Logic for Communicating Automata with Parameterized Topology

Benedikt Bollig

LSV, ENS Cachan, CNRS & Inria, France
bollig@lsv.ens-cachan.fr

Abstract. Communicating automata (CA) are a fundamental model of systems where a fixed finite number of processes communicate via message exchange through FIFO channels. In this paper, we introduce a parameterized version of CA (PCA). The parameter is the underlying communication topology, in which processes are linked via interfaces and arranged as graphs of bounded degree such as ranked trees, grids, rings, or pipelines. A given PCA can be run on any such topology. We provide Büchi-Elgot-Trakhtenbrot theorems for PCA, continuing the logical study that has established characterizations of classical CA in terms of (fragments of) monadic second-order (MSO) logic. In particular, we give translations of existential MSO logic to PCA that are correct for large and natural classes of topologies. Our main result relies on a locality theorem for first-order logic due to Schwentick and Barthelmann, and it uses, as a black-box, a construction by Genest, Kuske, and Muscholl from the non-parameterized case.

1 Introduction

The Büchi-Elgot-Trakhtenbrot theorem states that finite automata and monadic second-order (MSO) logic over words are expressively equivalent [5, 7, 18]. This connection between automata and logic constitutes one of the cornerstones in theoretical computer science, as it bridges the gap between high-level specifications and operational system models. Various extensions of that result followed, providing logical characterizations of tree automata [16], asynchronous automata [19], and graph acceptors [17], to mention just a few.

In this paper, we study an analogous question for communicating automata (CA), where finite-state machines can exchange messages through FIFO channels by performing send and receive actions. A single execution of a CA is captured by a message sequence chart (MSC), a directed acyclic graph visualizing the message flow. Its edges connect send events with corresponding receives, as well as successive events performed by a process. Actually, CA have been well studied in the case where the communication topology, which provides a set of processes and channels between them, is *fixed*. Logical characterizations in terms of (fragments of) MSO logic over MSCs have been given for unrestricted CA [2] and channel-bounded CA [13, 14, 10].

Here, we aim for a solution in a parameterized setting where the parameter is the communication topology. As a first step, we define *parameterized* CA (PCA), which can be run on an arbitrary communication topology of bounded degree (there is a fixed finite set of interface names). Our goal is a translation of a logic formula φ, interpreted over MSCs, into a PCA $\mathcal{A}$ that is *correct for* a large class $\mathfrak{T}$ of topologies: when $\mathcal{A}$ is run on topology $\mathcal{T} \in \mathfrak{T}$, then it accepts precisely the MSCs over $\mathcal{T}$ that are a model of φ.

As a first result, we show that the existential fragment of MSO logic (EMSO) involving message and process edges can be translated into PCA that are correct for all *prime* topologies. A topology is prime if, roughly speaking, topologies do not admit cycles with interface-name labels w^n for some $n \geq 2$. This captures pipelines, grids, and trees. The result relies on the construction from [2].

Our main result applies to channel-bounded systems and uses a different technique. It allows for the use of a richer logic, which can access the transitive closure of the process relation. We show that every EMSO formula can be translated into a PCA that is correct for any *unambiguous class* of topologies. A set of topologies is unambiguous if a cycle in one topology gives rise to a cycle in any other topology of that class. This is satisfied by the classes of pipeline, tree, and grid topologies, as well as "almost all" ring topologies. Our construction is based on the Schwentick-Barthelmann normal form of first-order formulas [15], which allows us to evaluate a formula over bounded portions of an MSC. We can then apply a result by Genest, Kuske, and Muscholl in the non-parameterized case [10]. Roughly speaking, we construct a bounded number of classical CA and glue them together towards a parameterized one. The main difficulty is the local evaluation of the underlying topology. Similar techniques are known from distributed algorithms [12], but they do not seem to be directly applicable here.

Finally, we show that we cannot hope for anything better as far as the logic is concerned. Indeed, the above translations are not possible any longer as soon as we add the transitive closure of the full edge relation to the logic.

Related Work. It seems that neither PCA nor expressiveness of parameterized systems in general in terms of logic have been considered in the literature. However, there is a lot of ongoing research on parameterized verification, which aims at showing that a *given* system is correct independently of the number of processes or the communication topology [4, 11, 1, 3, 6]. Our approach is different, since we *generate* from a high-level specification a system model that is correct for a class of topologies. Moreover, there have been a variety of automata constructions that exploit normal forms of first-order logic [17, 15, 8]. We actually borrow a technique from [8], but the overall construction is quite different.

Outline. In Section 2, we define network topologies and MSCs. Sections 3 and 4 introduce PCA and MSO logic. In Section 5, we show that any local EMSO formula can be translated into a PCA that is correct for all prime topologies. Section 6 presents our main result: a translation of EMSO logic with process order to channel-bounded PCA that are correct for unambiguous classes of topologies. In Section 7, we argue that this result is optimal. We conclude in Section 8.

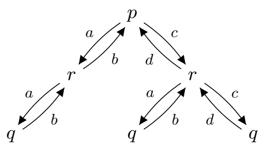

Fig. 1. Tree topology

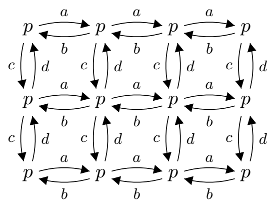

Fig. 2. Grid $\mathcal{T}^{3,4}_{\mathrm{grid}}$

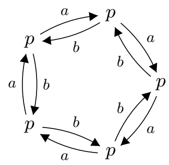

Fig. 3. Ring $\mathcal{T}^{5}_{\mathrm{ring}}$

2 Preliminaries

Communication Topologies. A (communication) topology consists of a finite set of nodes, each having a certain process type. A parameterized communicating automaton can run a process, of appropriate type, on each node. Every process has a finite number of interfaces, which allows it to talk to other processes. If there is an edge $u \xrightarrow{a} v$ in the topology, then u can talk to v via the interface name a. More precisely, it may execute $!a$ and send a message to v, or execute $?a$ and receive a message from v. So let $\mathcal{P} = \{p, q, r, \ldots\}$ and $\mathcal{N} = \{a, b, c, \ldots\}$ be nonempty finite sets of *process types* and *(interface) names*, respectively. We require that every name $a \in \mathcal{N}$ has a *dual* $\overline{a} \in \mathcal{N}$ such that $\overline{a} \neq a$ and $\overline{\overline{a}} = a$.

Definition 1. *A* topology *over $\mathcal{P}$ and $\mathcal{N}$ is a triple $\mathcal{T} = (V, \rightarrow, \pi)$ where V is a nonempty finite set of vertices (also called processes), $\rightarrow \subseteq V \times \mathcal{N} \times V$ (where we write $u \xrightarrow{a} v$ for $(u, a, v) \in \rightarrow$), and $\pi : V \rightarrow \mathcal{P}$ associates with each vertex a process type. We require that, for all $u, v, v' \in V$ and $a, b \in \mathcal{N}$,*

- *$u \xrightarrow{a} v$ implies $u \neq v$,*
- *$u \xrightarrow{a} v$ iff $v \xrightarrow{\overline{a}} u$, and*
- *if $u \xrightarrow{a} v$ and $u \xrightarrow{b} v'$, then $a = b$ iff $v = v'$.*

Note that nodes with the same process type do not need to have the same sets of outgoing interface names. We usually consider topologies up to isomorphism. The set of all topologies (over $\mathcal{P}$ and $\mathcal{N}$) is denoted by $\mathbb{T}$.

Example 1. Figures 1–4 show some example topologies over $\mathcal{P} = \{p, q, r\}$ and $\mathcal{N} = \{a, b, c, d\}$, where the dual of a is b and the dual of c is d, and vice versa. Figure 1 depicts a binary-tree topology. It has distinguished process types for the root (p), leaves that are not the root (q), and inner nodes (r). Interface a points to the left successor (if it exists) and c to the right successor. Figure 2 illustrates the grid topology $\mathcal{T}^{3,4}_{\mathrm{grid}}$. In general, topology $\mathcal{T}^{m,n}_{\mathrm{grid}}$ is uniquely given by its number $m \geq 1$ of rows and its number $n \geq 1$ of columns. Figure 3 shows the ring topology $\mathcal{T}^{5}_{\mathrm{ring}}$. In topology $\mathcal{T}^{n}_{\mathrm{ring}}$, $n \geq 3$ nodes of type p are arranged in a ring. Topology $\mathcal{T}^{8}_{\mathrm{lin}}$ from Figure 4 has eight nodes and a linear (or, pipeline) structure. Note that there are a distinguished first and last node with process types p and q, respectively, while other nodes are of type r. In the following, let $\mathfrak{T}_{\mathrm{lin}} = \{\mathcal{T}^{n}_{\mathrm{lin}} \mid n \geq 2\}$, $\mathfrak{T}_{\mathrm{grid}} = \{\mathcal{T}^{m,n}_{\mathrm{grid}} \mid m, n \geq 1\}$, and $\mathfrak{T}_{\mathrm{ring}} = \{\mathcal{T}^{n}_{\mathrm{ring}} \mid n \geq 3\}$. Moreover, let $\mathfrak{T}_{\mathrm{tree}}$ be the set of binary-tree topologies. Note that $\mathfrak{T}_{\mathrm{lin}} \subseteq \mathfrak{T}_{\mathrm{tree}}$.

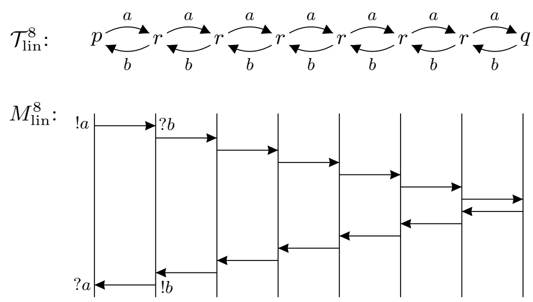

Fig. 4. Topology $\mathcal{T}^8_{\text{lin}}$ and MSC M^8_{lin} over $\mathcal{T}^8_{\text{lin}}$

Message Sequence Charts. The semantics of both an automaton and a logic formula will be defined as a set of messages sequence charts (MSCs). Each MSC depicts a single execution of a communicating system. It is formalized as a labeled directed acyclic graph whose nodes, the *events*, are associated with nodes from a given communication topology. Events are connected by message edges and process edges. The process edges define a total order for each node in the topology, and message edges obey a FIFO policy.

