcement is heard at the same time step it is made—it is easy to be inconsistent were we to require an announcement to be known to be true exactly when it is made, which is the same as when it is heard.

To see this, consider the announcement $(p \wedge \neg K_1 p)!_{\{1,2\}}^2$ of the Moore sentence [10] $p \wedge \neg K_1 p$. Suppose for simplicity that this announcement is made in a synchronous system. (This argument does not depend on synchrony, but synchrony enables a simpler example.) If this is an honest announcement, then agent 2 knows $p \wedge \neg K_1 p$ when the announcement is made, and in particular $K_2 \neg K_1 p$ holds at the time the announcement is made. When the announcement is heard by agent 1, then by synchrony agent 1 knows that agent 2 knows $p \wedge \neg K_1 p$, and in fact agent 2 knows that agent 1 knows $p \wedge \neg K_1 p$, and in particular, because $K_2 p$ implies p , we have $K_2 K_1 p$ holding when the announcement is heard by agent 1. If knowledge of the content of an announcement for honesty is determined at the same time as the announcement is heard, then we have both $K_2 \neg K_1 p$ and $K_2 K_1 p$ holding at the same time, an impossibility given our S5 interpretation of knowledge.

This argument justifies a temporal characterization of honest announcements:

- R5. For all r, θ, a, β : $(\mathcal{A}, r, 0) \models \neg \text{ann}_a(\theta, \beta)$ and for all $m \geq 1$, if $(\mathcal{A}, r, m) \models \text{ann}_a(\theta, \beta)$, then $(\mathcal{A}, r, m - 1) \models K_a \theta$.

Note that Unlike R1–4, property R5 is not a property of individual runs, but rather a property of systems as a whole, because of the presence of the knowledge operator. Thus, honesty is a global property of a system.

Definition 3.1. *An announcement system is honest if it satisfies property R5.*

Let \mathcal{C}^{hon} be the class of all honest announcement systems.

To axiomatize honest announcement systems, we need a more expressive language than the one we have been considering. In particular, we need to have formulas that relate two different time steps, to capture knowledge update. There are a few ways of doing this, and it is not clear a priori which is the more reasonable. For now, we add a past time temporal operator [9], $\ominus\varphi$, read “ φ is true in the previous time step, if one exists”, with the following semantics.

$$\begin{aligned} (\mathcal{A}, r, 0) &\models \ominus\varphi \\ (\mathcal{A}, r, m + 1) &\models \ominus\varphi \text{ iff } (\mathcal{A}, r, m) \models \varphi \end{aligned}$$

This is a so-called *weak* previous-time operator, meaning that it does not guarantee that a previous time actually exists. Let $\mathcal{L}_n^{K,A,\ominus}$ be the logic obtained by augmenting $\mathcal{L}_n^{K,A}$ with operator \ominus .

To axiomatize honest announcement systems, we need axioms to capture $\ominus\varphi$, on top of an axiom to capture property R5. Axioms for $\ominus\varphi$ are straightforward.

- P1.** $\ominus\varphi \wedge \ominus(\varphi \Rightarrow \psi) \Rightarrow \ominus\psi$
- P2.** $\ominus\varphi \Leftrightarrow (\ominus\text{false} \vee \neg\ominus(\neg\varphi))$
- P3.** $\neg\ominus\text{false} \wedge \ominus p \Rightarrow p$
- P4.** $p \Rightarrow \ominus p$
- P5.** $\neg\ominus\text{false} \wedge \ominus\text{announced}(j, \theta, \beta) \Rightarrow \text{announced}(j, \theta, \beta)$
- P6.** from φ infer $\ominus\varphi$

Note that **P2** essentially says that φ holds in the previous time step either when there is no previous time step (which can be conveniently expressed as $\ominus\text{false}$) or if $\neg\varphi$ does not hold in the previous time step. Axioms **P3–4** capture the fact that primitive propositions do not change truth value during a run. Axiom **P5** captures the fact that once an announcement has been made it remains in the local state of the receiving agents. Axiom **P6** is the necessitation rule for the previous time modality.

The honesty property R5 of announcement systems is now easy to capture:

- A4.** $\ominus\text{false} \Rightarrow \neg\text{ann}_a(\theta, \beta)$
- A5.** $\text{ann}_a(\theta, \beta) \Rightarrow \ominus K_a\theta$

Let \mathbf{AX}^{hon} be the axiomatization $\mathbf{AX}^{ann} \cup \{\mathbf{P1–5}, \mathbf{A4–5}\}$.

Theorem 3.2. \mathbf{AX}^{hon} is a sound axiomatization for $\mathcal{L}_n^{K,A,\ominus}$ relative to \mathcal{C}^{hon} .

Proof. It is trivial to check that **P1–5** are valid in all announcement systems, not just honest ones. Clearly, **A4** is valid in honest announcement systems, which do not allow announcements at time 0 by R5. We check that **A5** is valid in honest announcement systems. Let $(\mathcal{A}, r, m) \models \text{ann}_a(\theta, \beta)$. We consider two cases. If $m = 0$, then $(\mathcal{A}, r, m) \models \ominus K_a\theta$ trivially by the semantics of \ominus . If $m > 0$, then by property R5, $(\mathcal{A}, r, m - 1) \models K_a\theta$, so that $(\mathcal{A}, r, m) \models \ominus K_a\theta$. \square

What about what other agents can infer when an honest message is announced? The best we can do is to prove the following property.

