

Communication Pattern Models: An Extension of Action Models for Dynamic-Network Distributed Systems

Diego A. Velázquez* Armando Castañeda† David A. Rosenblueth
velazquez-diego@ciencias.unam.mx armando.castaneda@im.unam.mx drosenbl@unam.mx

Universidad Nacional Autónoma de México

Halpern and Moses were the first to recognize, in 1984, the importance of a formal treatment of knowledge in distributed computing. Many works in distributed computing, however, still employ informal notions of knowledge. Hence, it is critical to further study such formalizations. Action models, a significant approach to modeling dynamic epistemic logic, have only recently been applied to distributed computing, for instance, by Goubault, Ledent, and Rajsbaum. Using action models for analyzing distributed-computing environments, as proposed by these authors, has drawbacks, however. In particular, a direct use of action models may cause such models to grow exponentially as the computation of the distributed system evolves. Hence, our motivation is finding compact action models for distributed systems. We introduce *communication pattern models* as an extension of both ordinary action models and their update operator. We give a systematic construction of communication pattern models for a large variety of distributed-computing models called *dynamic-network models*. For a proper subclass of dynamic-network models called *oblivious*, the communication pattern model remains the same throughout the computation.

1 Introduction

A formal treatment of the concept of *knowledge* is important yet little studied in the distributed-computing literature. Authors in distributed computing often refer to the knowledge of the different agents or processes, but typically do so only *informally*. Hence, a formal basis of knowledge in distributed computing would increase the power to prove formal results. The first step in this direction was taken by Halpern and Moses [11]. A topological approach [12] to distributed computing resulted in a further connection [10] with epistemic logic. Such a connection uses epistemic “action models” [1, 8] to capture communication between agents. We observe, however, that the action models proposed in [10] for the *Iterated Immediate Snapshot* (IIS) computing model [4, 5] not only vary at each communication round, but each action model itself is structurally isomorphic to the resulting epistemic model, which paradoxically, requires knowing the desired result beforehand. Our objective is to develop a different connection between *dynamic-network models* [7, 14, 16] (which include the IIS model), and *Dynamic Epistemic Logic* (DEL) [1, 8], more appropriate for computing knowledge change in these systems.

Context. In a distributed system, communication is typically performed either by sending and receiving messages, or by writing to, and reading from, a shared memory. The communication patterns (i.e., who communicated with whom) that can occur may change from model to model. When designing and analyzing distributed systems, it is often the case that authors informally refer to what an agent “knows” after an agent performs some action. There is indeed a formal connection between distributed systems

*Diego A. Velázquez is a doctoral student at the Programa de Doctorado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, and is the recipient of a fellowship from CONACyT.

†Armando Castañeda was supported by PAPIIT project IN108720.

and epistemic logic: this connection was initiated by Halpern and Moses in 1984 [11], showing that distributed systems can be rigorously studied from an epistemic-logic viewpoint. Roughly, a distributed protocol is studied through an *epistemic model* with each of its states representing a possible *configuration* of the protocol. Since its discovery, the epistemic-based approach to distributed systems has been fruitful, as shown in the book by Fagin, Halpern, Moses, and Vardi [9].

An important connection between distributed computing and topology was discovered in three independent papers by Borowsky and Gafni [3], Herlihy and Shavit [13], and Saks and Zaharoglou [18] in 1993, and since then this approach has provided useful techniques to show a number of important results in this field. The book by Herlihy, Kozlov, and Rajsbaum [12] provides a comprehensive description of this connection.

Recently, Goubault, Ledent, and Rajsbaum have shown [10] that the epistemic-based approach can be directly connected to the topology-based approach to distributed systems. The topological approach studies a distributed protocol through its topological representation: a geometric object, called *simpli-cial complex*, where each of its faces is associated with a *configuration* of the protocol. In essence, Goubault, Ledent, and Rajsbaum established [10] a correspondence between the topological description of distributed protocols and epistemic models.

A second interesting result of these authors is that the communication patterns allowed in the IIS distributed model can also be described using epistemic-logic tools from DEL: the communication in a distributed model can be modeled with an action model capturing the communication events that can occur, and the *restricted modal product* operator shows how knowledge evolves after agents exchange information in a *communication round*. (A more thorough summary of [10] appears in Section 6.)

We observe that the action models of [10] describing communication in the IIS model have drawbacks: First, such action models are different for each communication round (an ideal representation of communication would not depend on the communication rounds that have been executed so far). In addition, the size of such action models grows exponentially as the computation develops. Moreover, such action models are structurally isomorphic to the epistemic models we wish to compute. The action models of [10], therefore, not only are not useful for computing the epistemic model resulting from a communication event, but are not a succinct representation of the communication that can happen in the IIS model. This phenomenon is opposite to the description of communication in the topological approach, where we have a geometric and compact description of the communication in the IIS model: the communication is clearly described as a subdivision [12, Chapter 11].¹

Contributions. We are interested in the following question: in the spirit of the action-model approach to DEL, is it possible to describe the communication in a distributed model in a compact manner? As a first step, we try to salvage the approach of [10], by attempting to find an action model applicable to every communication round for *two* agents with binary inputs in the IIS model. We exhibit a family of action models with a constant number of events, although each event is labeled with a precondition formula whose size does increase at each communication round. For obtaining these action models, it was crucial to know *in advance* the epistemic model after a communication round. We have not been able to find a similar family for *three or more* agents yet. The case of *m*-ary inputs for $m \geq 3$ would be even harder to analyze.

The drawbacks of the action models proposed in [10], together with our unsuccessful efforts to find action models for IIS of small size, are motivations for investigating a different approach. We hence consider an extension of action models that allows us to easily derive models of small size. Moreover, we study not only the IIS model but also a larger class of message-passing models called *dynamic-network*

¹Informally, a subdivision results from dividing the faces of a geometrical object into more faces preserving its shape.

models [7, 14, 16]. Roughly speaking, in a dynamic-network model, the agents execute infinite sequences of communication rounds. In each round, the agents communicate according to a *communication pattern* that specifies who communicates with whom in that round. A proper subclass of dynamic-network models are those known as *oblivious* that are specified with a set of communication patterns that can occur in any round, regardless of the communication patterns that have occurred so far in the execution. The IIS model can be defined as an oblivious dynamic-network model.

Our main contribution is a simple but powerful extension to the existing action models and its restricted modal product. For every dynamic-network model, we systematically define an infinite sequence of *communication pattern models* that represent how knowledge changes when agents communicate in the *full-information protocol*, hence making our approach amenable to be extended to automated formal verification of distributed systems. For the case of oblivious models, the communication pattern model remains the same all through the execution. Hence, we are able to model communication of oblivious dynamic-network models in constant space.