Definition 2. *An* MSC *over topology* $\mathcal{T} = (V, \rightarrow, \pi) \in \mathbb{T}$ *is a tuple* $M = (E, \lhd, \ell, \lambda)$ *where* E *is the nonempty finite set of* events, $\lhd \subseteq E \times E$ *is the acyclic edge relation, which is partitioned into* $\lhd_{\mathsf{proc}}$ *and* $\lhd_{\mathsf{msg}}$*, the mapping* $\ell : E \rightarrow V$ *determines the location of an event in the topology, and* $\lambda : E \rightarrow \{\, !a\,, ?a \mid a \in \mathcal{N}\}$ *determines its labeling.*[1] *Let* $E_! \stackrel{\text{def}}{=} \{e \in E \mid e \lhd_{\mathsf{msg}} f$ *for some* $f \in E\}$ *and, accordingly,* $E_? \stackrel{\text{def}}{=} \{e \in E \mid f \lhd_{\mathsf{msg}} e$ *for some* $f \in E\}$*. Moreover, given* $u \in V$*, let* $E_u \stackrel{\text{def}}{=} \{e \in E \mid \ell(e) = u\}$*. We require that the following hold:*

1. $E = E_! \uplus E_?$,
2. *for all* $u \in V$, $\lhd_{\mathsf{proc}} \cap (E_u \times E_u)$ *is the direct-successor relation of some total order on* E_u,
3. *for all* $(e, f) \in \lhd_{\mathsf{msg}}$*, there is* $a \in \mathcal{N}$ *such that* $\ell(e) \xrightarrow{a} \ell(f)$ *and* $(\lambda(e), \lambda(f)) = (!a, ?\overline{a})$,
4. *for all* $u, v \in V$, $e, e' \in E_u$*, and* $f, f' \in E_v$ *such that* $e \lhd_{\mathsf{msg}} f$ *and* $e' \lhd_{\mathsf{msg}} f'$*, we have* $e \lhd^*_{\mathsf{proc}} e'$ *iff* $f \lhd^*_{\mathsf{proc}} f'$ *(FIFO).*

We do not distinguish isomorphic MSCs over $\mathcal{T}$*.*

Example 2. Figure 4 illustrates the MSC M^8_{lin} over topology $\mathcal{T}^8_{\text{lin}} \in \mathfrak{T}_{\text{lin}}$. It is represented by several top-down process lines, one for each vertex in the topology. Arrows between process lines determine the relation $\lhd_{\mathsf{msg}}$, which connects a send event with a receive event. Note that we included only some of the event labelings. We may consider M^n_{lin} as the execution of a P2P protocol: a request from p is forwarded by $n-2$ processes of type r, until it reaches q. Afterwards, an acknowledgement is relayed back to p along the same way backwards.

[1] The labeling can be inferred from the other components, but will be explicitly needed when we define partial MSCs where events may be unmatched.

Our main result will deal with systems that have (existentially) B-bounded channels, for some $B \geq 1$ [10]. Intuitively, an MSC is B-bounded if it can be scheduled in such a way that, along the execution, there are never more than B messages in each channel. Formally, we define boundedness via linearizations. A linearization of an MSC $M = (E, \lhd, \ell, \lambda)$ over topology $\mathcal{T} = (V, \rightarrow, \pi)$ is any structure $M' = (E, \preceq, \ell, \lambda)$ such that $\preceq$ is a total order on E satisfying $\lhd^* \subseteq \preceq$. Then, M' is called *B-bounded* if, for all $f \in E$, $u, v \in V$, and $a \in \mathcal{N}$ such that $u \xrightarrow{a} v$, we have $|\{e \in E \mid e \preceq f,\, \ell(e) = u, \text{ and } \lambda(e) = !a\}| - |\{e \in E \mid e \preceq f,\, \ell(e) = v, \text{ and } \lambda(e) = ?\overline{a}\}| \leq B$. In other words, in any prefix of M', there are no more than B pending messages, in every "channel" (u, v). Now, we say that MSC M is B-bounded if there is a linearization of M that is B-bounded. For example, for all $n \geq 2$, the MSC M^n_{lin} (cf. Figure 4) is 1-bounded, because its (only) linearization is 1-bounded.

3 Parameterized Communicating Automata

Next, we introduce parameterized communicating automata, whose definition does not depend on a topology, but only on $\mathcal{P}$ and $\mathcal{N}$. The language of an automaton, a set of MSCs, is then parameterized by a topology.

Definition 3. *A* parameterized communicating automaton (PCA) *over $\mathcal{P}$ and $\mathcal{N}$ is a tuple (S, Msg, Δ, I, F) where*

- *S is the finite set of* states*,*
- *Msg is the finite set of* messages*,*
- *$I : \mathcal{P} \rightarrow 2^S$ associates with each process type a set of* initial states*,*
- *F is the* acceptance condition*: a finite boolean combination of statements $\langle \#(s) \geq k \rangle$ with $s \in S$ and $k \in \mathbb{N}$ (to be read as "s occurs at least k times as the terminal state of a process"), and*
- *$\Delta \subseteq S \times \Sigma_{\mathcal{A}} \times S$ is the set of* transitions.

Here, $\Sigma_{\mathcal{A}} = \{\, !_m a \,, ?_m a \mid a \in \mathcal{N} \text{ and } m \in Msg\}$ contains send actions $!_m a$ and receive actions $?_m a$. A transition $(s, \eta, s') \in \Delta$ is also written $s \xRightarrow{\eta} s'$.

A PCA can be run on any topology $\mathcal{T} = (V, \rightarrow, \pi)$ over $\mathcal{P}$ and $\mathcal{N}$. When a process $u \in V$ executes $(s, !_m a, s') \in \Delta$, it changes its local state from s to s' and writes m into the FIFO channel (u, v), provided $u \xrightarrow{a} v$. The message m can then be read out by process v executing a transition with action $?_m\overline{a}$. Note that the messages are abstracted away in the observable MSC behavior (they are in the spirit of stack symbols in pushdown automata). Formally, we define the semantics of PCA directly on MSCs. This is equivalent to an operational semantics in terms of an infinite transition system, but closer to the logical approach where formulas are evaluated over MSCs (see Section 4).

Let $\mathcal{T} = (V, \rightarrow, \pi) \in \mathbb{T}$ be a topology and $M = (E, \lhd, \ell, \lambda)$ be an MSC over $\mathcal{T}$. Set $V_M \stackrel{\text{def}}{=} \{u \in V \mid E_u \neq \emptyset\}$. A *(global) initial state* of $\mathcal{A}$ for M is a tuple $\iota = (\iota_u)_{u \in V_M}$ where $\iota_u \in I(\pi(u))$ for all $u \in V_M$. Given ι and $\rho : E \rightarrow S$ (which

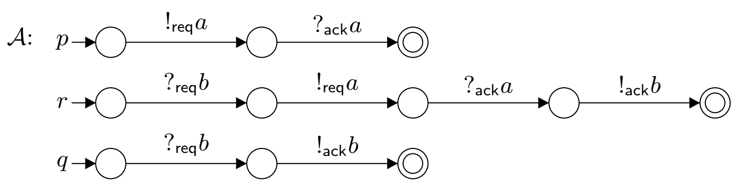

Fig. 5. PCA $\mathcal{A}$ over $\mathcal{P} = \{p, q, r\}$ and $\mathcal{N} = \{a, b\}$

is a candidate for being a run), we define another mapping $\rho_\iota^- : E \to S$, which returns the source state of a transition: For $(f, e) \in \lhd_{\mathsf{proc}}$, we let $\rho_\iota^-(e) = \rho(f)$; for a $\lhd_{\mathsf{proc}}$-minimal event $e \in E$, we let $\rho_\iota^-(e) = \iota_{\ell(e)}$.

A mapping $\rho : E \to S$ is called a *run* of $\mathcal{A}$ on M if there is an initial state $\iota = (\iota_u)_{u \in V_M}$ for M such that, for all $(e, f) \in \lhd_{\mathsf{msg}}$, there are $a \in \mathcal{N}$ and $m \in Msg$ satisfying $\ell(e) \xrightarrow{a} \ell(f)$, $\rho_\iota^-(e) \xRightarrow{!_m a} \rho(e)$, and $\rho_\iota^-(f) \xRightarrow{?_m \bar{a}} \rho(f)$. To determine if ρ is accepting, we define a multiset $h_\rho : S \to \mathbb{N}$ over S by $h_\rho(s) = |\{e \in E \mid e$ is $\lhd_{\mathsf{proc}}$-maximal and $\rho(e) = s\}|$. Now, ρ is *accepting* if h_ρ satisfies F in the expected manner; in particular, h_ρ satisfies $\langle \#(s) \geq k \rangle$ if $h_\rho(s) \geq k$. The MSC M is accepted by $\mathcal{A}$ if it admits an accepting run of $\mathcal{A}$. Note that h_ρ does not include any states of idle processes. So, a PCA cannot express "the topology has at least 5 processes", but only "at least 5 processes are active".

For a topology $\mathcal{T}$, the set of MSCs over $\mathcal{T}$ that are accepted by $\mathcal{A}$ is denoted by $L_{\mathcal{T}}(\mathcal{A})$. The restriction of $L_{\mathcal{T}}(\mathcal{A})$ to B-bounded MSCs is denoted by $L_{\mathcal{T}}^B(\varphi)$.

Example 3. Consider the PCA $\mathcal{A}$ from Figure 5. There, the acceptance condition F is simply the conjunction of formulas $\neg\langle \#(s) \geq 1 \rangle$ with s ranging over the states *without* double circle. Note that the messages req and ack do not occur in the accepted MSCs. In this example, we could actually do with just one message ($|Msg| = 1$). In general, however, message contents increase the expressive power of PCA. The MSC from Figure 4 is the only MSC accepted by $\mathcal{A}$ wrt. the topology $\mathcal{T}_{\mathrm{lin}}^8$. We actually have $L_{\mathcal{T}_{\mathrm{lin}}^n}(\mathcal{A}) = L^1_{\mathcal{T}_{\mathrm{lin}}^n}(\mathcal{A}) = \{M_{\mathrm{lin}}^n\}$ for all $n \geq 2$.

4 MSO Logic and Locality of FO logic

While PCA serve as a model of an implementation of a communicating system, we use monadic second-order (MSO) logic to *specify* properties of MSCs.

Logic. The MSO formulas over $\mathcal{P}$ and $\mathcal{N}$ are given by the following grammar:

$$\begin{aligned}\varphi \;::=\; & p(x) \mid\; !a(x) \mid\; ?a(x) \mid x \lhd_{\mathsf{proc}} y \mid x \lhd_{\mathsf{proc}}^* y \mid x \lhd_{\mathsf{msg}} y \mid x \lhd^* y \mid \\ & x = y \mid x \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x \varphi \mid \exists X \varphi\end{aligned}$$

where $p \in \mathcal{P}$, $a \in \mathcal{N}$, x and y are first-order variables (interpreted as events of an MSC), and X is a second-order variable (interpreted as a set of events), all taken from infinite supplies of variables. We use standard abbreviations such as $\varphi \wedge \psi$ for $\neg(\neg\varphi \vee \neg\psi)$, and $\varphi \to \psi$ for $\neg\varphi \vee \psi$.

Formulas are evaluated over MSCs $M = (E, \lhd, \ell, \lambda)$, over some topology $\mathcal{T} = (V, \rightarrow, \pi) \in \mathbb{T}$. Free variables x and X are interpreted by a mapping $\mathfrak{I}$ as an event $\mathfrak{I}(x) \in E$ and a set of events $\mathfrak{I}(X) \subseteq E$, respectively. The atomic formula $p(x)$ is true if $\pi(\ell(\mathfrak{I}(x))) = p$, i.e., the event associated with variable x is located on a process of type $p \in \mathcal{P}$. For η of the form $!a$ or $?a$, formula $\eta(x)$ is true if $\lambda(\mathfrak{I}(x)) = \eta$. Moreover, $x \lhd_{\mathsf{proc}}^* y$ is satisfied if $\mathfrak{I}(x) \lhd_{\mathsf{proc}}^* \mathfrak{I}(y)$. This allows us to say that two events are executed by the same process. Other formulas are interpreted as expected. Though $\lhd_{\mathsf{proc}}^*$ and also $\lhd^*$ can be defined in MSO in terms of $\lhd_{\mathsf{proc}}$ and $\lhd_{\mathsf{msg}}$, we include them explicitly in the logic, as they will be used in fragments in which they would no longer be expressible.