Proposition 3.3. *For all agents i , $K_i \text{ann}_a(\theta, \beta) \Rightarrow K_i \ominus \theta$ is provable in \mathbf{AX}^{hon} .*

Proof. This is a straightforward derivation first using **A5** and **K2** together with **P6** to derive $\text{ann}_a(\theta, \beta) \Rightarrow \ominus \theta$, and then using **K1** and **K5** to derive $K_i \text{ann}_a(\theta, \beta) \Rightarrow K_i \ominus \theta$, along with the usual propositional reasoning involved in such derivations. \square

In other words, if an agent i knows that θ has been announced, then that agent knows that θ was true in the previous time step. If all the agents know that θ has been announced, we can derive a similar result for common knowledge:

Proposition 3.4. *For all groups G of agents, $C_G \text{ann}_a(\theta, \beta) \Rightarrow C_G \ominus \theta$ is provable in \mathbf{AX}^{hon} .*

Proof. Similar to the derivation in the proof of Proposition 3.3, except for using the S5 properties of C_β instead of K_i , and using **C2** instead of the necessitation rule **K5**. \square

Note that this result is completely independent of whether the group G is the same as the group β of recipients of the announcement.

We see now that announcing a Moore sentence such as $p \wedge \neg K_1 p$, of the kind we saw at the beginning of this section, is not a problem with our interpretation of honesty. Again, consider making such an announcement in the context of a synchronous system. By Proposition 3.3 and **P1-P3**, we have that when agent 1 hears agent 2's announcement (given that he knows when exactly the announcement has been made), he gets to know that p , but also that before the announcement p was true and he did not know it then, that is $\text{ann}_2(p \wedge \neg K_1 p, \{1, 2\}) \Rightarrow K_1 p \wedge K_1 \ominus \neg K_1 p$. For agent 2 by axiom **A5** we have that just before he made his announcement agent 2 knew that p was true and agent 1 did not know p , that is, $\text{ann}_2(p \wedge \neg K_1 p, \{1, 2\}) \Rightarrow \ominus K_2(p \wedge \neg K_1 p)$.

To look at another example, consider the honest announcement system $\mathcal{A}^{\text{muddy}}$ for the muddy children puzzle in Example 2.2. By proposition 3.3 we have that when the children hear their father's announcement and $k - 1$ rounds of children's announcements (given that they know exactly when each announcement has been made—note that $\mathcal{A}^{\text{muddy}}$ is a synchronous system), each of the dirty children $i \in \{1, \dots, n\}$ knows that he must have previously been dirty, that is, if $i \in D$ in run r^D , then $(\mathcal{A}^{\text{muddy}}, r^D, k) \models K_i \ominus \text{dirty}[i]$. But they learn this after exactly $k - 1$ rounds, while just after they heard $k - 2$ announcements, they did not know that they are dirty, i.e. $(\mathcal{A}^{\text{muddy}}, r^D, k - 1) \models \neg K_i \ominus \text{dirty}[i]$.

Clearly, a sound axiomatization for *honest public announcement systems*, in which all announcements are honest and public, can be obtained by taking $\mathbf{AX}^{\text{hon}} \cup \{\mathbf{A3}\}$.

When the system under consideration is synchronous, then we can do a bit better than Propositions 3.3 and 3.4. First, synchronous systems induce a new axiom relating time and knowledge.

KP. $\ominus K_i \varphi \Rightarrow K_i \ominus \varphi$.

Intuitively, **KP** says that agents have total recall, and do not forget that they knew facts in the past—if an agent knew φ , then that agent knows now that φ was true then. The

converse is not valid, as the following system \mathcal{A} illustrates: suppose a system with two runs, r_1 and r_2 , and a primitive proposition p . Suppose that p is true on run r_1 and false on run r_2 . Suppose further that agent 1 consider $(r_1, 0)$ and $(r_2, 0)$ equivalent (and no other points equivalent to those), but does not consider any point equivalent to $(r_1, 1)$. It is easy to check that $(\mathcal{A}, r_1, 1) \models K_1 \ominus p$, but that $(\mathcal{A}, r_1, 1) \not\models \ominus K_1 p$.

Let $\mathbf{AX}^{hs} = \mathbf{AX}^{hon} \cup \{\mathbf{A2}', \mathbf{KP}\}$.

Theorem 3.5. \mathbf{AX}^{hs} is a sound axiomatization for $\mathcal{L}_n^{K,A,\ominus}$ relative to synchronous announcement systems in \mathcal{C}^{hon} .

Proof. We show \mathbf{KP} is valid in every synchronous announcement systems, not just honest ones. Suppose $(\mathcal{A}, r, m) \models \ominus K\varphi$. We show $(\mathcal{A}, r, m) \models K\ominus\varphi$. Consider two cases. First, the case where $m = 0$. Let $(r', m') \sim_i (r, m)$. By synchrony, $m' = 0$. By definition of \ominus , $(\mathcal{A}, r', m') \models \ominus\varphi$. Because (r', m') was arbitrary, $(\mathcal{A}, r, m) \models K\ominus\varphi$. Second, the case where $m > 0$. Let $(r', m') \sim_i (r, m)$. We claim that $(\mathcal{A}, r', m') \models \ominus\varphi$. By synchrony, $m' = m > 0$. So, consider $(r', m - 1)$. By property R4 of synchronous announcement systems, because $(r', m) \sim_i (r, m)$, we have $(r', m - 1) \sim_i (r, m - 1)$. By assumption $(\mathcal{A}, r, m - 1) \models K\varphi$, meaning that $(\mathcal{A}, r', m - 1) \models \varphi$. Thus, $(\mathcal{A}, r', m) \models \ominus\varphi$. Since (r', m') was arbitrary, we have $(\mathcal{A}, r, m) \models K\ominus\varphi$, as required. \square

In honest synchronous announcement systems, when an announcement is made, then every recipient of the announcement knows that θ was true before the announcement was made, and in fact, it is common knowledge amongst every subset of the recipients.