Structure of this paper. The rest of this paper is structured as follows. Section 2 gives an overview of drawbacks arising from a straightforward use of action models in some contexts in multi-agent systems and outlines our solution to overcome such shortcomings. After establishing notation and definitions in Section 3, we explain, in Section 4, an attempt to improve on [10] within the IIS model. Section 5 presents communication pattern models, our modification of action models. Comparison with existing work appears in Section 6, and Section 7 concludes this paper.

2 An Overview of Our Proposal

We first motivate our proposal by pointing out a limitation of action models and the restricted modal product that arises in some contexts when modeling the communication that can happen in a multi-agent system. Roughly speaking, sometimes it is impossible to have action models of “small size”. Our discussion here is informal as we are interested in high-level ideas at the moment, hence delaying formal definitions for the next sections.

The issue. Let us consider the well-known *coordinated attack* problem where two agents a and b wish to schedule an attack. Agent a has two possible preferences for scheduling the attack, n for noon or d for dawn, while agent b has no initial preference and wishes to learn a 's. Communication is unreliable: whenever an agent sends a message, such a message can get lost. The epistemic model M modeling the initial situation before any communication occurs has two worlds, one in which a prefers to attack at dawn and another one in which a prefers to attack at noon; b cannot distinguish between these two worlds. See model M in Fig. 1.

An action model is a generalization of an epistemic model, where vertices, called *events*, are labeled with arbitrary formulas (as opposed to sets of propositional variables) called *preconditions*. The restricted modal product of an epistemic model M and an action model A , denoted $M \otimes A$, is an epistemic model where each world is a pair (w, e) , such that w is a world in M , e is an event in A , and the precondition of e holds in w . Worlds (w, e) and (w', e') are connected with each other for agent a if both w and w' are connected in M for a , and e and e' are connected in A for a . World (w, e) is labeled with the same label as that of w . (Formal definitions of action model and restricted modal product appear in Subsect. 4.2.)

A simple action model A modeling that a sends its preference to b has three events: one for each preference $p \in \{d, n\}$ modeling that b successfully receives the preference, p , of a , with a precondition

specifying that the event can happen only if p is the preference of a , and a third event, modeling that a 's message gets lost, with precondition \top . Agent b can distinguish between all events since it either receives a 's message or not, but a cannot distinguish between events because messages can get lost. See action model A in Fig. 1. The restricted modal product $M \otimes A$ contains four worlds, one for each combination of a 's initial preference and successful/unsucessful communication.

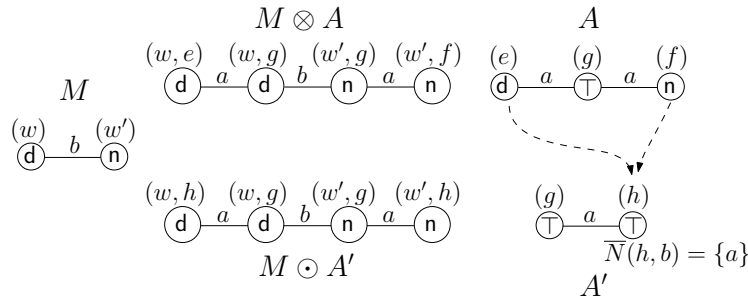


Figure 1: A smaller action model for the coordinated attack problem using our approach.

We observe that the action model A has the following inconvenience. If a has $x > 2$ preferences to schedule the attack instead of only two, a natural generalization of A has $x + 1$ events: A is akin to a star with a “central” event modeling that a 's message gets lost, and one event for each of the possible preferences of a . Thus, the size of the action model is proportional to the size of a 's *input space*.

Can we design a smaller action model for this situation? Can we design an action model with only two events, one corresponding to the case that a 's message gets lost and another corresponding to the case that a 's message (with distinct contents, either d or n) reaches b ? The answer is no. It is easy to see that if we have an action model A' with only one event h corresponding to a 's successful communication, and unavoidably with precondition $d \vee n = \top$ (see A' in Fig. 1, discarding for the moment the set $\bar{N}(h, b) = \{a\}$), then $M \otimes A'$ has again four worlds but now b cannot distinguish between the worlds (w, h) and (w', h) corresponding to the cases where the communication was successful (which is incorrect). A similar situation happens if a has more than two preferences: it is impossible to have an action model with an event that models the case that a 's communication is successful. We cannot get any smaller action model in this situation (it might be possible, however, to do so in further rounds) because the action models are designed to deal with “interpreted” events, namely, an event includes the information of the message encoded in its precondition, hence it has a limited ability to represent that *some* information is sent from one agent to another.

This property of action models is a problem in some contexts. Specifically, when studying computability in a given distributed model, it is often the case that the analysis is performed on *protocols* in which agents proceed in a sequence of *rounds* of communication, and in each round every agent sends *all* the information it has collected so far to all other agents; these protocols are called *full-information* in the distributed-computing literature. To be able to reason in the style of DEL, we would like to have an action model modeling the communication events that can happen in a round, and update the epistemic model with the help of the restricted modal product in each round. Drawbacks of the approach in [10] are that the size of the action models proposed there grows exponentially in the number of round and that such action models are structurally isomorphic to the resulting epistemic models. As we will see later, for the case of two agents, we have been able to find a family of action models with a constant number of actions (although the preconditions of the action model do change from round to round) but it is unclear how to find action models with this property for other cases.

A glimpse of our solution. Coming back to our initial example, how do we “fix” the problem in $M \otimes A'$, i.e., that b cannot distinguish between worlds (w, h) and (w', h) ? Our solution is based on the following observation: b must be able to distinguish between the two worlds because (1) it receives a message from a in the event h of A' , and (2) a can distinguish between the w and w' in M . Therefore, a must send information that makes b able to distinguish between the two worlds in $M \otimes A'$.

We define an extension to the action model formalism, which equips an action model with an additional function \bar{N} that maps every pair (e, a) to a set of agents. Intuitively, $\bar{N}(e, a)$ contains the agents that a receives messages from when the event e happens. The restricted modal product is modified by adding two conditions when updating the accessibility relation of an epistemic model. Such conditions say that an agent a cannot distinguish between two worlds (w, e) and (w', e') if and only if a receives messages from the same set of agents in e and e' (i.e., $\bar{N}(e, a) = \bar{N}(e', a)$) and each of these agents cannot distinguish between w and w' (namely, $\forall a' \in \bar{N}(e, a), w \sim_{a'} w'$). The idea is that if those agents sending information to a cannot distinguish between w and w' , then there is no information they send to a making (w, e) and (w', e') distinguishable to a . The new product is denoted \odot . Using this formalism, for the coordinated attack problem, we are able to define a *communication pattern model* A' with a *single* event (called communication pattern in our context) h corresponding to the case in which a 's message reaches b . In A' , $\bar{N}(h, b)$ is set to $\{a\}$, and \bar{N} is set to \emptyset in any other case. Figure 1 shows the model A' . Furthermore, the action model is correct regardless of the size of a 's input space, meaning that the very same action model produces the desired epistemic model if M represents the situation that a has $x > 2$ initial preferences.