The set FO of *first-order* formulas is the fragment of MSO without second-order quantification. Moreover, EMSO (*existential* MSO) is the set of formulas of the form $\exists X_1 \ldots \exists X_n \varphi$ such that $\varphi \in$ FO. For a nonempty set $\sigma \subseteq \{\lhd_{\mathsf{proc}}, \lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}, \lhd^*\}$ of relation symbols, the logics FO$[\sigma]$ and EMSO$[\sigma]$ restrict FO and EMSO, respectively: instead of $\{\lhd_{\mathsf{proc}}, \lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}, \lhd^*\}$, we can only access the relation symbols from σ. In this paper, we focus on the logics EMSO$[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, EMSO$[\lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}]$, and FO$[\lhd_{\mathsf{proc}}^*, \lhd^*]$. Note that the predicate $\lhd_{\mathsf{proc}}$ can be expressed in FO$[\lhd_{\mathsf{proc}}^*]$ in terms of $\lhd_{\mathsf{proc}}^*$.

Let $\mathcal{T} \in \mathbb{T}$ be a topology, and let $\varphi \in$ MSO be a sentence, i.e., a formula without free variables. The set of MSCs over $\mathcal{T}$ that satisfy φ is denoted by $L_{\mathcal{T}}(\varphi)$. When φ is not a sentence, then $L_{\mathcal{T}}(\varphi)$ contains the *pairs*, of an MSC and an interpretation of the free variables, that satisfy φ. The restriction of $L_{\mathcal{T}}(\varphi)$ to B-bounded MSCs is denoted by $L_{\mathcal{T}}^B(\varphi)$.

Example 4. Suppose $\mathcal{N}$ contains the names a and b with $\overline{a} = b$. Consider the sentence $\varphi = \forall x \forall y\, (!a(x) \wedge x \lhd_{\mathsf{msg}} y \;\rightarrow\; \exists y'(y \lhd_{\mathsf{proc}}^* y' \wedge !b(y'))) \in$ FO$[\lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}]$. It says that, whenever a process sends a message (say, a request) through interface a, then the receiving process sends (after reception) an acknowledgment to the requesting process. We have $M_{\mathrm{lin}}^n \in L_{\mathcal{T}_{\mathrm{lin}}^n}(\varphi)$ for all $n \geq 2$.

Locality of First-Order Logic. Next, we state a locality theorem due to Schwentick and Barthelmann [15].[2] It formalizes the intuition that FO can only reason about local *spheres*, which include elements whose distance from a given center is bounded by a parameter that depends on the formula.

Fix a nonempty set $\sigma \subseteq \{\lhd_{\mathsf{proc}}, \lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}, \lhd^*\}$ of relation symbols. Let $M = (E, \lhd, \ell, \lambda)$ be an MSC over some topology $\mathcal{T} = (V, \rightarrow, \pi)$. The distance $dist_M^\sigma(e, f)$ between events $e, f \in E$ is the minimal length of a path between e and f in the graph of M with edges given by σ, in either direction (or ∞ if such a path does not exist). For example, if $\sigma = \{\lhd_{\mathsf{proc}}^*, \lhd_{\mathsf{msg}}\}$, then $dist_M^\sigma(e, f)$ refers to the distance in the (undirected) graph $(E, \lhd_{\mathsf{proc}}^* \cup (\lhd_{\mathsf{proc}}^*)^{-1} \cup \lhd_{\mathsf{msg}} \cup \lhd_{\mathsf{msg}}^{-1})$. Moreover, $dist_M^\sigma(e, e) = 0$ and $dist_M^\sigma(e, f) = dist_M^\sigma(f, e) = 1$ for all $(e, f) \in \lhd_{\mathsf{msg}}$.

Let $R \geq 1$. A formula $\chi \in$ FO$[\sigma]$ is called (R, σ)-*local* around a first-order variable y if (i) y is not quantified in χ and (ii) χ is obtained from some FO$[\sigma]$-formula by replacing each subformula of the form $\exists z \psi$ with $\exists z(dist^\sigma(y, z) < R \wedge$

[2] Gaifman's normal form appears to be more difficult to deal with in our context.

ψ), and each subformula of the form $\forall z \psi$ with $\forall z(dist^{\sigma}(y,z) < R \rightarrow \psi)$. Here, $dist^{\sigma}(y,z) < R$ denotes the obvious FO[σ]-formula. We use strict inequality for technical reasons (cf. [8]). Adapted to our setting, [15] yields the following:

Fact 1 (Schwentick & Barthelmann, [15]). *Let $\varphi \in \mathrm{FO}[\sigma]$. There are $R \geq 1$ and $\varphi' = \exists x_1 \ldots \exists x_n \forall y \chi \in \mathrm{FO}[\sigma]$ such that χ is (R,σ)-local around y and, for all topologies $\mathcal{T} \in \mathbb{T}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\varphi')$.*

5 EMSO vs. PCA over Prime Topologies

Our aim is to synthesize, from a logical formula, a PCA that is correct, wrt. the formula, for all topologies from a given class. In this section, we consider the logic $\mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$. We show that, for every given sentence from that logic, there is a PCA that is equivalent over all *prime* topologies.

For a topology $(V, \rightarrow, \pi) \in \mathbb{T}$, nodes $u, v \in V$, and $w = a_1 \ldots a_n \in \mathcal{N}^*$, we write $u \stackrel{w}{\rightarrow} v$ if there is a w-labeled path from u to v, i.e., there are $u_0, \ldots, u_n \in V$ such that $u = u_0 \stackrel{a_1}{\rightarrow} u_1 \stackrel{a_2}{\rightarrow} \ldots \stackrel{a_n}{\rightarrow} u_n = v$.

Definition 4. *A topology $(V, \rightarrow, \pi) \in \mathbb{T}$ is called* prime *if, for all $u \in V$, $w \in \mathcal{N}^*$, and $n \geq 1$, $u \stackrel{w^n}{\rightarrow} u$ implies $u \stackrel{w}{\rightarrow} u$.*

In other words, a prime topology satisfies the following monotonicity property: If $u \stackrel{w}{\rightarrow} v$ with $u \neq v$, then starting from u and "applying" w several times will never lead back to u. For example, all topologies in $\mathfrak{T}_{\mathrm{lin}}$, $\mathfrak{T}_{\mathrm{tree}}$, and $\mathfrak{T}_{\mathrm{grid}}$ are prime, while none of the topologies in $\mathfrak{T}_{\mathrm{ring}}$ is prime.

Theorem 1. *Let $\varphi \in \mathrm{EMSO}[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ be a sentence. There is a PCA $\mathcal{A}$ such that, for all prime topologies $\mathcal{T} \in \mathbb{T}$, we have $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.*

Proof (sketch). Set $\sigma_0 = \{\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}\}$. Given a sentence $\varphi \in \mathrm{EMSO}[\sigma_0]$, there are, according to Fact 1, a radius $R \geq 1$ and $\varphi' = \exists X_1 \ldots \exists X_m \exists x_1 \ldots \exists x_n \forall y \chi \in \mathrm{EMSO}[\sigma_0]$ such that $\chi \in \mathrm{FO}[\sigma_0]$ is (R, σ_0)-local around y and, for all $\mathcal{T} \in \mathbb{T}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\varphi')$. The free variables of χ can be considered as unary predicates and are dealt with by projection from an extended alphabet. By means of the acceptance condition of a PCA, one can make sure that variables x_i are indeed interpreted as exactly one event. So, it essentially remains to translate the formula $\forall y \chi$ into a PCA. For this, we use the *sphere automaton*, a PCA that "detects" neighborhoods of radius R in an input MSC (including possible interpretations of free variables). More precisely, it accepts any MSC, over any given *prime* topology. Moreover, in any accepting run, the state assigned to event e tells us whether χ holds, or not, when y is interpreted as e. A sphere automaton is presented in [2] for a fixed, known topology, but it is actually independent of that topology. In the proof, it is only needed that MSCs are prime, essentially in the same sense as for topologies. But MSCs over prime topologies are indeed prime (cf. Appendix A) so that we obtain the desired sphere automaton. As a last step, the latter is restricted to states that signal that χ holds. There is, however, another subtlety. Satisfaction of χ in an MSC depends on the neighborhood of y of radius R but also on the truth values of propositions involving only the free variables of $\forall y \chi$. Following [8], the PCA will guess and verify these truth values. By means of the acceptance condition, we can make sure that the guess is consistent throughout a run. Note that we could have used Hanf's normal form for the proof, but did not do so to avoid additional notation. ☐

On the other hand, the translation of PCA to EMSO$[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ is not restricted to topologies of a particular form. The proof of the following theorem is by a standard construction and omitted.

Theorem 2. *Let $\mathcal{A}$ be a PCA. There is a sentence $\varphi \in$ EMSO$[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ such that, for all topologies $\mathcal{T} \in \mathbb{T}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\mathcal{A})$.*

6 EMSO vs. PCA over Unambiguous Topology Classes

In EMSO$[\lhd_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, one cannot express that two events are executed by the same process, while this is possible in EMSO$[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ using $\lhd^*_{\mathsf{proc}}$. To capture EMSO$[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ by PCA, we impose a channel bound and restrict to certain *unambiguous* classes of topologies. In turn, this will allow us to apply our results to ring topologies. For $w = a_1 \dots a_n \in \mathcal{N}^*$, let $|w|$ denote the length n of w.

Definition 5. *Let $k \in \mathbb{N}$. A set $\mathfrak{T} \subseteq \mathbb{T}$ of topologies is* k-unambiguous *if, for all $w \in \mathcal{N}^*$ with $|w| \leq k$, all topologies $(V, \rightarrow, \pi), (V', \rightarrow', \pi') \in \mathfrak{T}$, and all nodes $u, v \in V$ and $u', v' \in V'$ such that $u \xrightarrow{w} v$ and $u' \xrightarrow{w}{}' v'$, we have $u = v$ iff $u' = v'$.*

In other words, if a topology from $\mathfrak{T}$ admits a directed cycle of length $\leq k$ with label w, then following w (if possible) will always form a cycle, in any topology from $\mathfrak{T}$. For example, the sets $\mathfrak{T}_{\mathrm{lin}}$, $\mathfrak{T}_{\mathrm{tree}}$, $\mathfrak{T}_{\mathrm{grid}}$, and $\{\mathcal{T}^n_{\mathrm{ring}} \mid n \geq \max\{3, k+1\}\}$ are all k-unambiguous, for every $k \in \mathbb{N}$.

For $\varphi \in$ EMSO$[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$, let $R_\varphi \geq 1$ denote the radius associated with the first-order kernel of φ according to Fact 1. Our main result is the following:

Theorem 3. *Let $\varphi \in$ EMSO$[\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}]$ be a sentence, $B \geq 1$, and $\mathfrak{T} \subseteq \mathbb{T}$ be a $(2R_\varphi + 1)$-unambiguous set of topologies. There is a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\varphi)$.*

In particular, the constructed automaton is correct, wrt. φ, for the classes of pipeline, tree, and grid topologies, as well as for *almost all* ring topologies. The rest of this section is devoted to the proof of Theorem 3.

Proof. Set $\sigma^*_0 = \{\lhd^*_{\mathsf{proc}}, \lhd_{\mathsf{msg}}\}$. Thanks to Fact 1, the problem reduces to constructing a PCA from a formula $\forall y \chi \in$ FO$[\sigma^*_0]$ where χ is local around y (cf. also proof of Theorem 1). We restrict to the case where y is the only free variable of χ. Further free variables are dealt with like in the proof of Theorem 1, see also [8, page 806] and Remark 1 in the forthcoming construction. So, for the rest of the proof, we fix $R, B \geq 1$, a $(2R+1)$-unambiguous set $\mathfrak{T} \subseteq \mathbb{T}$ of topologies, and a sentence $\forall y \chi(y) \in$ FO$[\sigma^*_0]$ such that $\chi(y)$ is (R, σ^*_0)-local around y. We build a PCA $\mathcal{A}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\forall y \chi(y))$.