Proposition 3.6.

- (a) $\text{ann}_a(\theta, \beta) \Rightarrow K_i \ominus \theta$ is provable in \mathbf{AX}^{hs} , for $i \in \beta$;
- (b) $\text{ann}_a(\theta, \beta) \Rightarrow C_{\beta'} \ominus \theta$ is provable in \mathbf{AX}^{hs} , for $\beta' \subseteq \beta$.

Proof. Part (a) follows directly from $\mathbf{A2}'$ and Proposition 3.3. Part (b) follows from part (a) and $\mathbf{C2}$. \square

A sound axiomatization for *synchronous honest public announcement systems* is obtained by taking $\mathbf{AX}^{hps} = \mathbf{AX}^{hon} \cup \{\mathbf{A2}', \mathbf{A3}, \mathbf{KP}\}$.

A variant of honest announcement systems are *dishonest announcement systems* in which agents can only lie, that is announce the negation of something they know is true. These are defined by replacing $K_a\theta$ by $\neg K_a\theta$ in the definition of honest announcement systems. A sound axiomatization can be obtained by replacing $K_a\theta$ by $\neg K_a\theta$ in $\mathbf{A5}$.

4 Relationship with Public Announcement Logic

We compare our logic $\mathcal{L}_n^{K,A}$ with *Public Announcement Logic* (PAL) [18] The syntax of PAL is as follows

$$F, G ::= p \mid F \wedge G \mid \neg F \mid K_i F \mid [N!^j] F$$

where $K_i F$ stands for “agent i knows that F holds” and $[N!^j]F$ stands for “after agent j announces N , proposition F holds”,² where N is given by

$$N, M ::= p \mid N \wedge M \mid \neg N \mid K_i N$$

In order to be compatible with the syntax of our logic, we work here with the original syntax of PAL where announcement formulas cannot themselves be announced. As we pointed out in Section 2, it is not immediately clear how to concretely interpret nested announcements, and indeed even the abstract semantics of PAL is not entirely intuitive on this subject. We therefore focus on the simpler logic here, and save the investigation of nested announcements for future work.

We show that our models validate the axioms of PAL. To make this statement precise, we translate PAL formulas to formulas in our logic in a way that translates PAL-provable formulas to provable formulas with a similar interpretation in our logic. We denote the translation of a PAL formula F by $\llbracket F \rrbracket$ and define it inductively as follows

$$\begin{aligned} \llbracket p \rrbracket &\triangleq p \\ \llbracket F \wedge G \rrbracket &\triangleq \llbracket F \rrbracket \wedge \llbracket G \rrbracket \\ \llbracket \neg F \rrbracket &\triangleq \neg \llbracket F \rrbracket \\ \llbracket K_i F \rrbracket &\triangleq K_i \llbracket F \rrbracket \\ \llbracket [N!^j]F \rrbracket &\triangleq \llbracket [N!^j](\llbracket F \rrbracket) \rrbracket \end{aligned}$$

where

$$\llbracket [N!^j](\llbracket F \rrbracket) \rrbracket \triangleq \mathbf{ann}_j(\llbracket N \rrbracket, \{1, \dots, n\}) \Rightarrow \llbracket F \rrbracket$$

Other than the usual propositional logic and epistemic axioms of S5, PAL uses the following specialized axioms to deal with announcements

- D1. $[N!^j]p \Leftrightarrow (\mathit{prec}(N!^j) \Rightarrow p)$
- D2. $[N!^j]\neg F \Leftrightarrow (\mathit{prec}(N!^j) \Rightarrow \neg[N!^j]F)$
- D3. $[N!^j]K_i F \Leftrightarrow (\mathit{prec}(N!^j) \Rightarrow K_i[N!^j]F)$

D1 is referred to as *preservation of facts* and says that epistemic actions do not affect the truth value of primitive propositions. D2 is referred to as *partial functionality*, and D3 is the *action-knowledge* axiom and says that agents know the consequences of announcements: they know a proposition after an announcement exactly when they know that making the announcement yields that proposition.

The $\mathit{prec}(N!^j)$ stands for the *precondition* of an announcement, that is, the formulas that should be true in order for the announcement to be made. Since the runs of our systems correspond to the actual evolution of a scenario independently of the logic, our runs encode

²Strictly speaking, the syntax of PAL does not explicitly mention announcers. Announcements are simply written $N!$. Announcers can usually be inferred from the context; here we make them explicit to more easily compare PAL to our logic.

when each announcement is being made. So we can simply translate the precondition of an announcement to the proposition that says the announcement has just been made, as follows:

$$\llbracket \text{prec}(N!^j) \rrbracket \triangleq \text{ann}_j(\llbracket N \rrbracket, \{1, \dots, n\})$$

For instance, in honest public announcement systems, this will imply $\ominus K_j \llbracket N \rrbracket$ and thus $\ominus \llbracket N \rrbracket$; in dishonest public announcements it implies $\ominus \neg \llbracket N \rrbracket$. It is this flexibility, this possibility we have of interpreting preconditions in different ways, that lets our logic be more flexible than PAL, for instance by not being restricted to scenarios where every agent is honest.³

Consider the axiomatization \mathbf{AX}^{hps} for synchronous honest public announcement systems given in §3. The following theorem shows that our translation of PAL formulas preserves provability.