3 Analyzing distributed computing models

In this section, we give some introductory definitions and fix the notation. We assume some familiarity with basic epistemic logic. We refer to the language of multiagent epistemic logic as \mathcal{L}_K . Additionally, we consider a non-empty finite set of agents $Ag = \{a_1, \dots, a_n\}$ and a non-empty finite set of propositions $Props$, unless specified otherwise.

Our models of interest. We are interested in *dynamic-network models* [7, 14, 16], in which a set of $n \geq 2$ failure-free agents proceed in an infinite sequence of *synchronous* rounds of communication. Each agent is a *state machine*. In each round, the communication is specified with a *communication graph*, namely a directed graph whose vertex set is Ag , with each edge (a_i, a_j) indicating that a message from a_i to a_j is successfully delivered in that round. The *in-neighborhood* of an agent a_i in a communication graph G , namely the set of agents a_j such that (a_j, a_i) is an arrow in G , is denoted $N_G^-(a_i)$. Let CP_{Ag} denote the set with all communication graphs with vertex set Ag . Thus, a dynamic-network model Adv is specified with a set of infinite sequences of graphs of CP_{Ag} , that we call *adversary*. Intuitively, we say that an adversary Adv is *oblivious* if in every round, any communication graph in a given set can happen, regardless of the communication graphs that have happened in previous rounds. This is formalized as follows. We say that a finite sequence S of communication graphs is a *prefix* of Adv if S is a prefix of a sequence in Adv . An adversary Adv is oblivious if there exists a non-empty subset $X \subseteq CP_{Ag}$ such that the graphs in X are the prefixes of Adv of length one, and for every finite sequence S that is a prefix of Adv , it holds that $S \cdot G$ is a prefix of Adv , for every graph $G \in X$. Thus, an oblivious adversary is simply specified through the set X of communication graphs; we will say that $Adv = X$.

Protocols. Each agent locally executes a *protocol* that specifies the messages that the agent sends in a round, depending on the local state of the agent at the beginning of the round. Each agent starts the

computation with a private input, which is the state of the agent at the beginning of the first round. Since we are interested in modeling how knowledge can evolve through the computation, we assume that in every round every agent attempts to communicate to everybody all it knows so far. Formally, every agent locally executes the *full-information* protocol, namely, in every round an agent sends to all other agents all the information such an agent has collected so far. Therefore, the full-information protocol captures all that an agent can know in an execution. The full-information protocol is an important tool in distributed-computing computability research.

Executions and configurations. An *execution* E of an adversary Adv is a pair (I, S^∞) , where $I = (v_1, v_2, \dots, v_n)$ is an *input vector* denoting that agent a_i starts with input v_i , with v_i belonging to an *input space*, denoted In , and S^∞ is a sequence of Adv . An r -*execution* of Adv is a pair (I, S) , where I is an input vector and S is a prefix of Adv with $|S| = r$. A *configuration* C is an n -tuple whose i -th position is a local state of agent a_i (thus input vectors are configurations). We say that a_i *does not distinguish* between configurations C and C' if and only if $C(i) = C'(i)$. An r -execution (I, S) *ends* at a configuration C if each agent a_i has the local state $C(i)$ after the execution of the sequence of communication rounds described by S with the inputs stated by I ; alternatively, we say that C is the configuration at the *end* of (I, S) . Note that for the empty sequence, denoted $[], I$ is the configuration at the end of the 0-execution $(I, [])$, for every input vector I .

Our representation. We use *epistemic models*, that we name M^r , for representing the r -executions of a given adversary Adv . An *epistemic model* for Ag and a set of propositions $Props$ is a triple $M = (W, \sim, L)$, where W is a finite set of worlds, $\sim : Ag \rightarrow \wp(W \times W)$ assigns an equivalence relation to each agent, and $L : W \rightarrow \wp(Props)$ assigns a set of true-valued propositions to each world. Each world in M^r represents an r -execution and the accessibility relations represent the indistinguishability relations over the configurations at the end of the r -executions of Adv .

The initial epistemic model (M^0) . We build the initial epistemic model $M^0 = (W^0, \sim^0, L^0)$ for Ag and In with $Props = \{in_{a,v} \mid a \in Ag \wedge v \in In\}$ so that $W^0 = \{I \mid I \text{ is an input vector for } Ag \text{ and } In\}$, $I \sim_{a_i}^0 I'$ if and only if $I(i) = I'(i)$, and $L(I) = \{in_{a_i,v} \in Props \mid I(i) = v\}$. The epistemic model M^0 for the agents $Ag = \{a, b\}$ and binary inputs $In = \{0, 1\}$ is depicted in Fig. 2.

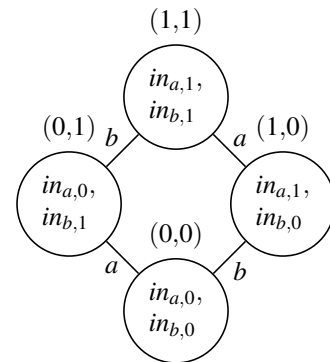


Figure 2: Model M^0 for agents a and b with binary inputs.

4 Action models and the IIS model

In this section, we first present the IIS model. Next, we give the definition of action models. Finally, we exhibit our best action-model solution of modeling IIS for agents a and b with binary inputs.

4.1 Iterated Immediate Snapshot distributed-computing model

The IIS model [5] is a fundamental model that fully captures what can be solved in asynchronous wait-free shared-memory systems with process-crash failures. We can define IIS as a (failure-free) synchronous oblivious dynamic-network adversary. The set describing the adversary is as follows. For every

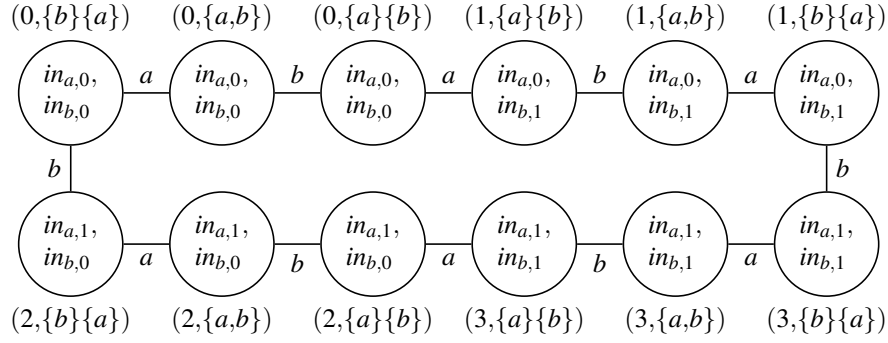


Figure 3: Epistemic model M_{IIS}^1 that represents the configurations at the end of the first round of the full-information protocol for two agents a and b with binary inputs in the IIS model.

sequence of non-empty subsets of Ag , $S = [C_1, C_2, \dots, C_k]$, satisfying that $\text{Ag} = \bigcup C_i$ and $C_i \cap C_j = \emptyset$ whenever $i \neq j$, the adversary has the communication graph with a directed edge (a, b) for every pair of agents $a \in C_i, b \in C_j$ with $1 \leq i \leq j \leq k$. We say that C_i is a *concurrency class*. In Fig. 3, we show the epistemic model M_{IIS}^1 that represents the configurations at the end of the first round of the full-information protocol for processes a and b with binary input in the IIS model.