The Idea. We exploit locality of $\chi(y)$. To know whether $M, e \models \chi(y)$ (i.e., M satisfies $\chi(y)$ when y is interpreted as e), it is sufficient to look at a bounded neighborhood of e. This is illustrated in Figure 6. Consider any event on the vertex u that is marked by a rectangle. All events with distance at most $R = 1$ from e (wrt. σ_0^*) are necessarily contained in the area highlighted in gray. This area contains all the information that we need to infer satisfaction of $\forall y \chi(y)$ provided that the outermost universal quantifier is restricted to events on u. Thus, we deal with an area of fixed "width", i.e., over a communication topology of bounded size, with u as a distinguished *center*. For such fixed *sphere topologies*, there exists a translation from MSO logic, involving $\lhd_{\mathsf{proc}}^*$ and $\lhd_{\mathsf{msg}}$, to non-parameterized communicating automata (CA) [10]. For [10] to apply, we have to restrict to B-bounded MSCs. Given a sphere topology θ, we can then construct a CA $\mathcal{B}_\theta$ that recognizes MSCs over θ (i.e., of bounded width) satisfying $\chi(y)$ for all events e located on the center u. Up to isomorphism, there are only finitely many sphere topologies θ of radius R and, thus, finitely many CA $\mathcal{B}_\theta$. To obtain a PCA running on MSCs of unbounded width, we will *glue* these CA together.

Every process u will run an automaton $\mathcal{B}_\theta$ and make sure that the bounded MSC in its neighborhood is accepted by $\mathcal{B}_\theta$. More precisely, it guesses a sphere θ, supposing that its topology neighborhood looks like θ, and runs a copy of $\mathcal{B}_\theta$. Whenever u communicates with neighboring processes, this guess is forwarded in terms of messages. Processes receiving the guess also have to simulate $\mathcal{B}_\theta$. Since neighboring processes also have to verify their own guess, a process will have to run several CA simultaneously. The main difficulty, however, is to detect the sphere topology around one given process in an MSC, i.e., to verify that a guess is correct. It will actually be sufficient to detect only the subsphere whose edges are really involved in the execution of the MSC, since CA $\mathcal{B}_\theta$ and $\mathcal{B}_{\theta'}$ cannot distinguish between (bounded-width) MSCs M whenever θ is a substructure of θ' and M uses only channels present in θ. Note that this procedure of guessing and forwarding sphere topologies is not able to detect cycles in the sphere topology by itself. It is only correct thanks to the fact that the underlying set $\mathfrak{T}$ of topologies is $(2R+1)$-unambiguous. Here, $2R+1$ is the maximal length of a cycle through a sphere center that is needed to cover a given edge in the sphere. In the following, we will formalize these ideas.

Communicating Automata over Sphere Topologies. Let $\mathcal{T} = (V, \rightarrow, \pi) \in \mathbb{T}$. For $u, v \in V$, let $dist_{\mathcal{T}}(u, v)$ denote the distance (i.e., the minimal length of a path, or ∞ if such a path does not exist) between u and v in the edge-labeled graph $(V, \rightarrow)$. By $R\text{-Sph}(\mathcal{T}, u)$, we denote the *$R$-sphere* of $\mathcal{T}$ around u, i.e., the substructure of $\mathcal{T}$ induced by the vertices $v \in V$ such that $dist_{\mathcal{T}}(u, v) \leq R$, with u as an additional constant called *center*. Note that $R\text{-Sph}(\mathcal{T}, u)$ is always a topology (when we ignore the constant). The top of Figure 7 depicts $1\text{-Sph}(\mathcal{T}_{\mathrm{lin}}^8, u)$ where u is given as in Figure 6.

We let $R\text{-}Spheres(\mathbb{T}) \overset{\text{def}}{=} \{R\text{-Sph}(\mathcal{T}, u) \mid \mathcal{T} = (V, \rightarrow, \pi) \in \mathbb{T} \text{ and } u \in V\}$. Any element from $R\text{-}Spheres(\mathbb{T})$ is called an (R-)*sphere*. We (mostly) do not distinguish isomorphic spheres so that the number of R-spheres is finite.

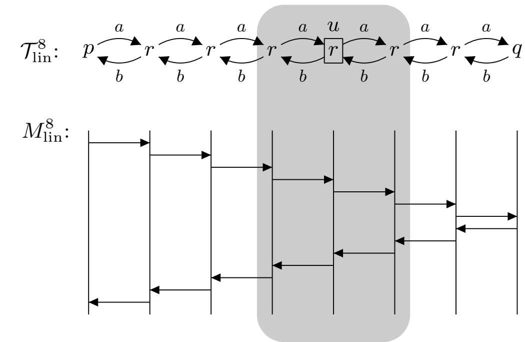

Fig. 6. Local view of a process in an MSC

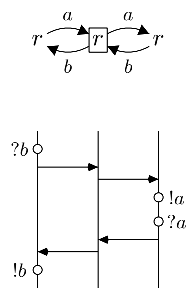

Fig. 7. An MSC over a 1-sphere

We will now define MSCs over an R-sphere $\theta = (\mathcal{U}, \gamma)$ with $\mathcal{U} = (U, \rightsquigarrow, \xi)$. These MSCs are *connected*, are empty or have at least one event on γ, and may have unmatched events (those whose communication partners are beyond $\mathcal{U}$). Formally, an MSC over θ is a tuple $M = (E, \lhd, \ell, \lambda)$ where the components are like in an MSC over an ordinary topology (Definition 2), with the only difference that E may be empty. In particular, $\ell : E \to U$ maps an event to a process. We adopt the definition of $E_!$, $E_?$, and E_u. Additionally, we may have *unmatched* events. So, we let $E_{\text{unm}} \stackrel{\text{def}}{=} E \setminus (E_! \cup E_?)$. We require that $E = E_! \uplus E_? \uplus E_{\text{unm}}$ and that 2.–4. from Definition 2 hold.[3] Moreover, if $E \neq \emptyset$, then

- the graph $(E, \lhd)$ is connected,
- there is $e \in E$ such that $\ell(e) = \gamma$, and
- for all $e \in E_{\text{unm}}$ and $a \in \mathcal{N}$ such that $\lambda(e) \in \{!a, ?a\}$, we have both $\mathit{dist}_{\mathcal{U}}(\gamma, \ell(e)) = R$ and there is no $u \in U$ such that $\ell(e) \stackrel{a}{\rightsquigarrow} u$.

Note that this definition actually depends on R, which we had fixed.

The definition of B-bounded MSCs over θ is literally the same as for MSCs over topologies. This means that unmatched events are discarded and considered as internal actions, i.e., they do not count when computing the difference between sends and receives. An example of a 1-bounded MSC over a 1-sphere is depicted in Figure 7. It corresponds to the gray-shaded area in Figure 6.

Definition 6. *Let $\theta = (\mathcal{U}, \gamma)$ be an R-sphere with $\mathcal{U} = (U, \rightsquigarrow, \xi)$. A* communicating automaton (CA) *over θ is a tuple $\mathcal{B} = (S, \Delta, \iota, F)$ where*

- *S is the finite set of* states*,*
- *$\iota : U \to S$ associates with each process an initial state,*
- *$F : U \to 2^S$ associates with each process a set of* local final states*, and*
- *$\Delta \subseteq S \times \Sigma_{\mathcal{B}} \times S$ is the set of transitions.*

Here, the set of actions is $\Sigma_{\mathcal{B}} = \{\, !_m a\, , ?_m a \mid a \in \mathcal{N} \text{ and } m \in S\}$. We require that, for all $(s, !_m a, s') \in \Delta$, we have $m = s'$. This is sufficient and will simplify our constructions.

[3] In Definition 2, replace V with U, and $\to$ with $\rightsquigarrow$.

Runs of CA are defined similarly to PCA, but we have to consider unmatched events. Let $M = (E, \lhd, \ell, \lambda)$ be an MSC over θ, and let $\rho : E \to S$ be a mapping. We define $\rho^- : E \to S$ as follows: For $(f, e) \in \lhd_{\mathsf{proc}}$, let $\rho^-(e) = \rho(f)$; for a $\lhd_{\mathsf{proc}}$-minimal event $e \in E$, we let $\rho^-(e) = \iota(\ell(e))$. Then, ρ is a *run* of $\mathcal{B}$ on M if

- for all $(e, f) \in \lhd_{\mathsf{msg}}$ and names $a \in \mathcal{N}$ with $\ell(e) \overset{a}{\rightsquigarrow} \ell(f)$, we have both $(\rho^-(e), !_{\rho(e)}a, \rho(e)) \in \Delta$ and $(\rho^-(f), ?_{\rho(e)}\overline{a}, \rho(f)) \in \Delta$,
- for all $e \in E_{\mathrm{unm}}$ and $a \in \mathcal{N}$ with $\lambda(e) = !a$, we have $(\rho^-(e), !_{\rho(e)}a, \rho(e)) \in \Delta$,
- for all $e \in E_{\mathrm{unm}}$ and $a \in \mathcal{N}$ with $\lambda(e) = ?a$, we have $(\rho^-(e), ?_m a, \rho(e)) \in \Delta$ for some $m \in S$ (the message is irrelevant).

The run is *accepting* if $\rho(e) \in F(\ell(e))$ for all $\lhd_{\mathsf{proc}}$-maximal events $e \in E$. By $L(\mathcal{B})$, we denote the set of MSCs over θ for which there is an accepting run of $\mathcal{B}$. Note that $L(\mathcal{B})$ always contains the empty MSC over θ.

Let $\mathcal{T} = (V, \to, \pi) \in \mathbb{T}$ be a topology, $u \in V$, and $M = (E, \lhd, \ell, \lambda)$ be an MSC over $\mathcal{T}$. Set $H = \{(u, a, v) \in \to \mid u = \ell(e) \text{ and } v = \ell(f) \text{ for some } (e, f) \in \lhd_{\mathsf{msg}} \cup \lhd_{\mathsf{msg}}^{-1}\}$. Let U be the set of nodes $v \in V$ such that u and v are connected in the graph (V, H) by a path using at most R edges. By $R\text{-Sph}(M, u)$ (somewhat abusing notation), we denote the restriction of M to events in $\ell^{-1}(U)$. We also define $R\text{-Sph}(\mathcal{T}, u) \restriction M \stackrel{\text{def}}{=} ((U, H_{|U}, \pi_{|U}), u)$ where $H_{|U} = \{(u, a, v) \in H \mid u, v \in U\}$ and $\pi_{|U}$ is the restriction of π to U. Then, $R\text{-Sph}(\mathcal{T}, u) \restriction M$ is an R-sphere and $R\text{-Sph}(M, u)$ is an MSC over $R\text{-Sph}(\mathcal{T}, u) \restriction M$. For example, consider the MSC M^8_{lin} and topology $\mathcal{T}^8_{\text{lin}}$ from Figure 6. Figure 7 shows $1\text{-Sph}(M^8_{\text{lin}}, u)$ and $1\text{-Sph}(\mathcal{T}^8_{\text{lin}}, u) \restriction M^8_{\text{lin}} = 1\text{-Sph}(\mathcal{T}^8_{\text{lin}}, u)$.