Theorem 4.1. *If F is a formula provable in PAL, then $\llbracket F \rrbracket$ is provable in \mathbf{AX}^{hps} .*

Proof. It suffices to show that all the axioms of PAL are provable in \mathbf{AX}^{hps} . The propositional axioms of PAL translate to the propositional axioms **Prop** and **MP**, and the epistemic axioms translate to axioms **K1–5**, so it is enough to show that PAL’s announcement axioms are provable in \mathbf{AX}^{hps} .

For simplicity, in this proof, we shorthand the translation of a formula as the formula itself, that is we write N instead of $\llbracket N \rrbracket$ and F instead of $\llbracket F \rrbracket$. Since all the announcements in the proof are to the same receivers, we also shorthand $\text{ann}_j(N, \{1, \dots, n\})$ to $\text{ann}_j(N)$.

- D1 translates to the following tautology.

$$\left(\text{ann}_j(N) \Rightarrow p \right) \Leftrightarrow \left(\text{ann}_j(N) \Rightarrow p \right)$$

- D2 translates to the following, which is easily derivable using **Prop**.

$$\left(\text{ann}_j(N) \Rightarrow \neg F \right) \Leftrightarrow \left(\text{ann}_j(N) \Rightarrow \neg(\text{ann}_j(N) \Rightarrow F) \right)$$

- D3 translates to

$$\left(\text{ann}_j(N) \Rightarrow K_i F \right) \Leftrightarrow \left(\text{ann}_j(N) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F) \right)$$

We first show $(\text{ann}_j(N) \Rightarrow K_i F) \Rightarrow (\text{ann}_j(N) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F))$. Using **Prop**, this is equivalent to $(\text{ann}_j(N) \wedge \text{ann}_j(N) \Rightarrow K_i F) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F)$. To prove this implication, we first assume $\text{ann}_j(N)$ and $\text{ann}_j(N) \Rightarrow K_i F$. By **MP**, we get $K_i F$. By synchrony (**A2'**) and **MP** we also get $K_i \text{ann}_j(N)$. Therefore, we can conclude $K_i(\text{ann}_j(N)) \wedge K_i F$, which is equivalent to $K_i(\text{ann}_j(N) \wedge F)$ by **K1–5**, which is equivalent to $K_i(\text{ann}_j(N) \wedge (\text{ann}_j(N) \Rightarrow F))$ by **K1–5** and **Prop**, which is equivalent

³In most versions of PAL, the precondition is the formula that is being announced $\text{prec}(N!^j) = N$, which implies that in a scenario every true formula could be announced. This is of course not the case for most scenarios. In more general versions of PAL, such as Dynamic Epistemic Logic (DEL) [2], the precondition is kept as a variable whose instantiation depends on the scenario under examination.

to $K_i \text{ann}_j(N) \wedge K_i(\text{ann}_j(N) \Rightarrow F)$ by **K1–5**, which in turns implies $K_i(\text{ann}_j(N) \Rightarrow F)$. Thus, $(\text{ann}_j(N) \wedge \text{ann}_j(N) \Rightarrow K_i F) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F)$, as required.

The reverse implication is similar, and simpler. To show $(\text{ann}_j(N) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F)) \Rightarrow (\text{ann}_j(N) \Rightarrow K_i F)$, it is sufficient to prove the equivalent (by **Prop**) $(\text{ann}_j(N) \wedge \text{ann}_j(N) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F)) \Rightarrow K_i F$. Therefore, assume $\text{ann}_j(N)$ and $\text{ann}_j(N) \Rightarrow (K_i(\text{ann}_j(N) \Rightarrow F))$. By **MP**, we get $K_i(\text{ann}_j(N) \Rightarrow F)$. By **K1** and **Prop**, this last is equivalent to $K_i \text{ann}_j(N) \Rightarrow K_i F$. By synchrony (**A2'**) and by the assumption that $\text{ann}_j(N)$ holds, we know that $K_i \text{ann}_j(N)$ holds, and therefore by **MP** that $K_i F$ holds, which is what we wanted. Thus $(\text{ann}_j(N) \wedge \text{ann}_j(N) \Rightarrow K_i(\text{ann}_j(N) \Rightarrow F)) \Rightarrow K_i F$, as required.

Informally speaking, for the \Rightarrow direction, assume that if an announcement has just happened, then agent i gets to know that F is true. By synchrony, this also implies that he knows that the announcement has just happened, so he knows the conjunction of both, which implies the right hand side. For the \Leftarrow direction, assume that if an announcement has just happened, then if agent i knows that the announcement has just happened then he knows that F is true. But by synchrony, i already knows that an announcement has just happened, so he knows that F is true, which is what the left-hand side is expressing. \square

Thus every derivable formula of PAL is derivable (after translation) in \mathbf{AX}^{hps} . Although we do not yet have a similar result for *truth* (that our translation preserves the truth of PAL formulas), a weaker corollary about preservation of *validity* (truth in all models) immediately follows. Proving that our translation preserves truth of PAL formulas within a specific model requires developing a faithful translation of the PAL models to the models of our logic. We save this for future work, and return to this topic in the conclusion.