4.2 Action models

Action models were introduced in [1] as a general way to model dynamics of knowledge via events.

Definition 1 (Action model). *An action model A is a triple (E, R, Pre) , where E is a non-empty finite set of events, $R : P \rightarrow \wp(E \times E)$ is a function that associates each agent with a relation over the set of events, and $\text{Pre} : E \rightarrow \mathcal{L}_K$ is a function that associates each event with a precondition.*

Definition 2 (Syntax). *Let $A = (E, R, \text{Pre})$ be an action model over Ag and Props. The language \mathcal{L}_{\otimes} is given by the following BNF $\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid [(A, e)]\varphi$, where $a \in \text{Ag}$, and $p \in \text{Props}$, $e \in E$ and (A, e) is an update.*

Definition 3 (Restricted modal product). *Let $M = (W, \sim, L)$ be an epistemic model over Ag and Props. Let $A = (E, R, \text{Pre})$ be an action model. $M' = (W', \sim', L') = M \otimes A$ is defined as follows:*

- $W' = \{(w, e) \in W \times E \mid M, w \models \text{Pre}(e)\}$
- $\sim'_a = \{((w, e), (w', e')) \in W' \times W' \mid w \sim_a w' \wedge e R_a e'\}$
- $L'((w, e)) = L(w)$

Definition 4 (Semantics). *Let $M = (W, R, L)$ be an epistemic model over Ag and Props. Let $A = (E, R, \text{Pre})$ be an action model. Let $p \in \text{Props}$ be a proposition. Let $w, w' \in W$ be worlds. Let $a \in \text{Ag}$ be an agent. Let $e \in E$ be an event. Let $\varphi, \psi \in \mathcal{L}_{\otimes}$ be formulas.*

$$\begin{aligned}
M, w \models p & \quad \text{iff } p \in L(w) \\
M, w \models \neg\varphi & \quad \text{iff } M, w \not\models \varphi \\
M, w \models \varphi \wedge \psi & \quad \text{iff } M, w \models \varphi \text{ and } M, w \models \psi \\
M, w \models K_p\varphi & \quad \text{iff } M, w' \models \varphi \text{ for all } w' \text{ such that } w R(p) w' \\
M, w \models [(A, e)]\varphi & \quad \text{iff } M, w \models \text{Pre}(e) \text{ implies } M \otimes A, (w, e) \models \varphi
\end{aligned}$$

4.3 Our best action-model solution for IIS

We now present our best action-model approach of modeling IIS for agents a and b with binary inputs. We exploit the fact that for two-agent IIS, *the epistemic models will always be bipartite graphs*. We can hence partition the set of worlds in M^i into two sets W_1^i and W_2^i so that any pair of distinct worlds in the same set can be distinguished by both agents. For each set W_j^i , we use three events to represent the different sequences of concurrency classes that can happen in a round: $\{a\}\{b\}$, $\{a,b\}$, and $\{b\}\{a\}$ (see Fig. 4). Thus we have six events: three for operating with the worlds in W_1^i , and three for operating with the worlds in W_2^i . The sketch of the action model is shown in Fig. 5. In such a sketch, the preconditions, ϕ_1 and ϕ_2 , change from round to round. ϕ_j is a disjunction of formulas identifying the worlds in W_j^i . A formula identifying a world is a conjunction of the formulas describing the local state of each agent. In Appendix A, we define functions that compute an epistemic logic formula that describes the local state of an agent. For the first round, if we consider $W_1^0 = \{(0,0), (1,1)\}$ and $W_2^0 = \{(0,1), (1,0)\}$, the preconditions are: $\phi_1 = (in_{a,0} \wedge in_{b,0}) \vee (in_{a,1} \wedge in_{b,1})$, and $\phi_2 = (in_{a,0} \wedge in_{b,1}) \vee (in_{a,1} \wedge in_{b,0})$.

This approach appears to be a succinct representation of the full-information execution dynamics. There are, however, still issues. We would like to represent communication defined by an oblivious model *just once* because the allowed communication patterns are the same regardless of the round. All correct action models we have been able to find have preconditions that change from round to round. Moreover, the size of the formulas we get from the ϕ functions grows exponentially in the number of rounds. This suggests that in certain cases, a straightforward application of action models might not be ideal.

We have not been able to find a similar family of action models for three agents. We would need to analyze if the corresponding epistemic models are always n -partite, and how we could join all the needed events. Finding action models for the case of m -ary inputs for $m \geq 3$ would be even harder. Making things worse, the analysis might be different in distinct models: we would need to study each model to take advantage of its own characteristics. All these facts motivated us to look for a different and more appropriate approach.

5 Communication pattern models

Intuitively, a communication pattern model can be viewed as a non-directed graph whose vertices have two labels: a formula and a communication graph.

5.1 Definition of communication pattern models

First, we define our *communication pattern models*. Then, we define the syntax of our language. After that, we define our restricted modal product. Finally, we define our language semantics.

Definition 5 (Communication pattern model). \mathcal{P} is a tuple (CP, R, Pre, \bar{N}) , where CP is a non-empty finite set whose elements are called communication patterns, $R : Ag \rightarrow \wp(CP \times CP)$ is a function that as-

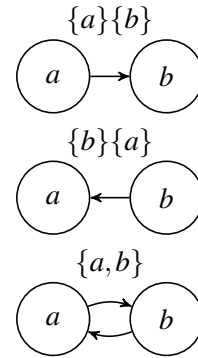


Figure 4: Communication graphs for two-agent IIS.

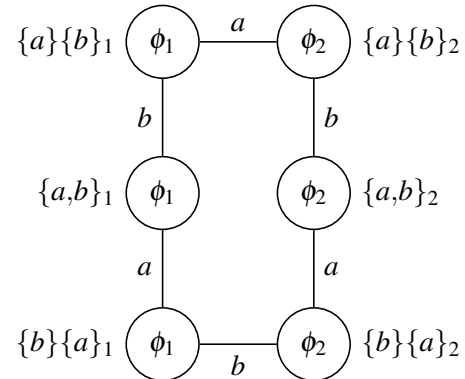


Figure 5: Sketch of the action model for two-agent IIS with binary inputs.

sociates each agent with an equivalence relation over the set of communication patterns, $Pre : CP \rightarrow \mathcal{L}_K$ is a function that associates each communication pattern with a precondition, and $\bar{N} : CP \times Ag \rightarrow \wp(Ag)$ is a function that associates a (communication pattern, agent)-pair with a subset of Ag .