The following theorem follows from a result by Genest, Kuske, and Muscholl:

Theorem 4 (cf. [10], Theorem 4.1). *There is a collection $(\mathcal{B}_\theta)_{\theta \in R\text{-}Spheres(\mathbb{T})}$ of CA $\mathcal{B}_\theta$ over θ such that the following holds, for all topologies $\mathcal{T} = (V, \to, \pi) \in \mathbb{T}$, all $u \in V$, and all B-bounded MSCs $M = (E, \lhd, \ell, \lambda)$ over $\mathcal{T}$:*

$$\begin{aligned} & M, e \models \chi(y) \textit{ for all } e \in E_u \\ \iff\ & R\text{-Sph}(M, u) \in L(\mathcal{B}_{R\text{-Sph}(\mathcal{T}, u) \restriction M}) \,. \end{aligned}$$

The result from [10] applies, since $R\text{-Sph}(M, u)$ is B-bounded and, given $e \in E_u$, all events $f \in E$ with $\mathit{dist}_M^{\sigma_0^*}(e, f) \leq R$ are covered by $R\text{-Sph}(M, u)$. Moreover, as MSCs over spheres are connected, local final states are enough.

Remark 1. When χ has more free variables than just y, $\mathcal{B}_\theta$ also depends on an assignment of truth values to propositions in χ over these variables.

The Construction. According to Theorem 4, it will be sufficient that each process of the PCA that we are going to construct identifies a subsphere of its actual topology neighborhood. So, we set $R\text{-}\mathit{Sub}(\mathfrak{T}) \stackrel{\text{def}}{=} \{R\text{-Sph}(\mathcal{T}, u) \restriction M \mid \mathcal{T} = (V, \to, \pi) \in \mathfrak{T},\ u \in V, \text{ and } M \text{ an MSC over } \mathcal{T}\}$. Note that, for $\mathcal{T} \in \mathfrak{T}$ and a process u of $\mathcal{T}$, $R\text{-}\mathit{Sub}(\mathfrak{T})$ includes $R\text{-Sph}(\mathcal{T}, u)$ as well as some spheres that consist only of one single node. We fix the finitely many CA $(\mathcal{B}_\theta)_{\theta \in R\text{-}\mathit{Sub}(\mathfrak{T})}$ according to Theorem 4, where $\mathcal{B}_\theta = (S_\theta, \Delta_\theta, \iota_\theta, F_\theta)$ such that the sets S_θ are pairwise disjoint. We say that $w \in \mathcal{N}^*$ is *circular* if there are $\mathcal{T} = (V, \rightarrow, \pi) \in \mathfrak{T}$ and $u \in V$ such that $u \xrightarrow{w} u$. The PCA $\mathcal{A} = (S, Msg, \Delta, I, F)$ with $S = Msg$ is defined as follows ($\mathcal{U}$ will always refer to $(U, \rightsquigarrow, \xi)$ and $\mathcal{U}'$ to $(U', \rightsquigarrow', \xi')$):

States. A state $t \in S$ is a nonempty set of tuples $\kappa = (\mathcal{U}, \gamma, \alpha, s, H)$ where $(\mathcal{U}, \gamma) \in R\text{-}Sub(\mathfrak{T})$ is a guessed sphere, $\alpha \in U$ is the *active process*, $s \in S_{(\mathcal{U},\gamma)}$ is the current state of the CA $\mathcal{B}_{(\mathcal{U},\gamma)}$ that is simulated, and $H \subseteq \mathcal{N}$ is the "history" containing the names that have been used by the active process. Intuitively, a process whose current state contains κ simulates process α in $\mathcal{B}_{(\mathcal{U},\gamma)}$, supposing that its topology neighborhood resembles $(\mathcal{U}, \gamma, \alpha)$. So, we require that, for all $\kappa = (\mathcal{U}, \gamma, \alpha, s, H) \in t$ and $\kappa' = (\mathcal{U}', \gamma', \alpha', s', H') \in t$, the following hold:

(a) $\xi(\alpha) = \xi'(\alpha')$,
(b) if $\gamma = \alpha$ and $\gamma' = \alpha'$, then $\kappa = \kappa'$, and
(c) if $(\mathcal{U}, \gamma, \alpha) = (\mathcal{U}', \gamma', \alpha')$, then $\kappa = \kappa'$.

Let $t \in S$ and $a \in \mathcal{N}$. Let $t_{\xrightarrow{a}}$ denote the set of tuples $(\mathcal{U}, \gamma, \alpha, s, H) \in t$ such that $\alpha \overset{a}{\rightsquigarrow} u$ for some $u \in U$. We say that a is *enabled* in t if, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in t$, $u \in U$, and $w \in \mathcal{N}^{\leq 2R}$ such that wa is circular and $u \overset{w}{\rightsquigarrow} \alpha$, we have $\alpha \overset{a}{\rightsquigarrow} u$.

Initial and Final States. For $p \in \mathcal{P}$, a state $t \in S$ is contained in $I(p)$ if, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in t$, we have $\xi(\alpha) = p$, $s = \iota_{(\mathcal{U},\gamma)}(\alpha)$, and $H = \emptyset$. Towards the final states, let G be the set of states $t \in S$ such that, for all tuples $(\mathcal{U}, \gamma, \alpha, s, H) \in t$, we have $s \in F_{(\mathcal{U},\gamma)}(\alpha)$ and $\{a \in \mathcal{N} \mid \alpha \overset{a}{\rightsquigarrow} u \text{ for some } u \in U\} \subseteq H$. The latter means that H contains all verification obligations imposed by the guessed topology $\mathcal{U}$. Then, F is defined as $\bigwedge_{t \in S \setminus G} \neg\langle \#(t) \geq 1 \rangle$.

Send Transitions (ST). The triple $(t^-, !_m a, t) \in S \times \Sigma_{\mathcal{A}} \times S$ is contained in Δ if $m = t$, a is enabled in t, and there is a bijection $\Phi : t^- \rightarrow t$ such that $\Phi(\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-) = (\mathcal{U}, \gamma, \alpha, s, H)$ implies

1. $(\mathcal{U}^-, \gamma^-, \alpha^-) = (\mathcal{U}, \gamma, \alpha)$, i.e., the executing process maintains its guesses,
2. $(s^-, !_s a, s) \in \Delta_{(\mathcal{U},\gamma)}$, which simulates a step of process α in the CA $\mathcal{B}_{(\mathcal{U},\gamma)}$,
3. α has an a-successor in $\mathcal{U}$ or $dist_{\mathcal{U}}(\gamma, \alpha) = R$, and
4. $H = H^- \cup \{a\}$, which marks interface a as "checked".

Receive Transitions (RT). The triple $(t^-, ?_m a, t) \in S \times \Sigma_{\mathcal{A}} \times S$ is contained in Δ if a is enabled in t and there are bijections $\Phi : t^- \rightarrow t$ and $\hat{\Phi} : m_{\xrightarrow{\bar{a}}} \rightarrow t_{\xrightarrow{a}}$ as well as a mapping $\mu : t_{\xrightarrow{a}} \rightarrow \bigcup_{\theta \in R\text{-}Sub(\mathfrak{T})} S_\theta$ (associating with a tuple a message of a CA) such that the following hold:

(a) $\Phi(\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-) = (\mathcal{U}, \gamma, \alpha, s, H)$ implies (let $\kappa = (\mathcal{U}, \gamma, \alpha, s, H)$)
   1. $(\mathcal{U}^-, \gamma^-, \alpha^-) = (\mathcal{U}, \gamma, \alpha)$,
   2. $(s^-, ?_k a, s) \in \Delta_{(\mathcal{U},\gamma)}$ for some k such that, if $\kappa \in t_{\xrightarrow{a}}$, then $k = \mu(\kappa)$,
   3. α has an a-successor in $\mathcal{U}$ or $dist_{\mathcal{U}}(\gamma, \alpha) = R$, and
   4. $H = H^- \cup \{a\}$.

(b) $\hat{\Phi}(\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-) = (\mathcal{U}, \gamma, \alpha, s, H)$ implies
   1. $\mu(\mathcal{U}, \gamma, \alpha, s, H) = s^-$,
   2. $(\mathcal{U}^-, \gamma^-) = (\mathcal{U}, \gamma)$, i.e., the guessed sphere is forwarded, and
   3. $\alpha \overset{a}{\rightsquigarrow} \alpha^-$ (assuming $\mathcal{U} = (U, \rightsquigarrow, \xi)$).

(c) For all $(\mathcal{U}, \gamma, \alpha, s, H) \in (m \setminus m_{\xrightarrow{\bar{a}}}) \cup (t \setminus t_{\xrightarrow{a}})$, we have $dist_{\mathcal{U}}(\gamma, \alpha) = R$.

Note that the mappings required in (ST) and (RT) are unique, if they exist.

This concludes the construction of the PCA $\mathcal{A}$. We claim that, for all $\mathcal{T} \in \mathfrak{T}$, $L^B_{\mathcal{T}}(\mathcal{A}) = L^B_{\mathcal{T}}(\forall y \chi(y))$. The correctness proof can be found in Appendix B. It crucially relies on Theorem 4 and the fact that $\mathfrak{T}$ is $(2R+1)$-unambiguous. □

7 Beyond Implementability

In terms of logic, Theorem 3 cannot be improved. One can show that there is no translation of formulas into PCA when we add $\lhd^*$ to the logic (even when we restrict to 1-bounded MSCs and tree topologies). This has to be contrasted with the expressive equivalence of MSO and CA over *fixed* topologies when imposing any existential bound on the channels [10].

In the following theorem, we consider the tree topologies from $\mathfrak{T}_{\mathrm{tree}}$, over $\mathcal{P} = \{p, q, r\}$ and $\mathcal{N} = \{a, b, c, d\}$ (cf. Example 1).

Theorem 5. *There exists a sentence $\varphi \in \mathrm{FO}[\lhd^*_{\mathsf{proc}}, \lhd^*]$ such that, for all PCA $\mathcal{A}$ over $\mathcal{P}$ and $\mathcal{N}$, there is $\mathcal{T} \in \mathfrak{T}_{\mathrm{tree}}$ with $L^1_{\mathcal{T}}(\mathcal{A}) \neq L^1_{\mathcal{T}}(\varphi)$.*

The proof can be found in Appendix C. It uses a technique from [17], which was employed to show that FO (with reflexive transitive closure relations) and a local variant of EMSO are incomparable over *pictures*.

8 Conclusion

In this paper, we developed a framework for communicating systems with parameterized network topology. Our main contributions are (i) a notion of communicating automaton that is independent of a particular topology and (ii) characterizations of PCA in terms of existential fragments of MSO logic. For (ii), we have to restrict to prime topologies and, respectively, unambiguous topology classes. While this is optimal wrt. the logics considered, it will be worthwhile to examine if larger topology classes exist that generalize our results. Our framework may carry over to Zielonka's asynchronous automata [19] with binary actions. These automata have been considered in [9] over tree architectures to get decidability of the controller-synthesis problem. This also raises the question about a parameterized formulation of the control problem. Another interesting goal would be a framework including topologies of unbounded degree such as unranked trees. One may also consider "classical" parameterized verification: Given a PCA $\mathcal{A}$, is there a topology $\mathcal{T}$ such that $L_{\mathcal{T}}(\mathcal{A}) \neq \emptyset$? Since those questions are undecidable in general, one has to impose restrictions, on PCA and/or on the topologies.