Corollary 4.2. *If F is a valid formula of PAL, then $\llbracket F \rrbracket$ is valid in synchronous honest public announcement systems.*

Proof. Assume that F is a valid formula of PAL, since PAL is complete, F is derivable from the axioms of PAL. As a result, by theorem 4.1 its translation $\llbracket F \rrbracket$ becomes derivable in \mathbf{AX}^{hps} and since \mathbf{AX}^{hps} is sound, we obtain that $\llbracket F \rrbracket$ is valid in synchronous honest public announcement systems. \square

Example 4.3. *In the muddy children puzzle in Example 2.2,*

$$\left[\left(\bigvee_{i=1}^n \text{dirty}[i] \right)!^0 \right] K_i \bigvee_{i=1}^n \text{dirty}[i]$$

is derivable in PAL. Its translation to our logic

$$\text{ann}_0 \left(\bigvee_{i=1}^n \text{dirty}[i], \{0, 1, \dots, n\} \right) \Rightarrow K_i \bigvee_{i=1}^n \text{dirty}[i]$$

*is derivable: Proposition 3.6 derives $\text{ann}_0(\bigvee_{i=1}^n \text{dirty}[i], \{0, 1, \dots, n\}) \Rightarrow K_i \ominus \bigvee_{i=1}^n \text{dirty}[i]$. This implies $\text{ann}_0(\bigvee_{i=1}^n \text{dirty}[i], \{0, 1, \dots, n\}) \Rightarrow K_i \bigvee_{i=1}^n \ominus \text{dirty}[i]$, since \ominus preserves disjunction in one direction. The latter implies $\text{ann}_0(\bigvee_{i=1}^n \text{dirty}[i], \{0, 1, \dots, n\}) \Rightarrow K_i \bigvee_{i=1}^n \text{dirty}[i]$ by **P3** and the fact that there is a previous time step before the father's announcement.*

So PAL is interpretable in our logic. What about the other way around? Is our logic interpretable in PAL, or is our logic more expressive? One way to answer this question is to provide a converse to Theorem 4.1. Part of the problem here is the previous-time modality of our logic.⁴ As we saw, the previous-time modality seems necessary to handle Moore sentences. This suggests that our logic is more expressive, in some sense, than PAL. Recently, PAL has been extended with temporal modalities, including a previous-time operator [13, 19]. The study of how these logics relate to ours is left to future work.

Theorem 4.1 and corollary 4.2 extend to *Public Announcement Logic with Common Knowledge*. The syntax of this logic is an extension of PAL's with a common knowledge operator $C_\beta F$, which we translate as

$$\llbracket C_\beta F \rrbracket \triangleq C_\beta \llbracket F \rrbracket.$$

Its axiomatization is the same as PAL's except that axiom D3 changes to

$$\text{D3}'. \quad \llbracket N! \rrbracket C_\beta F \Leftrightarrow (\text{prec}(N!) \Rightarrow C_\beta \llbracket N! \rrbracket F).$$

Theorem 4.4. *If F is a formula provable in PAL with common knowledge, then $\llbracket F \rrbracket$ is provable in \mathbf{AX}^{hps} .*

Proof. The validity of D3' follows from $C_\beta \text{ann}_j(N, \{1, \dots, n\}) \Leftrightarrow \text{ann}_j(N, \{1, \dots, n\})$, which is derivable from axioms of common knowledge and synchrony. \square

Corollary 4.5. *If F is a valid formula of PAL with common knowledge, then $\llbracket F \rrbracket$ is valid in synchronous honest public announcement systems.*

Proof. Assume that F is a valid formula of PAL with common knowledge, since PAL with common knowledge is complete, F is derivable from the axioms of PAL with common knowledge. As a result, by theorem 4.1 its translation $\llbracket F \rrbracket$ becomes derivable in \mathbf{AX}^{hps} and since \mathbf{AX}^{hps} is sound, we obtain that $\llbracket F \rrbracket$ is valid in synchronous honest public announcement systems. \square

5 Private Announcements and Belief

Private announcements, that is, announcements not broadcasted to every agent in the system, have held no special status in this work until now. Several researchers have argued, however, that we can say something more in the presence of private announcements. Roughly speaking, they propose that when an agent does not hear an announcement, she should assume that no announcement has been made, and reason accordingly.

Consider the following sample scenario with three agents, Alice, Bob, and Zoe. Alice tosses a coin, the outcome of which is not seen by any agent other than Alice. Alice then announces to herself and to Bob the result of the toss. Suppose we have two propositions,

⁴A natural translation that comes to mind is the map that simply forgets the previous-time modality, that is, the map $\llbracket \cdot \rrbracket^{-1}$ from $\mathcal{L}_n^{K,A}$ to PAL, such that $\llbracket \odot \varphi \rrbracket^{-1} \triangleq \llbracket \varphi \rrbracket^{-1}$, sends the implication by an announcement (for both ann and announced formulas) to the dynamic modality, i.e. $\llbracket \text{ann}_j(\varphi, \beta) \Rightarrow \psi \rrbracket^{-1} \triangleq \llbracket \varphi \rrbracket^{-1!j} \llbracket \psi \rrbracket^{-1}$, and acts as identity on all other formulas. It is not hard to show that this translation does not preserve provability; the announcement of a Moore sentence such as the one in §3 provides a counterexample.