We can think of communication patterns $cp \in CP$ as communication events. The \bar{N} function describes the communication graph associated with a communication pattern: $\bar{N}(cp, a)$ is the in-neighborhood of a in such a communication graph.

Definition 6 (Syntax). *Let \mathcal{P} be a communication pattern model over Ag and Props. The language \mathcal{L}_{\odot} is given by the following BNF:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_a\varphi \mid [(\mathcal{P}, cp)]\varphi$$

where $p \in Props$, $a \in Ag$, $cp \in CP$ and (\mathcal{P}, cp) is an update.

Definition 7 (Restricted modal product). *Let $M = (W, \sim, L)$ be an epistemic model over Ag and Props. Let $\mathcal{P} = (CP, R, Pre, \bar{N})$ a communication pattern model. Let $a \in Ag$ an agent. $(W', \sim', L') = M' = M \odot \mathcal{P}$ is defined as follows:*

- $W' = \{(w, cp) \in W \times CP \mid w \in W \wedge cp \in CP \wedge M, w \models Pre(cp)\}$
- $\sim'_a = \{((w, cp), (w', cp')) \in W' \times W' \mid w \sim_a w' \wedge cp R_a cp' \wedge \underbrace{\bar{N}(cp, a) = \bar{N}(cp', a)}_{\text{underlined}} \wedge \underbrace{w \sim_{a'} w' \forall a' \in \bar{N}(cp, a)}_{\text{underlined}}\}$
- $L'((w, cp)) = L(w)$

Intuitively, the first underlined condition requires agent a to receive information from the same set of processes in both cp and cp' , and the second one requires all processes in such a set to send the same information since such processes are required not to distinguish between w and w' .

Definition 8 (Semantics). *Let $M = (W, \sim, L)$ be an epistemic model over Ag and Props. Let $w, w' \in W$ be worlds. Let $a \in Ag$ be an agent. Let $\mathcal{P} = (CP, R, Pre, \bar{N})$ be a communication pattern model. Let $cp \in CP$ be a communication pattern. Let $\varphi, \psi \in \mathcal{L}_{\odot}$ be formulas.*

$$\begin{aligned} M, w \models p & \quad \text{iff } p \in L(w) \\ M, w \models \neg\varphi & \quad \text{iff } M, w \not\models \varphi \\ M, w \models \varphi \wedge \psi & \quad \text{iff } M, w \models \varphi \text{ and } M, w \models \psi \\ M, w \models K_a\varphi & \quad \text{iff } M, w' \models \varphi \text{ for all } w' \text{ such that } w \sim_a w' \\ M, w \models [(\mathcal{P}, cp)]\varphi & \quad \text{iff } M, w \models Pre(cp) \text{ implies } M \odot \mathcal{P}, (w, cp) \models \varphi \end{aligned}$$

Action models and communication pattern models. Communication pattern models are at least as general as action models. Notice that we can build a *degenerate* communication pattern model given an action model. Let $A = (E, R, Pre)$ be an action model. We build a communication pattern model $\mathcal{P} = (E, R, Pre, \bar{N})$ so that $\bar{N}(e, a) = \emptyset \forall (e, a) \in E \times Ag$. It is easy to see that $M \otimes A = M \odot \mathcal{P}$ holds.

5.2 Communication pattern models for arbitrary adversaries

Consider any adversary Adv and the initial model M^0 defined in Section 3. Here we define an infinite sequence $\mathcal{P}^1, \mathcal{P}^2, \dots$ of communication pattern models that succinctly model the evolution of knowledge in the executions of Adv . More precisely, Theorem 1 in the next section will show that the epistemic model $M^r = (W^r, \sim^r, L^r) = M^0 \odot \mathcal{P}^1 \odot \mathcal{P}^2 \odot \dots \odot \mathcal{P}^r$ captures how knowledge changes after r rounds of communication.

For every $i \geq 1$, the communication pattern model $\mathcal{P}^i = (CP^i, R^i, Pre^i, \bar{N}^i)$ is defined as follows:

- $CP^i = \{cp \in CP_{Ag} \mid \exists \text{ an } i\text{-execution } (I, S \cdot cp) \text{ of } Adv\}$.
- For every $a \in Ag$, $R_a^i = \{(cp, cp') \in CP^i \times CP^i \mid N_{cp}^-(a) = N_{cp'}^-(a)\}$.
- For every $(cp, a) \in CP^i \times Ag$, $\bar{N}^i(cp, a) = N_{cp}^-(a)$.
- For every $cp \in CP^i$, let $\mathscr{W}_{cp}^{i-1} = \{(I, S) \mid \exists \text{ an } i\text{-execution } (I, S \cdot cp) \text{ of } Adv\}$. Thus, $Pre^i(cp) = \bigvee_{(I, S) \in \mathscr{W}_{cp}^{i-1}} \varphi(I, S)$, where $\varphi(I, S) = \bigvee_{0 \leq i \leq n} \varphi_i(a_i, C(i))$. See Appendix A for the definition of φ_i .

The case of oblivious dynamic-network models. Following the definition of CP^i , we can see that, for any oblivious adversary Adv , $CP^i = Adv$, for each $i \geq 1$. Thus, all \mathscr{P}^i have the same set of communication patterns. Moreover, for each $cp \in CP^i$, \mathscr{W}_{cp}^{i-1} contains *all* $(i-1)$ -executions of Adv , and hence $Pre^i(cp)$ can be set to \top . Therefore, $\mathscr{P}^1 = \mathscr{P}^2 = \dots$. The communication pattern model representing dynamics for IIS with agents a and b is depicted in Fig. 6. For clarity, the function \bar{N} is not depicted; however, it can be obtained from the in-neighborhoods of the communication graphs in Fig. 4. It is worth observing that the communication pattern in Fig. 6, omitting \bar{N} , and the usual modal product \otimes do not model IIS for two agents, not even for the first round. Namely, $M^1 = M^0 \otimes \mathscr{P}_{\text{two-IIS}}$ has “undesirable” pairs in agent relations which make M^1 structurally different from a 12-cycle, which is the structure of the epistemic model for two processes with binary inputs after one round of communication in IIS (see Fig. 3).