References

1. P. A. Abdulla, A. Bouajjani, B. Jonsson, and M. Nilsson. Handling global conditions in parameterized system verification. In *Proc. of CAV'99*, volume 1633 of *LNCS*, pages 134–145. Springer, 1999.
2. B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006.
3. A. Bouajjani, P. Habermehl, and T. Vojnar. Verification of parametric concurrent systems with prioritised fifo resource management. *Formal Methods in System Design*, 32(2):129–172, 2008.
4. M. C. Browne, E. M. Clarke, and O. Grumberg. Reasoning about networks with many identical finite state processes. *Inf. Comput.*, 81(1):13–31, 1989.
5. J. Büchi. Weak second order logic and finite automata. *Z. Math. Logik, Grundlag. Math.*, 5:66–62, 1960.
6. G. Delzanno, A. Sangnier, and G. Zavattaro. Parameterized verification of ad hoc networks. In *Proc. of CONCUR'10*, volume 6269 of *LNCS*. Springer, 2010.
7. C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–52, 1961.
8. P. Gastin and D. Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. *Inf. Comput.*, 208(7):797–816, 2010.
9. B. Genest, H. Gimbert, A. Muscholl, and I. Walukiewicz. Asynchronous games over tree architectures. In *Proc. of ICALP'13*, volume 7966 of *LNCS*, pages 275–286. Springer, 2013.
10. B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Inf. Comput.*, 204(6):920–956, 2006.
11. S. M. German and A. P. Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992.
12. S. Grumbach and Z. Wu. Logical locality entails frugal distributed computation over graphs (extended abstract). In *Proc. of WG'09*, volume 5911 of *LNCS*, pages 154–165, 2010.
13. J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Inf. Comput.*, 202(1):1–38, 2005.
14. D. Kuske. Regular sets of infinite message sequence charts. *Inf. Comput.*, 187:80–109, 2003.
15. T. Schwentick and K. Barthelmann. Local normal forms for first-order logic with applications to games and automata. *Discrete Mathematics & Theoretical Computer Science*, 3(3):109–124, 1999.
16. J. W. Thatcher and J. B. Wright. Generalized finite automata theory with application to a decision problem of second-order logic. *Math. Systems Theory*, 2(1):57–81, 1968.
17. W. Thomas. Elements of an automata theory over partial orders. In *Proc. of POMIV 1996*, volume 29 of *DIMACS*. AMS, 1996.
18. B. A. Trakhtenbrot. Finite automata and monadic second order logic. *Siberian Math. J*, 3:103–131, 1962. In Russian; English translation in *Amer. Math. Soc. Transl.* **59**, 1966, 23–55.
19. W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.

A Missing Details for Proof of Theorem 1

MSCs over Prime Topologies. Recall that the construction of a PCA from a formula is based on the sphere automaton. For the construction from [2], which we recall below, to be applicable, we have to show that MSCs are prime just like topologies. Let us define when an MSC is prime.[4] Consider the set $\mathbb{D} = \{\mathsf{proc}, \mathsf{proc}^{-1}\} \cup \{\mathsf{msg}_a, \mathsf{msg}_a^{-1} \mid a \in \mathcal{N}\}$ of *directions*. Let $M = (E, \lhd, \ell, \lambda)$ be an MSC, over some topology. Every direction $\delta \in \mathbb{D}$ defines a binary relation $[\![\delta]\!]_M \subseteq E \times E$ as follows: $[\![\mathsf{proc}]\!]_M = \lhd_{\mathsf{proc}}$, $[\![\mathsf{proc}^{-1}]\!]_M = \lhd_{\mathsf{proc}}^{-1}$, $[\![\mathsf{msg}_a]\!]_M = \{(e,f) \in \lhd_{\mathsf{msg}} \mid \lambda(e) = !a\}$, and $[\![\mathsf{msg}_a^{-1}]\!]_M = [\![\mathsf{msg}_a]\!]_M^{-1}$. This is extended to strings $w = \delta_1 \ldots \delta_n \in \mathbb{D}^*$: we let $(e,f) \in [\![w]\!]_M$ if $(e,f) \in \mathrm{id}_E \circ [\![\delta_1]\!]_M \circ \ldots \circ [\![\delta_n]\!]_M$ (where $\circ$ is the relation product). The MSC M is called *prime* if, for all $w \in \mathbb{D}^*$, $n \geq 1$, and $e \in E$, we have that $(e,e) \in [\![w^n]\!]_M$ implies $(e,e) \in [\![w]\!]_M$. Note that MSCs are in general not prime, i.e., for arbitrary topologies. However, we can show the following:

Lemma 1. *Let $\mathcal{T} = (V, \rightarrow, \pi)$ be a prime topology and $M = (E, \lhd, \ell, \lambda)$ be an MSC over $\mathcal{T}$. Then, M is prime.*

Proof. Let $w \in \mathbb{D}^*$, $n \geq 1$, and $e \in E$, and suppose $(e,e) \in [\![w^n]\!]_M$. We build the "projection" $\langle w \rangle \in \mathcal{N}^*$ of w to the alphabet $\mathcal{N}$ so that it can be applied to the topology $\mathcal{T}$. It is defined by $\langle \mathsf{proc} \rangle = \langle \mathsf{proc}^{-1} \rangle = \varepsilon$, $\langle \mathsf{msg}_a \rangle = a$, and $\langle \mathsf{msg}_a^{-1} \rangle = \overline{a}$.

From $(e,e) \in [\![w^n]\!]_M$, we deduce $\ell(e) \xrightarrow{\langle w \rangle^n} \ell(e)$, which implies $\ell(e) \xrightarrow{\langle w \rangle} \ell(e)$, since $\mathcal{T}$ is prime. Towards a contradiction, assume that $(e,e) \notin [\![w]\!]_M$ (which implies $w \neq \varepsilon$ and $n > 1$). Consider the unique event $e_1 \in E$ such that $(e, e_1) \in [\![w]\!]_M$. Due to $\ell(e) \xrightarrow{\langle w \rangle} \ell(e)$, we have either $e \lhd_{\mathsf{proc}}^+ e_1$ or $e_1 \lhd_{\mathsf{proc}}^+ e$. Suppose $e \lhd_{\mathsf{proc}}^+ e_1$. The other case is analogous. As $(e,e) \in [\![w^n]\!]_M$, there are $e_2, \ldots, e_n \in E$ such that $(e, e_1), (e_1, e_2), \ldots, (e_{n-1}, e_n) \in [\![w]\!]_M$. Thus, $(e, e_n) \in [\![w^n]\!]_M$. As MSCs obey a FIFO policy, we moreover have $e \lhd_{\mathsf{proc}}^+ e_1 \lhd_{\mathsf{proc}}^+ e_2 \lhd_{\mathsf{proc}}^+ \cdots \lhd_{\mathsf{proc}}^+ e_n$ and, therefore, $e \neq e_n$. But this contradicts $(e,e) \in [\![w^n]\!]_M$. We conclude that M is prime. $\square$

Sphere Automaton. Next, we adapt the construction of the sphere automaton from [2] to our setting. We build a PCA that is able to tell us whether, at a given event, a local formula holds. Again, we restrict to local formulas with only one free variable. To decide if a local formula holds, it is actually sufficient for the PCA to detect *spheres*, i.e., local neighborhoods of radius $R \in \mathbb{N}$. Let us define spheres formally. Let $\mathcal{T} = (V, \rightarrow, \pi)$ be a topology, $M = (E, \lhd, \ell, \lambda)$ be an MSC over $\mathcal{T}$, and $e \in E$. The *R-sphere* of M around e, denoted by $R\text{-Sph}(M, e)$, is the restriction of M to events of distance at most R from e. More precisely, it is defined as the structure $\mathcal{M} = (E', \lhd', \pi', \lambda', e)$ where

[4] The property is exploited in [2, Claim 4.1] without being called prime. In [B. Bollig. On the expressive power of 2-stack visibly pushdown automata. *Logical Methods in Computer Science*, 4(4:16), 2008.], a weaker property (which is implied by *prime*) is named *circular*. There, the sphere automaton is constructed for nested words.

$E' = \{f \in E \mid dist_M^{\sigma_0}(e,f) \leq R\}$, $\lhd'$ is given by $\lhd'_{\mathsf{proc}} = \lhd_{\mathsf{proc}} \cap (E' \times E')$ and $\lhd'_{\mathsf{msg}} = \lhd_{\mathsf{msg}} \cap (E' \times E')$, and π' and λ' are mappings with domain E': for all $e \in E'$, $\pi'(e) = \pi(\ell(e))$ and $\lambda'(e) = \lambda(e)$. Note that $\mathcal{M}$ is independent of a topology, as an event is mapped by π' to a process type rather than a process.

Let $\mathbb{S}_R \stackrel{\text{def}}{=} \{R\text{-Sph}(M,e) \mid M$ is an MSC over some topology and e is an event of $M\}$ denote the set of R-spheres that arise from MSCs, ranging over all topologies. Note that $\mathbb{S}_R$ is finite up to isomorphism. Interestingly, the sphere automaton that we are going to construct has similarities with the PCA built in Section 6. However, the crucial difference is that the sphere automaton is supposed to detect spheres, while the automaton from Section 6 evaluated topologies. So, the sphere automaton will guess a sphere, for each event e, and verify that the guessed sphere is indeed isomorphic to the R-sphere around e.

Lemma 2. *Let $R \in \mathbb{N}$. There are a PCA $\mathcal{A} = (S, Msg, \Delta, I, F)$ and a mapping $\nu : S \to \mathbb{S}_R$ such that the following hold, for all prime topologies $\mathcal{T} \in \mathbb{T}$ and all MSCs $M = (E, \lhd, \ell, \lambda)$ over $\mathcal{T}$:*

- *$M \in L_{\mathcal{T}}(\mathcal{A})$, and*
- *for all accepting runs ρ of $\mathcal{A}$ on M and all events $e \in E$, we have $\nu(\rho(e)) \cong R\text{-Sph}(M,e)$.*

We define the PCA $\mathcal{A} = (S, Msg, \Delta, I, F)$ with $S = Msg$ as follows:

States. A state $t \in S$ is either $\emptyset_p$ for some $p \in \mathcal{P}$ (the empty set with some annotated type; we set $proc(\emptyset_p) = p$ and let $\nu(t)$ be an arbitrary sphere), or a nonempty set of tuples $(\mathcal{M}, \gamma, \alpha, col)$ where $(\mathcal{M} = (E, \lhd, \pi, \lambda), \gamma) \in \mathbb{S}_R$, $\alpha \in E$ is the *active event*, and $col \in \{1, \ldots, 4 \cdot maxE^2 + 1\}$ is a color with $maxE$ the maximal number of events of an R-sphere from $\mathbb{S}_R$. The coloring is needed to distinguish isomorphic overlapping spheres. For nonempty t, we require that the following hold:

- there is a unique tuple $(\mathcal{M}, \gamma, \alpha, col) \in t$ such that $\gamma = \alpha$ (in that case, we set $\nu(t) = (\mathcal{M}, \gamma)$),
- for all $((E, \lhd, \pi, \lambda), \gamma, \alpha, col), ((E', \lhd', \pi', \lambda'), \gamma', \alpha', col') \in t$, we have $\pi(\alpha) = \pi'(\alpha')$ and $\lambda(\alpha) = \lambda'(\alpha')$ (we set $proc(t) = \pi(\alpha)$ and $label(t) = \lambda(\alpha)$), and
- if $(\mathcal{M}, \gamma, \alpha, col) \in t$ and $(\mathcal{M}, \gamma, \alpha', col) \in t$, then $\alpha = \alpha'$.

Initial and Final States. For all $p \in \mathcal{P}$, we let $I(p) = \{\emptyset_p\}$. Let G be the set of nonempty states $t \in S$ such that there is $(\mathcal{M}, \gamma, \alpha, col) \in t$ with α *not* $\lhd_{\mathsf{proc}}$-maximal in $\mathcal{M}$. We set $F = \bigwedge_{t \in G} \neg\langle \#(t) \geq 1 \rangle$.