H and T , true when the result of the coin toss is heads and tails, respectively. It is simple to model this scenario in our framework using a system consisting of two runs r^H and r^T , where the initial state of Alice and the environment in each run records whether the coin lands heads (run r^H) or tails (run r^T). At time 1, Alice announces the result of the coin toss, her announcement is recorded in the local states of Alice and Bob. Thus, the local states of the environment are $r_e^T(1) = \langle T, T!_{\{A,B\}}^A \rangle$, and the local states of agents are $r_A^T(1) = \langle T, T!_{\{A,B\}}^A \rangle$, $r_B^T(1) = \langle -, T!_{\{A,B\}}^A \rangle$, but $r_Z^T(1) = \langle - \rangle$, and similarly for $r_Z^H(1)$. It is easy to check that $(r^H, 0) \models \neg K_B(H) \wedge \neg K_B(T)$, while $(r^H, 1) \models K_A(H) \wedge K_B(H) \wedge \neg K_Z(H) \wedge \neg K_Z(T)$, and $(r^H, 1) \models \neg K_Z K_B H$.

If we do take for granted an interpretation of private announcements with the view that an agent having not heard an announcement assumes that that announcement was never made, then Zoe thinks that $\neg K_B(H)$ at the second state of r^H . We say “thinks” here instead of “knows” because knowledge is generally taken to satisfy the truth axiom $K_i \varphi \Rightarrow \varphi$, and so clearly Zoe cannot know $\neg K_B(H)$.

In some sense, this interpretation of private announcements, and most importantly, of what information agents can derive from private announcements, amounts to a form of belief. Intuitively, Zoe *believes* that Bob does not know H because, having not seen the private announcement, she assumes that there was no announcement. DEL [2], for instance, defines \Box_i to be a belief operator with that kind of interpretation. In DEL, for the above scenario, it is possible to derive that $\Box_Z \neg \Box_B(H)$. DEL’s definition of belief is somewhat abstract, however, and given a purely proof-theoretic interpretation.

Our concrete semantics, based on actual system evolution, let us explore how to rationally capture such an interpretation for belief. Intuitively, to capture such an interpretation, we need to define belief in such a way that the outsider agent i who did not hear any announcement considers possible other states where other agents have in fact not received the purported private announcement. But of course, those states where agents have not received the announcement are not “real” states, that is, they are not states in the scenario being modeled, and therefore need to be added to the model. The question becomes, then, what states, and therefore what runs, should be added to the system to capture this kind of belief. It turns out that it is in general not wholly possible to extract such an interpretation for a description of the scenario being modeled, but rather we have to make assumptions on the initial knowledge of the various agents in the scenario about the *strategy* followed by all the agents in the scenario.

In order to make all of this precise, we need to carefully examine how systems are, in fact, generated. We follow an approach inspired by [4, Chapter 5]. Intuitively, a system is generated by each agent following a strategy (a protocol, a program), which is simply a description of what actions the agent may take as a function of her local state. Actions, for our purposes, will consist of making announcements (although we could incorporate other actions, such as tossing coins to randomize strategies). A joint strategy P consists of a strategy for every agent in the system, and describes how the system evolves.

We can formalize the general idea above as follows. For illustration purposes, here, we define only how to generate *synchronous* announcement systems. Fix a set L_i of local states for agent i (the local states that arise in some system) and a set A_i of possible actions that agent i can perform. An action is any (finite) number of announcements. This can be

formalized by taking A_i to be the class of all finite subsets of $\{\mathbf{ann}(\theta, \beta) \mid \theta \in \mathcal{L}_n^K, \beta \subseteq \{1, \dots, n\}\}$; we view \emptyset as the action of making no announcement.

A *strategy* P_i for agent i is a function that associates with every local state in L_i a nonempty subset of actions in A_i . Intuitively, $P_i(s)$ is the set of actions that agent i may perform in local state s . Notice that agent i 's actions can depend only on her local state. Also, and in contrast to some definitions of strategy in other fields, strategies for us are nondeterministic—they do not prescribe which actual action to take in a specific state, but give the set of possible actions that can be performed.

To consider the effect of all the agents' strategies on the global state of the system, we define a *joint strategy* (P_1, \dots, P_n) , consisting of a strategy for each of the agents, that associates with each global state a subset of possible *joint actions*, a subset of $A \subseteq A_1 \times \dots \times A_n$.

Joint actions transform global states. To capture their effect, associate with every joint action $(\mathbf{a}_1, \dots, \mathbf{a}_n)$ a function from global states to global states. Formally, we let $\tau(\mathbf{a}_1, \dots, \mathbf{a}_n)$ be a function taking a global state (s_e, s_1, \dots, s_n) to $(s'_e, s'_1, \dots, s'_n)$, where

$$\begin{aligned} s'_i &= s_i \cdot \langle \{\theta!_\beta^j \mid \mathbf{ann}(\theta, \beta) \in \mathbf{a}_j \text{ and } i \in \beta\} \rangle & \text{for } i \in \{1, \dots, n\} \\ s'_e &= s_e \cdot \langle \{\theta!_\beta^j \mid \mathbf{ann}(\theta, \beta) \in \mathbf{a}_j\} \rangle \end{aligned}$$

and \cdot denotes sequence concatenation.