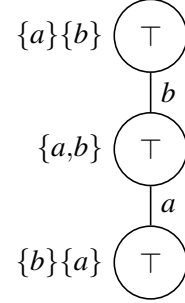


Figure 6: Communication pattern model $\mathscr{P}_{\text{two-IIS}}$ for two-agent IIS.

5.3 The \odot product reflects the change in local states through rounds

The dynamic epistemic logic that we present is focused on reasoning about computations. In particular, we are interested in modeling how *configurations* change in the full-information protocol. A key point is that when updating an epistemic model with our modal product, the resulting epistemic model models how the local states of agents change. Theorem 1 below states that our communication pattern models do model knowledge dynamics. The theorem formalizes this claim using the following notion.

Let Adv be an adversary. For every $i \geq 0$, we define the set $\mathscr{C}_{Adv}^i = \{C \mid \text{there is an } i\text{-execution } (I, S) \text{ of } Adv \text{ that ends in the configuration } C\}$. Let $\mathscr{P}^1, \mathscr{P}^2, \dots$ be an infinite sequence of communication pattern models. We say that the sequence $\mathscr{P}^1, \mathscr{P}^2, \dots$ *reflects* the adversary Adv if for each $r \geq 1$, there is a bijection $f^r : W^r \rightarrow \mathscr{C}_{Adv}^r$ such that $w \sim_{a_i} w'$ if and only if a_i does not distinguish between $f^r(w)$ and $f^r(w')$, where M^0 is the initial epistemic model and $M^r = (W^r, \sim^r, L^r) = M^0 \odot \mathscr{P}^1 \odot \mathscr{P}^2 \odot \dots \odot \mathscr{P}^r$. If $\mathscr{P}^1 = \mathscr{P}^2 = \dots$, we simply say that \mathscr{P}^1 reflects Adv .

Theorem 1 (Main result). *Let Adv be an adversary and $\mathscr{P}^1, \mathscr{P}^2, \dots$ be the communication pattern models built from Adv , as described in Subsection 5.2. Then, $\mathscr{P}^1, \mathscr{P}^2, \dots$ reflects Adv .*

Let \mathscr{E}_{Adv}^r be the set of all r -executions of Adv . Let I, I' be two input vectors for Ag and In . The proof of Theorem 1 will be as follows. First, we will present two lemmas whose proof we omit because of space restrictions. Then, we will prove by induction that $w_r \sim_{a_i} w'_r$ if and only if a_i does not distinguish between $f^r(w_r)$ and $f^r(w'_r)$.

Consider $E_{r+1} = (I, [cp_1, cp_2, \dots, cp_r, cp_{r+1}]) \in \mathscr{E}_{Adv}^{r+1}$, and $E_r = (I, [cp_1, cp_2, \dots, cp_r]) \in \mathscr{E}_{Adv}^r$. We define $g^r : \mathscr{E}_{Adv}^r \rightarrow \mathscr{E}_{Adv}^r$ as follows:

$$g^0((I, [])) = I.$$

$$g^{r+1}(E_{r+1}) = C_{r+1} = (C_{r+1}(1), C_{r+1}(2), \dots, C_{r+1}(n))$$

where

$$C_{r+1}(i)(j) = \begin{cases} g^r(E_r)(j) & \text{if } a_j \in N_{cp_{r+1}}^-(a_i) \cup \{a_i\} \\ \perp & \text{otherwise} \end{cases}.$$

Lemma 1. g^r is a bijection.

Consider $w_r = (\dots((I, cp_1), cp_2) \dots, cp_r) \in W^r$. We define $h^r : W^r \rightarrow \mathcal{E}_{Adv}^r$ as follows:

$$h^r(w_r) = (I, [cp_1, cp_2, \dots, cp_r]).$$

Lemma 2. h^r is a bijection.

Now, we start with the proof of Theorem 1.

Proof. We define

$$f^r : W^r \rightarrow \mathcal{C}^r = g^r \circ h^r.$$

Since g^r and h^r are bijective, f^r is bijective.

Now we prove, by induction on the round number r , that the epistemic model M^r reflects indistinguishability between configurations.

Base case.

Consider $I, I' \in W^0$, $C_I = f^0(I) = (I(1), I(2), \dots, I(n))$, and $C_{I'} = f^0(I') = (I'(1), I'(2), \dots, I'(n))$. By construction of \sim^0 , $I \sim_{a_i}^0 I'$ if and only if $I(i) = I'(i)$ holds. Since a_i does not distinguish between C_I and $C_{I'}$ if and only if $I(i) = I'(i)$ holds, $I \sim_{a_i}^0 I'$ if and only if a_i does not distinguish between C_I and $C_{I'}$ holds.

Inductive hypothesis.

Consider $M^r = (W^r, \sim^r, L^r) = M^0 \odot \mathcal{P}^1 \odot \mathcal{P}^2 \odot \dots \odot \mathcal{P}^r$, and $w_r, w'_r \in W^r$. We assume that $f_r : W^r \rightarrow \mathcal{C}_{Adv}^r$ satisfies that $w_r \sim_{a_i}^r w'_r$ if and only if a_i does not distinguish between $f^r(w_r)$ and $f^r(w'_r)$.

Inductive step.

Consider $w_{r+1} = (w_r, cp_{r+1})$, $w'_{r+1} = (w'_r, cp'_{r+1}) \in W^{r+1}$. We need to prove that $w_{r+1} \sim_{a_i}^{r+1} w'_{r+1}$ if and only if a_i does not distinguish between $f^{r+1}(w_{r+1})$ and $f^{r+1}(w'_{r+1})$.

Consider $w_{r+1}, w'_{r+1} \in W^{r+1}$. By definition of f^{r+1} , we know that

$$f^{r+1}(w_{r+1}) = C_{r+1} = (C_{r+1}(1), C_{r+1}(2), \dots, C_{r+1}(n))$$

where

$$C_{r+1}(i)(j) = \begin{cases} f^r(w_r)(j) & \text{if } a_j \in \bar{N}^{r+1}(cp_{r+1}, a_i) \cup \{a_i\} \\ \perp & \text{otherwise} \end{cases}$$

and

$$f^{r+1}(w'_{r+1}) = C_{r+1}' = (C_{r+1}'(1), C_{r+1}'(2), \dots, C_{r+1}'(n))$$

where

$$C_{r+1}'(i)(j) = \begin{cases} f^r(w'_r)(j) & \text{if } a_j \in \bar{N}^{r+1}(cp'_{r+1}, a_i) \cup \{a_i\} \\ \perp & \text{otherwise} \end{cases}.$$

By the definition of \odot , $w_{r+1} \sim_{a_i}^{r+1} w'_{r+1}$ if and only if $w_r \sim_{a_i}^r w'_r$, $cp_{r+1} R_{a_i}^{r+1} cp'_{r+1}$, $\bar{N}^{r+1}(cp_{r+1}, a_i) = \bar{N}^{r+1}(cp'_{r+1}, a_i)$, and $w_r \sim_{a_j}^r w'_r \forall a_j \in \bar{N}^{r+1}(cp_{r+1}, a_i)$.