Send Transitions. In the following, we let $\mathcal{M}$ refer to $(E, \lhd, \pi, \lambda)$. The triple $(t^-, !_m a, t) \in S \times \Sigma_{\mathcal{A}} \times S$ is contained in Δ if the following hold:

(1) $m = t$ and $label(t) = !a$,
(2) $proc(t^-) = proc(t)$,
(3) for all $(\mathcal{M}, \gamma, \alpha, col) \in t$ and $e \in E$, we have $e \lhd_{\mathsf{proc}} \alpha$ iff $(\mathcal{M}, \gamma, e, col) \in t^-$,

(4) for all $(\mathcal{M}, \gamma, \alpha, col) \in t^-$ and $e \in E$, we have $\alpha \lhd_{\mathsf{proc}} e$ iff $(\mathcal{M}, \gamma, e, col) \in t$,

(5) for all $(\mathcal{M}, \gamma, \alpha, col) \in t$, if α is $\lhd_{\mathsf{proc}}$-minimal in $\mathcal{M}$ and $t^- \neq \emptyset$, then $dist_{\mathcal{M}}^{\sigma_0}(\gamma, \alpha) = R$, and

(6) for all $(\mathcal{M}, \gamma, \alpha, col) \in t^-$, if α is $\lhd_{\mathsf{proc}}$-maximal in $\mathcal{M}$, then $dist_{\mathcal{M}}^{\sigma_0}(\gamma, \alpha) = R$.

Receive Transitions. The triple $(t^-, ?_m a, t) \in S \times \Sigma_{\mathcal{A}} \times S$ is contained in Δ if (2)–(6) as above as well as the following hold:

(7) $label(t) = ?a$,

(8) for all $(\mathcal{M}, \gamma, \alpha, col) \in t$ and $e \in E$, we have $e \lhd_{\mathsf{msg}} \alpha$ iff $(\mathcal{M}, \gamma, e, col) \in m$,

(9) for all $(\mathcal{M}, \gamma, \alpha, col) \in m$ and $e \in E$, we have $\alpha \lhd_{\mathsf{msg}} e$ iff $(\mathcal{M}, \gamma, e, col) \in t$.

This concludes the construction of $\mathcal{A}$. The correctness proof in the sense of Lemma 2 follows the same lines as that of [2]. To show that cycles in spheres are correctly simulated by a given input MSC M, we use the fact that M is prime. Let $\mathcal{T} = (V, \rightarrow, \pi)$ be a prime topology, $M = (E, \lhd, \ell, \lambda)$ be an MSC over $\mathcal{T}$ (i.e., according to Lemma 1, M is prime), and ρ be an accepting run of $\mathcal{A}$ on M. Let $e_0 \in E$ and $w \in \mathbb{D}^*$, and consider $\nu(\rho(e_0)) = (\mathcal{M} = (E', \lhd', \pi', \lambda'), \gamma)$. We show that, then, $(\gamma, \gamma) \in [\![w]\!]_{\mathcal{M}}$ (with the obvious meaning) implies $(e_0, e_0) \in [\![w]\!]_M$.

So, suppose $(\gamma, \gamma) \in [\![w]\!]_{\mathcal{M}}$. By the construction of $\mathcal{A}$, we have that, for all $n \geq 1$, there is $e_n \in E$ such that $(e_0, e_n) \in [\![w^n]\!]_M$ (cf. proof of [2, Claim 4.11]). In particular, $(e_n, e_{n+1}) \in [\![w]\!]_M$ for all $n \in \mathbb{N}$. Towards a contradiction, assume $e_1 \neq e_0$. Note that this implies $w \neq \varepsilon$. As M is prime, we have $e_2 \neq e_0$. But we also have $e_2 \neq e_1$: otherwise, there would be events $f_1, f_2, f \in E$ and $d \in \mathbb{D}$ such that $f_1 \neq f_2$, $(f_1, f) \in [\![d]\!]_M$, and $(f_2, f) \in [\![d]\!]_M$, which is a contradiction, as f can have at most one d-predecessor. Continuing this scheme, we get $e_n \notin \{e_0, \ldots, e_{n-1}\}$ for all $n \geq 1$. But this is a contradiction to the fact that E is finite. We deduce $(e_0, e_0) \in [\![w]\!]_M$.

B Missing Details for Proof of Theorem 3

We show the correctness of the PCA $\mathcal{A}$ constructed in Section 6:

Lemma 3. *For all topologies $\mathcal{T} \in \mathfrak{T}$ and all B-bounded MSCs M over $\mathcal{T}$, we have $M \models \forall y \chi(y)$ iff $M \in L_{\mathcal{T}}(\mathcal{A})$.*

The rest of the section is devoted to the proof of Lemma 3. So, suppose $\mathcal{T} = (V, \rightarrow, \pi) \in \mathfrak{T}$ and let $M = (E, \lhd, \ell, \lambda)$ be a B-bounded MSC over $\mathcal{T}$. Let $v \in V$ and $M_v = (Ev_v, \lhd_v, \ell_v, \lambda_v) = R\text{-Sph}(M, v)$. Moreover, let $\tau_v = ((U_v, \rightsquigarrow_v, \xi_v), v)$ denote $R\text{-Sph}(\mathcal{T}, v) \restriction M$. Note that, since M is B-bounded, M_v is B-bounded, too. Finally, let $\mathcal{B}_v = (S_v, \Delta_v, \iota_v, F_v) \stackrel{\text{def}}{=} \mathcal{B}_{\tau_v}$, which is a CA over τ_v.

"$\Rightarrow$": Suppose $M \models \forall y \chi(y)$. By Theorem 4, we have $M_v \in L(\mathcal{B}_v)$, for all $v \in V$. Thus, for all $v \in V$, there is an accepting run $\rho_v : Ev_v \to S_v$ of $\mathcal{B}_v$ on M_v. From the collection $(\rho_v)_{v \in V}$, we define a mapping $\rho : E \to S$ by

$$\rho(e) = \{(\tau_v, \ell(e), \rho_v(e), H_e) \mid v \in V \text{ such that } e \in Ev_v\} \tag{1}$$

where $H_e = \{a \in \mathcal{N} \mid \lambda(f) \in \{!a, ?a\}$ for some $f \in E$ with $f \lhd_{\mathsf{proc}}^* e\}$. Towards an appropriate initial state of $\mathcal{A}$ for M, we let $\zeta = (\zeta_v)_{v \in V_M}$ where

$$\zeta_v = \{(\tau_v, \ell(e), \iota_v(\ell(e)), \emptyset) \mid e \in Ev_v\}$$

We show that ρ is an accepting run of $\mathcal{A}$ on M, which implies $M \in L_{\mathcal{T}}(\mathcal{A})$.

Clearly, $\zeta_v \in S$, for all $v \in V_M$. So, let us prove that $\rho(e) \in S$ for all events $e \in E$. Suppose tuples $\kappa = (\mathcal{U} = (U, \rightsquigarrow, \xi), \gamma, \alpha, s, H) \in \rho(e)$ and $\kappa' = (\mathcal{U}' = (U', \rightsquigarrow', \xi'), \gamma', \alpha', s', H') \in \rho(e)$.

(a) By (1), we have $\xi(\alpha) = \xi'(\alpha') = \ell(e)$.
(b) Assume $\gamma = \alpha$ and $\gamma' = \alpha'$. By (1), $\kappa = (\tau_{\ell(e)}, \ell(e), \rho_{\ell(e)}(e), H_e)$ and $\kappa' = (\tau_{\ell(e)}, \ell(e), \rho_{\ell(e)}(e), H_e)$ so that $\kappa = \kappa'$.
(c) Assume $(\mathcal{U}, \gamma, \alpha) \cong (\mathcal{U}', \gamma', \alpha')$. By (1), this implies $\gamma = \gamma'$ (due to isomorphism, $\ell(e) \overset{w}{\rightsquigarrow} \gamma$ iff $\ell(e) \overset{w}{\rightsquigarrow}' \gamma'$ for all w, which is impossible if $\gamma \neq \gamma'$). We deduce $\kappa = \kappa'$.

Next, we show that ρ is a run of $\mathcal{A}$ on M. Let $e \in E$ und $a \in \mathcal{N}$ such that $\lambda(e) \in \{!a, ?a\}$. We show that a is enabled in $\rho(e)$. Let $v \in V$ such that $e \in Ev_v$. Moreover, let $u \in U_v$ and $w \in \mathcal{N}^{\leq 2R}$ such that wa is circular and $u \overset{w}{\rightsquigarrow}_v \ell(e)$. Since, $\tau_v = R\text{-Sph}(\mathcal{T}, v) \restriction M$ and $\mathcal{T} \in \mathfrak{T}$ with $\mathfrak{T}$ being $(2R+1)$-unambiguous, we have $\ell(e) \overset{a}{\rightsquigarrow}_v u$. Thus, a is enabled in $\rho(e)$.

Let $(e, f) \in \lhd_{\mathsf{msg}}$ and $a \in \mathcal{N}$ be the unique name satisfying $\ell(e) \xrightarrow{a} \ell(f)$. We show that $\rho_\zeta^-(e) \xRightarrow{!_{\rho(e)} a} \rho(e)$ and $\rho_\zeta^-(f) \xRightarrow{?_{\rho(e)} \overline{a}} \rho(f)$. We define a bijection $\Phi_e : \rho_\zeta^-(e) \to \rho(e)$ as follows:

Case 1: If e is $\lhd_{\mathsf{proc}}$-minimal, then we have

$$\rho_\zeta^-(e) = \{(\tau_v, \ell(e), \iota_{\tau_v}(\ell(e)), \emptyset) \mid v \in V \text{ such that } e \text{ is an event of } M_v\}.$$

In that case, we set $\Phi_e(\tau_v, \ell(e), \iota_v(\ell(e)), \emptyset) = (\tau_v, \ell(e), \rho_v(e), H_e)$.

Case 2: If e is not $\lhd_{\mathsf{proc}}$-minimal, then there is e^- such that $e^- \lhd_{\mathsf{proc}} e$. We have

$$\rho_\zeta^-(e) = \{(\tau_v, \ell(e^-), \rho_v(e^-), H_{e^-}) \mid v \in V \text{ such that } e \in Ev_v\}.$$

In that case, we set $\Phi_e(\tau_v, \ell(e^-), \rho_v(e^-), H_{e^-}) = (\tau_v, \ell(e), \rho_v(e), H_e)$.

Note that $|\rho_\zeta^-(e)| = |\rho(e)| = |\{v \in V \mid e \in Ev_v\}|$. So, Φ_e is well-defined and indeed a bijection. It remains to verify that Φ_e has the desired properties. We consider only Case 2. So, suppose $\Phi_e(\tau_v, \ell(e^-), \rho_v(e^-), H_{e^-}) = (\tau_v, \ell(e), \rho_v(e), H_e)$. We show (ST).

1. We have $(\tau_v, \ell(e^-)) = (\tau_v, \ell(e))$.
2. As ρ_v is a run, $(\rho_v(e^-), !_{\rho_v(e)} a, \rho_v(e)) \in \Delta_{\tau_v}$.
3. We have $e \in Ev_v$. By the definition of $M_v = R\text{-Sph}(M, v)$, we have that $\ell(e)$ has an a-successor in τ_v, or $dist_{\tau_v}(v, \ell(e)) = R$.
4. Clearly, $H_e = H_{e^-} \cup \{a\}$.

We define the bijection $\Phi_f : \rho^-_\zeta(f) \to \rho(f)$ and verify (RTa) accordingly. It remains to define the bijection $\hat{\Phi}_{(e,f)} : \rho(e)_{\overset{a}{\to}} \to \rho(f)_{\overset{\overline{a}}{\to}}$ as well as a mapping $\mu : \rho(f)_{\overset{\overline{a}}{\to}} \to \bigcup_{v \in V} S_v$.