Given a joint strategy (with the above associated interpretation τ for joint actions) and a set of initial global states, we can generate a system in a straightforward way. Intuitively, the system consists of all the runs that are obtained by running the joint strategy from one of the initial global states. More formally, say that run r is *consistent with joint strategy* P if it could have been generated by P , that is, for all m , $r(m+1)$ is the result of applying a joint action \mathbf{a} that could have been performing according to joint strategy P to $r(m)$. (More precisely, there exists a joint action $(\mathbf{a}_1, \dots, \mathbf{a}_n)$ such that $\mathbf{a}_i \in P_i(r_i(m))$ and $r(m+1) = \tau(\mathbf{a}_1, \dots, \mathbf{a}_n)(r(m))$.) Given a set I of global states, a joint strategy P , and an interpretation π , the announcement system $\mathcal{A}^{P,I} = (\mathcal{R}^{P,I}, \pi)$ consists of all the runs $\mathcal{R}^{P,I}$ consistent with joint strategy in \mathcal{P} that starts in some initial global state in I .

In our example at the beginning of the section, initial states for Alice include the result of the coin toss, either T or H , the initial state for Bob is empty, the initial state for Zoe is empty. Let I_{ABZ} be the corresponding set of initial states. The strategy P_A for Alice is to announce H or T to $\{A, B\}$ in the first state, and do nothing thereafter. The strategy P_B for Bob is to do nothing. The strategy P_Z for Zoe is to do nothing. The strategies are simple, and it is easy to check that (with the appropriate interpretation for the primitive propositions) $\mathcal{A}^{(P_A, P_B, P_Z), I_{ABZ}}$ is the system we described at the beginning of the section.

To understand the notion of belief where an agent behaves as though no announcement has been made at all when she does not hear an announcement, we need to understand what states the agent believes exist, and we propose to use the notion of strategy, and most importantly, those strategies that an agent believes *other* agents are using, as opposed to the real ones they are using.

Presumably, in the actual scenario, each agent is using a strategy.⁵ What we need to add to the scenario is the strategy that each agent believes the other agents are using. Thus,

⁵For simplicity, here, we consider scenarios where agents use a single strategy. It is not difficult to

for instance, Zoe, in the above example, may believe that Alice tosses a coin but does not send an announcement to anyone. (This is of course not Alice’s actual strategy, who does in fact make an announcement to Bob.)

We take an announcement system *for belief* to be an announcement system generated by a joint strategy $P = (P_1, \dots, P_n)$ and a set of initial states I , along with a map \mathcal{P}_i for every agent i indicating the strategy $\mathcal{P}_i(j)$ that agents i believes agent j is using. Because agents are generally not mistaken about their own strategy, we take $\mathcal{P}_i(i)$ to be the actual strategy P_i that agent i is using. If \mathcal{A} is the system for belief as defined above, define $\mathcal{A}_i = \mathcal{A}^{(\mathcal{P}_i(1), \dots, \mathcal{P}_i(n)), I}$ for each agent i , that is, the system generated by the strategies that agent i believes the other agents are using. In other words, \mathcal{A}_i represents agent i ’s view of the scenario.

Returning to our coin-tossing example, we can suppose that Zoe does not believe Alice does any announcement, meaning that $\mathcal{P}_Z(A)$ is the strategy where Alice does nothing. We can take $\mathcal{P}_Z(B)$ to be the strategy where Bob does nothing as well. ($\mathcal{P}_Z(Z)$ is Zoe’s actual strategy, again to do nothing.) Thus, \mathcal{A}_Z here is the announcement system where there are no announcement made on any run.

We can define a new modality B_i that captures what agents believe in an announcement system for belief. At a point (r, m) of \mathcal{A} , take

$$(\mathcal{A}, r, m) \models B_i\varphi \text{ iff for all } (r', m') \text{ in } \mathcal{A}_i \text{ with } (r', m') \sim_i (r, m), (\mathcal{A}_i, r', m') \models \varphi$$

We extend \sim_i to all points of \mathcal{A} and \mathcal{A}_i by again taking two points to be \sim_i -equivalent if agent i has the same local states at both points. An agent believes φ at some point (r, m) of \mathcal{A} if φ is true at every point *of the system that the agent believes is the system describing the actual scenario* in which she has the same local state.

We can now check that if \mathcal{A} is the announcement system for belief generated by joint strategy (P_A, P_B, P_{AZ}) and initial states I_{ABZ} corresponding to the coin-tossing example, and letting \mathcal{P}_Z be as above, we get $(\mathcal{A}, r^H, 1) \models B_Z \neg K_B(H)$, that is, Zoe believes that Bob does not know that H was the result of the coin toss, clearly a false belief, but consistent with the interpretation of belief we set out to capture.

That $B_i\varphi$ represents belief is justified by the fact that it satisfies the KD45 axioms for belief—it satisfies all the axioms for knowledge except for $B_i\varphi \Rightarrow \varphi$, at least on the runs corresponding to the real system \mathcal{A} . It is then easy to show that the translation of axiom D3 where K_i is replaced with B_i is still derivable in our logic. Note that if an agent i believes that $\mathcal{P}_i(j) = P_j$ for all j , that is, that the strategies that the other agents are using are the actual strategies they are using, then $B_i\varphi \Rightarrow K_i\varphi$ holds for all runs of \mathcal{A} , so that $B_i\varphi \Rightarrow \varphi$ holds for all runs of \mathcal{A} .