$C_{r+1}(i)(j) = \perp$ if and only if $C_{r+1}' = \perp$ holds because by construction of \mathcal{P}^{r+1} , $cp_{r+1} R_{a_i}^{r+1} cp'_{r+1}$ if and only if $\bar{N}^{r+1}(cp_{r+1}, a_i) = \bar{N}^{r+1}(cp'_{r+1}, a_i)$ holds. Then, $C_{r+1}(i)(j) = C_{r+1}'(i)(j)$ holds if and only if $f^r(w_r)(j) = f^r(w'_r)(j)$ holds for all agents in $\bar{N}^{r+1}(cp_{r+1}, a_i)$. By the inductive hypothesis, we have

that $f^r(w_r)(j) = f^r(w'_r)(j) \forall a_j \in \overline{N}^{r+1}(cp_{r+1}, a_i)$ holds. Then, $w_{r+1} \sim_{a_i}^{r+1} w'_{r+1}$ holds if and only if $C_{r+1}(i)(j) = C'_{r+1}(i)(j)$. $C_{r+1}(i)(j) = C'_{r+1}(i)(j)$ holds if and only if $C_{r+1}(i) = C'_{r+1}(i)$ holds. Hence, $C_{r+1}(i) = C'_{r+1}(i)$ holds if and only if a_i does not distinguish between C_{r+1} and C'_{r+1} . \square

Corollary 1 (Constant space). *Modeling an oblivious adversary Adv with communication pattern models require constant space.*

Proof. Let \mathcal{P} be the communication pattern model for Adv built as described in Subsection 5.2. By Theorem 1, \mathcal{P} reflects Adv. Moreover, \mathcal{P} remains the same in all rounds. \square

6 Related work

The formal treatment of knowledge in distributed computing was pioneered by Halpern and Moses in [11]. Perhaps their most important result is having proved that common knowledge amounts to simultaneity. The book by Fagin, Halpern, Moses, and Vardi [9] was pivotal, as it summarized numerous results and compared different approaches to studying many aspects of knowledge in a system of agents.

Action models first appeared in [1]. Such a formalism, however, was only considered for modeling evolution of knowledge in distributed systems, as far as we know, in [10], by Goubault, Ledent, and Rajsbaum and in [17], by Pflieger and Schmid.

Closer to our work is [10], where the authors exhibit a tight connection between the topological approach [12] to distributed processing and Kripke models. A second contribution of [10] is employing the *restricted modal product* operator of action models to model knowledge change between agents after a round. A third important result is employing action models to represent “tasks”. A task is the equivalent of a function in distributed computability. The task defines the possible inputs to the agents, and for each set of inputs, it specifies the set of outputs that the agents may produce. By representing the task itself, the possibility of solving a task amounts to the existence of a certain simplicial map.

The objective of [17], which uses action models as well, is that of obtaining lower limits on the number of bits necessary for implementing a protocol that is specified with an initial epistemic model and an infinite sequence of action models that describe how the epistemic model is updated through an infinite sequence of communication rounds. Like us, [17] uses dynamic-network models. Unlike us, [17] assumes that the action model are given. As a result, [17] does not build an action model and is not concerned with the size of the action models.

The work in [2] exhibits drawbacks similar to the ones we found when using the action model framework in other contexts. The authors propose an extension of epistemic models adding a function and an update mechanism. Adding such a function decreases the number of events needed to represent certain problems. Our proposal, however, can be directly applied to the context of distributed systems by the communication between agents.

7 Concluding Remarks

The formalization of knowledge in the distributed-computing literature has still to have a more significant impact. The evidence is that many papers in distributed computing refer to knowledge informally.

At the same time, in the epistemic-logic literature, the formalism of action models has emerged as an important mechanism for modeling the evolution of knowledge. Hence, the works by Goubault, Ledent,

and Rajsbaum [10], establishing a connection between action models and a topological approach to distributed systems, and by Pflieger and Schmid [17], modeling a dynamic-network protocol by an initial epistemic model and an infinite sequence of action models, are relevant.

The approach of [10] operates an action model with an epistemic model capturing knowledge at a certain point in time, to obtain a new epistemic model for knowledge after one round of communication. We observed however, that the action models proposed in [10] for IIS have certain inconveniences. Such action models are structurally isomorphic to the desired epistemic model, hence the number of events grows exponentially in the round number.

We proposed a family of action models with six events, for the case of two agents with binary inputs, whose preconditions change from round to round. For obtaining such a family however, we needed to know in advance the structure of the epistemic models in further rounds. Furthermore, the analysis for more agents or even more inputs seems to be more difficult. Hence, a generalization of such a family is unclear for IIS. Moreover, the analysis would depend on how the epistemic models change in different distributed-computing models.

To overcome these disadvantages, we proposed an extension of action models for dealing with communication patterns, called *communication pattern models*. Our models work for a large variety of distributed-computing models, called dynamic-network models. Using our extension, we were able define communication pattern models systematically for every round of execution in the full-information protocol. In the case of oblivious models, which includes IIS, the communication pattern model remains the same all through the computation. In either case, our approach can be applied in automated distributed-systems verification. We emphasize the fact that communication pattern models as presented in this work are designed to deal with the full-information protocol. We plan to analyze modifying definitions to deal with arbitrary protocols.

Communication pattern models were presented as an extension of action models. It is possible, however, to present the same idea with a set of communication graphs. When analyzing arbitrary dynamic-network models, there should be a precondition for each communication pattern. When analyzing oblivious models there is no need of such precondition because it is always true. An advantage of presenting communication pattern models as an extension of action models is that of studying how an action model can be seen in an agent-communication perspective.

An alternative approach to modeling distributed systems epistemically is by the use of *interpreted* systems, as in [11], or in the more recent papers by Castañeda, Gonczarowski, and Moses [6], as well as Moses [15]. In these works, protocols are modeled explicitly, and indistinguishability is generated directly from the local states; consequently there is no need for a communication pattern model (or an action model) that models the dynamics of the system. Since we use epistemic models and communication pattern models, we need to show that the indistinguishability relation that they generate coincides with the one based on local states in the corresponding model, which is shown in Theorem 1. A benefit of our approach, however, is that the communication pattern models that we compute are arguably a succinct representation of the communication that can occur in a model.

8 Acknowledgment

We should like to thank Hans van Ditmarsch for his insightful comments.