6 Conclusion

Logics of announcements have proved popular as formal languages for reasoning about agents that explicitly exchange messages involving statements about their and other agents’ knowledge. The main question that these logics must address is how to update the state of

generalize to agents using one of a set of strategies, but the result does not offer more insight, although it may prove more applicable in practice.

knowledge of an agent based on hearing an announcement of some other agent’s knowledge. Assessing whether this knowledge update is suitable for a specific application of a logic of announcements requires a clear semantics that makes the interpretation of the update apparent. Unfortunately, the semantics of most logics of announcements has remained fairly abstract. There has been comparatively little work on developing natural semantics for logics of announcement grounded in the actual execution of a system, or evolution of a scenario.

We have initiated such a development in this paper, describing a concrete semantics for a simple logic of announcements, where the semantics is given in terms of sets of runs of a system, each run recording exactly what is happening in the system during each execution. This means, in particular, that our models are independent of the choice of the logic used to reason about them. In fact, announcements have no special status in our models. An announcement is just a message like any other. Announcements impact the knowledge of other agents like any other message in systems of communicating agents. And to model situations where announcements have a specific meaning that can affect other agents in specific ways, it suffices to impose constraints on the possible executions of a system. We examine honesty as a particularly simple instance of constraints on executions allowing for a connection to be made between a message being announced and information about the content of the message being disseminated.

Our work here is a first step in the development of natural grounded semantics for logics of announcements. There are several questions to be resolved, several choices to explore. A question that we have tried to address, but to which we do not yet have a satisfactory answer, is that of the exact relationship between PAL (for instance), and our logic. As it stands, the relationship we exhibit in this paper is a preservation of provability (and therefore of validity): if a formula F is provable in PAL, then its translation is provable in our logic. While this is the right starting point, it does not completely capture the intuition that our translation preserves the meaning of PAL formulas. A stronger theorem would establish that a PAL formula that is true in a model M remains true after translation in the systems \mathcal{A}_M corresponding to model M . This aspect of the relationship between PAL and our logic remains to be studied, possibly along the lines of recent results of [15, 13] on relating models of ETL [12] to models of DEL [2].

To more firmly ground our concrete semantics, we need to extend our logic to be able to interpret more expressive logics of announcements [18], dynamic versions of PAL such as DEL. The issues that arise here amounts to understanding the relationship between the branching time semantics of DEL with the more linear time semantics of the runs-and-systems framework. One possibility is to move to a branching-time form of runs and systems, such as found in [17]. Aside from looking at more dynamic forms of PAL, it would also be interesting to look at temporal extensions of PAL, such as recent work by Sack [13] and Yap [19]. Finally, we believe some of the sound axiomatizations we presented in this paper are in fact complete for their respective classes of announcement systems, or close to be. It would be interesting to establish such completeness results.

Acknowledgement. The original ideas behind this paper were formed while attending the “Mathematical Methods for Security Protocols” workshop organized by Prakash Panangaden at the Bellairs Research Institute of McGill University in March 2007. The authors

would like to express their gratitude to Prakash and Bellairs. M.S. would also like to thank Lena Kurzen for fruitful discussions on Moore sentences at LOFT'08.

References

- [1] A. Baltag, B. Coecke, and M. Sadrzadeh. Epistemic actions as resources. *Journal of Logic and Computation*, 17(3), 2007.
- [2] A. Baltag and L. S. Moss. Logics for epistemic programs. *Synthese*, 139:165–224, 2004.
- [3] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.
- [4] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
- [5] J. Gerbrandy. Dynamic epistemic logic. In L. S. Moss et al, editor, *Logic, Language, and Information*, volume 2. CSLI Publication, 1999.
- [6] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [7] J. Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.
- [8] S. Kripke. A semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
- [9] O. Lichtenstein, A. Pnueli, and L. Zuck. The glory of the past. In *Proc. Conference on Logics of Programs*, number 193 in Lecture Notes in Computer Science, pages 196–218. Springer-Verlag, 1985.
- [10] G. E. Moore. *A Reply to my Critics*, volume 4 of *The Library of Living Philosophers*, pages 535–677. Northwestern University, 1942.
- [11] P. Panangaden and K. Taylor. Concurrent common knowledge. *Distributed Computing*, 6:73–93, 1992.
- [12] R. Parikh and R. Ramanujan. A knowledge based semantics of messages. *Journal of Logic, Language and Information*, 12(4):453–467, 2003.
- [13] J. Sack. Temporal languages for epistemic programs. *Journal of Logic, Language and Information*, 17(2):183–216, 2008.
- [14] J. van Benthem. One is a lonely number. Technical Report PP-2002-27, ILLC, Amsterdam, 2002.
- [15] J. van Benthem, J. Gerbrandy, and E. Pacuit. Merging frameworks for interaction: DEL and ETL. In *Proc. 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK'07)*. ACM Press, 2007.

- [16] W. van der Hoek and M. Wooldridge. Time, knowledge, and cooperation: Alternating-time temporal epistemic logic. In *Proc. COORDINATION'02*, 2002.
- [17] R. van der Meyden and K. Wong. Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75(1):93–123, 2003.
- [18] H. van Ditmarsch, W. van der Hoek, and B. Kooi. *Dynamic Epistemic Logic*. Synthese Library. Springer-Verlag, 2007.
- [19] A. Yap. Product update and looking backwards. ILLC Prepublication Series 39, ILLC, Amsterdam, 2006.