References

- [1] Alexandru Baltag, Lawrence S. Moss & Slawomir Solecki (1998): *The logic of public announcements, common knowledge, and private suspicions*. In: *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 1998)*, pp. 43–56. Available at <http://dl.acm.org/citation.cfm?id=645876.671885>.
- [2] Adam Bjorndahl & Will Nalls (2021): *Endogenizing Epistemic Actions*. *Studia Logica*, doi:10.1007/s11225-020-09937-8.
- [3] Elizabeth Borowsky & Eli Gafni (1993): *Generalized FLP impossibility result for t-resilient asynchronous computations*. In: *Proceedings of the Twenty-Fifth ACM Symposium on Theory of Computing (STOC 1993)*, pp. 91–100, doi:10.1145/167088.167119.
- [4] Elizabeth Borowsky & Eli Gafni (1993): *Immediate atomic snapshots and fast renaming (extended abstract)*. In: *Proceedings of the Twelfth Annual ACM Symposium on Principles of Distributed Computing (PODC 1993)*, pp. 41–51, doi:10.1145/164051.164056.
- [5] Elizabeth Borowsky & Eli Gafni (1997): *A simple algorithmically reasoned characterization of wait-free computation (extended abstract)*. In: *Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing (PODC 1997)*, ACM, pp. 189–198, doi:10.1145/259380.259439.
- [6] Armando Castañeda, Yannai A. Gonczarowski & Yoram Moses (2014): *Unbeatable Consensus*. In Fabian Kuhn, editor: *Distributed Computing*, Springer Berlin Heidelberg, p. 91–106, doi:10.1007/978-3-662-45174-8_7.
- [7] Bernadette Charron-Bost & André Schiper (2009): *The Heard-Of Model: Computing in Distributed Systems with Benign Faults*. *Distributed Comput.* 22(1), pp. 49–71, doi:10.1007/s00446-009-0084-6.
- [8] Hans van Ditmarsch, Wiebe van der Hoek & Barteld Kooi (2008): *Dynamic Epistemic Logic*. Springer, doi:10.1007/978-1-4020-5839-4.
- [9] Ronald Fagin, Joseph Y. Halpern, Yoram Moses & Moshe Y. Vardi (1995): *Reasoning about Knowledge*. MIT Press, doi:10.7551/mitpress/5803.001.0001.
- [10] Eric Goubault, Jérémy Ledent & Sergio Rajsbaum (2018): *A simplicial complex model for dynamic epistemic logic to study distributed task computability*. In: *Proceedings of the Ninth International Symposium on Games, Automata, Logics, and Formal Verification (GandALF 2018)*, pp. 73–87, doi:10.4204/EPTCS.277.6.
- [11] Joseph Y. Halpern & Yoram Moses (1984): *Knowledge and common knowledge in a distributed environment*. In: *Proceedings of the Third ACM Symposium on Principles of Distributed Computing (PODC 1984)*, pp. 50–61, doi:10.1145/79147.79161.
- [12] Maurice Herlihy, Dmitry Kozlov & Sergio Rajsbaum (2014): *Distributed Computing Through Combinatorial Topology*. Morgan-Kaufmann, doi:10.1016/C2011-0-07032-1.
- [13] Maurice Herlihy & Nir Shavit (1993): *The asynchronous computability theorem for t-resilient tasks*. In: *Proceedings of the Twenty-Fifth ACM Symposium on Theory of Computing (STOC 1993)*, pp. 111–120, doi:10.1145/167088.167125.
- [14] Fabian Kuhn & Rotem Oshman (2011): *Dynamic networks: models and algorithms*. *SIGACT News* 42(1), pp. 82–96, doi:10.1145/1959045.1959064.
- [15] Yoram Moses (2016): *Relating Knowledge and Coordinated Action: The Knowledge of Preconditions Principle*. *Electronic Proceedings in Theoretical Computer Science* 215, p. 231–245, doi:10.4204/eptcs.215.17.
- [16] Thomas Nowak, Ulrich Schmid & Kyrill Winkler (2019): *Topological Characterization of Consensus under General Message Adversaries*. In Peter Robinson & Faith Ellen, editors: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC 2019)*, ACM, pp. 218–227, doi:10.1145/3293611.3331624.
- [17] Daniel Pflieger & Ulrich Schmid (2018): *On knowledge and communication complexity in distributed systems*. In: *International Colloquium on Structural Information and Communication Complexity (SIROCCO 2018)*, Springer, pp. 312–330, doi:10.1007/978-3-030-01325-7_27.

- [18] Michael Zacks & Fotios Zaharoglou (1993): *Wait-free k-set agreement is impossible: the topology of public knowledge*. In: *Proceedings of the Twenty-Fifth ACM Symposium on Theory of Computing (STOC 1993)*, pp. 101–110, doi:10.1145/167088.167122.

A Views and epistemic formulas

Here, we first show a way of thinking about local states in distributed computing called *views*. We then give a formal way of representing such views with an epistemic-logic formula.

In the distributing-computing literature, it is common to regard the local states of the agents as their views. We can think of a view of an agent as a single variable whose value changes from round to round. Such a view takes different values depending on the round.

Definition 9 (View). *Consider $S_k = [cp_1, cp_2, \dots, cp_k]$, and $S_{k+1} = S_k \cdot cp_{k+1}$ so that (I, S_{k+1}) is a $k+1$ -execution. The view of an agent a_i in a execution (I, S) , $view(a_i, (I, S))$ for short, in the full-information protocol is defined inductively as follows:*

$$view(a_i, (I, [])) = I(i).$$

$$view(a_i, (I, S_{k+1})) = [view[1], view[2], \dots, view[n]]$$

where

$$view[j] = \begin{cases} view(a_j, (I, S_k)) & \text{if } a_j \in N_{cp_{k+1}}^- \cup \{a_i\} \\ \perp & \text{otherwise} \end{cases}$$

In the full-information protocol, each agent tries to communicate its whole local state to the other agents. If a_i receives a message from a_j , a_i will know all that a_j knew in the previous round, otherwise a_i will not be able to know what a_j could know.

Now, we formalize the notion of views building an epistemic logic formula for the view of a_i .

Definition 10. *Let $Views^k$ be the set of all possible views of the agents in Ag at the end of the k -th round. Let $Views_i^k$ be the set of all possible views of the agent a_i at the end of the k -th round. Consider $view = [view[1], view[2], \dots, view[n]] \in Views^{k+1}$. We define the functions $\varphi_k : P \times Views^k \rightarrow \mathcal{L}_K$, for all $k \in \mathbb{N} \cup \{0\}$ as follows:*

$$\varphi_0(a_i, v) = in_{a_i, v}.$$

where $v \in In$.

$$\varphi_{k+1}(a_i, view) = \bigwedge_{j=1}^n \begin{cases} K_{a_i}(\varphi_k(a_j, view[j])) & \text{if } view[j] \neq \perp \\ \bigwedge_{view' \in Views_i^k} \neg K_{a_i}(\varphi_k(a_j, view')) & \text{otherwise} \end{cases}$$