We have

$$\begin{aligned}\rho(e)_{\overset{a}{\to}} = \{&(\tau_v, \ell(e), \rho_v(e), H_e) \mid \\ &v \in V \text{ such that } e \in Ev_v \text{ and } \ell(e) \overset{a}{\rightsquigarrow}_v u \text{ for some } u \in U_v\}\,, \\ \rho(f)_{\overset{a}{\to}} = \{&(\tau_v, \ell(f), \rho_v(f), H_f) \mid \\ &v \in V \text{ such that } f \in Ev_v \text{ and } \ell(f) \overset{\overline{a}}{\rightsquigarrow}_v u \text{ for some } u \in U_v\}\,.\end{aligned}$$

For $v \in V$ with $e \in Ev_v$ and $\ell(e) \overset{a}{\rightsquigarrow}_v u$ for some $u \in U_v$, we have $f \in Ev_v$. This follows from the definition of M_v. We set $\hat{\Phi}_{(e,f)}(\tau_v, \ell(e), \rho_v(e), H_e) = (\tau_v, \ell(f), \rho_v(f), H_f)$. Note that $\hat{\Phi}_{(e,f)}$ is bijective. Similarly, for $v \in V$ with $f \in Ev_v$ and $\ell(f) \overset{\overline{a}}{\rightsquigarrow}_v u$ for some $u \in U_v$, we have that $e \in Ev_v$. In that case, we set $\mu(\tau_v, \ell(f), \rho_v(f), H_f) = \rho_v(e)$. With this definition, (RTb) is directly verified. In (RTc), we have to show that, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in (\rho(e) \setminus \rho(e)_{\overset{a}{\to}}) \cup (\rho(f) \setminus \rho(f)_{\overset{\overline{a}}{\to}})$, we have $dist_{\mathcal{U}}(\gamma, \alpha) = R$. So, consider $(\tau_v, \ell(e), \rho_v(e), H_e)$ with $v \in V$ such that $e \in Ev_v$ and $\ell(e)$ does not have an a-successor in τ_v. Since, then, e is an unmatched event of M_v, we have $dist_{\tau_v}(v, \ell(e)) = R$ by the definition of MSC M_v. The reasoning for $(\mathcal{U}, \gamma, \alpha, s, H) \in \rho(f) \setminus \rho(f)_{\overset{\overline{a}}{\to}}$ is analogous.

It remains to show that ρ is accepting. Let G be the set of states $t \in S$ such that, for all $(\mathcal{U} = (U, \rightsquigarrow, \xi), \gamma, \alpha, s, H) \in t$, we have $s \in F_{(\mathcal{U},\gamma)}(\alpha)$ and $\{a \in \mathcal{N} \mid \alpha \overset{a}{\rightsquigarrow} u \text{ for some } u \in U\} \subseteq H$. We have to show that, for all $e \in E$ that are $\lhd_{\mathsf{proc}}$-maximal, we have $\rho(e) \in G$.

So, let $e \in E$ be $\lhd_{\mathsf{proc}}$-maximal. Suppose $v \in V$ such that $e \in E_v$. We have to show that $\rho_v(e) \in F_v(\ell(e))$ and $\{a \in \mathcal{N} \mid \ell(e) \overset{a}{\rightsquigarrow}_v u \text{ for some } u \in U_v\} \subseteq H_e$, i.e., for all $a \in \mathcal{N}$ such that $\ell(e)$ has an outgoing a-edge in τ_v, there is $f \lhd^*_{\mathsf{proc}} e$ such that $\lambda(f) \in \{!a, ?a\}$. The former holds since ρ_v is an accepting run. The latter holds since, by definition, the MSC M_v "covers" the a-labeled edge.

"$\Leftarrow$": Now, suppose $M \in L_{\mathcal{T}}(\mathcal{A})$. There is an accepting run $\rho : E \to S$ of $\mathcal{A}$ on M, say, with initial state $\zeta = (\zeta)_{v \in V_M}$ for M. In particular, for all $(e, f) \in \lhd_{\mathsf{msg}}$, we have $\rho^-_\zeta(e) \xRightarrow{!_{\rho(e)} a} \rho(e)$ and $\rho^-_\zeta(f) \xRightarrow{?_{\rho(e)} \overline{a}} \rho(f)$.

We will show $M \models \forall y \chi(y)$. By Theorem 4, it is sufficient to prove $M_{v_0} \in L(\mathcal{B}_{v_0})$ for all $v_0 \in V_M$, i.e., to determine accepting runs $\rho_{v_0} : Ev_{v_0} \to S_{v_0}$ of $\mathcal{B}_{v_0}$ on M_{v_0}. Let $e, f \in E$ such that $(e, f) \in \lhd_{\mathsf{msg}}$. Suppose $a \in \mathcal{N}$ such that $\ell(e) \overset{a}{\to} \ell(f)$. According to (ST) and (RT), consider the unique mappings $\Phi_e : \rho^-_\zeta(e) \to \rho(e)$, $\Phi_f : \rho^-_\zeta(f) \to \rho(f)$, $\mu_f : \rho(f)_{\overset{\overline{a}}{\to}} \to \bigcup_{\theta \in R\text{-}Sub(\mathfrak{T})} S_\theta$, and $\hat{\Phi}_{(e,f)} : \rho(e)_{\overset{a}{\to}} \to \rho(f)_{\overset{\overline{a}}{\to}}$.

So pick $v_0 \in V_M$. For all events $e \in E$ of M located on v_0, the state $\rho(e)$ contains exactly one tuple of the form $(\mathcal{U}, \gamma, \gamma, s, H)$ (where sphere center and active node coincide). Set $\kappa_e = (\mathcal{U}, \gamma, \gamma, s, H)$ and $\rho_{v_0}(e) = s$. Note that, by (ST) and (RT), $\theta_{v_0} \overset{\text{def}}{=} (\mathcal{U}, \gamma)$ is invariant along all events on v_0.

We claim

$$\tau_{v_0} \cong \theta_{v_0} \,. \tag{2}$$

Recall that, hereby, $\tau_{v_0} = R\text{-Sph}(\mathcal{T}, v_0) \upharpoonright M_{v_0}$. In particular, (2) implies that $\rho_{v_0}(e) \in S_{v_0}$ for all events e on v_0. Before we prove (2), we define ρ_{v_0} for all other events of M_{v_0}. In doing so, whenever ρ_{v_0} is defined on e (so that κ_e has also been determined), it will be defined for the direct process predecessor e^- and process successor e^+ (if they exist), using the bijections Φ_e and Φ_{e^+}. The tuples κ_{e^-} and κ_{e^+} are defined accordingly.

So suppose that we defined ρ_{v_0} for all events of M_{v_0} that are located on nodes v with $0 \leq \mathit{dist}_{\tau_{v_0}}(v_0, v) \leq k < R$. Consider an event $f \in \mathit{Ev}_{v_0}$ that is located on some v with $\mathit{dist}_{\tau_{v_0}}(v_0, v) = k + 1$. There is $e \in \mathit{Ev}_{v_0}$ located on a node with distance k to v_0 (i.e., $\rho_{v_0}(e)$ is already defined) such that e and f form a message.

- Suppose $(e, f) \in \lhd_{\mathsf{msg}}$ where $\lambda_{v_0}(e) = !a$, i.e., e sends a message via interface a. Assume $\kappa_e = (\mathcal{U} = (U, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$ (by induction, this will mean $\rho_{v_0}(e) = s$). Due to (2), we have $\kappa_e \in \rho_{v_0}(e)\xrightarrow{a}$. By, (RTb), there is $\alpha' \in U$ such that $\alpha \overset{a}{\rightsquigarrow} \alpha'$ and $\hat{\Phi}_{(e,f)}(\kappa_e) = (\mathcal{U}, \gamma, \alpha', s', H')$ for some s', H'. We set $\rho_{v_0}(f) = s'$.
- Suppose $(f, e) \in \lhd_{\mathsf{msg}}$ where $\lambda_{v_0}(e) = ?a$, i.e., e receives via interface a. Assume $\kappa_e = (\mathcal{U} = (U, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$, i.e., $\rho_{v_0}(e) = s$. Due to (2), we have $\kappa_e \in \rho_{v_0}(e)\xrightarrow{a}$. Again, due to (RTb), there is $\alpha' \in U$ such that $\alpha \overset{a}{\rightsquigarrow} \alpha'$, and we have $\hat{\Phi}^{-1}_{(f,e)}(\kappa_e) = (\mathcal{U}, \gamma, \alpha', s', H')$ for some s', H'. We set $\rho_{v_0}(f) = s'$.

Note that ρ_{v_0} is well defined. So let us show that ρ_{v_0} is a run of $\mathcal{B}_{v_0}$ on M_{v_0}. Pick $e \in \mathit{Ev}_{v_0}$.

- Suppose e is of type $!a$. We have $\rho^-_\zeta(e) \xRightarrow{!_{\rho(e)}a} \rho(e)$. Assume first that e is process-minimal on some process w. Suppose $\kappa_e = (\mathcal{U}, \gamma, \alpha, s, H)$. Recall that $\rho_{v_0}(e) = s$. Thanks to (ST), $\rho^-_\zeta(e)$ contains $(\mathcal{U}, \gamma, \alpha, s^-, \emptyset)$, with $s^- = \iota_{v_0}(w)$ and $(s^-, !_s a, s) \in \Delta_{v_0}$. Moreover, $H = \{!a\}$. If e is not process-minimal, then e has a process-predecessor e^-. By (ST), κ_{e^-} is of the form $(\mathcal{U}, \gamma, \alpha, \rho_{v_0}(e^-), H^-)$, for some H^-, and we have $(\rho_{v_0}(e^-), !_s a, s) \in \Delta_{v_0}$ and $H = H^- \cup \{!a\}$.
- Suppose e is of type $?a$ and let $e^-, f \in \mathit{Ev}_{v_0}$ be such that $e^- \lhd_{\mathsf{proc}} e$ and $f \lhd_{\mathsf{msg}} e$. Suppose $\kappa_e = (\mathcal{U} = (U, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$, $\kappa_{e^-} = (\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-)$, and $\kappa_f = (\mathcal{U}', \gamma', \alpha', s', H')$. By (RT), $(\mathcal{U}, \gamma) = (\mathcal{U}', \gamma')$, $(\mathcal{U}^-, \gamma^-, \alpha^-) = (\mathcal{U}, \gamma, \alpha)$, $\alpha \overset{a}{\rightsquigarrow} \alpha'$, and $\alpha' \overset{\overline{a}}{\rightsquigarrow} \alpha$. Finally, $(s^-, ?_{s'} a, s) \in \Delta_{v_0}$. The cases where e does not have a process-predecessor or where e is unmatched are similar.

Next, we show that ρ_{v_0} is accepting. Let $e \in \mathit{Ev}_{v_0}$ be process-maximal and suppose $\kappa_e = (\mathcal{U}, \gamma, \alpha, s, H)$. From the fact that ρ is accepting, we can deduce that $s \in F_{v_0}(\alpha)$.

To finish the proof, it remains to show (2), i.e., $\tau_{v_0} \cong \theta_{v_0}$. This is done using the H-component of a state as well as the fact that $\mathfrak{T}$ is $(2R + 1)$-unambiguous.

For $v \in V$ and a triple $(\mathcal{U}, \gamma, \alpha)$, we write $(\mathcal{U}, \gamma, \alpha) \in \rho(v)$ if there are s, H, and an event $e \in E_v$ such that $(\mathcal{U}, \gamma, \alpha, s, H) \in \rho(e)$.

Suppose $\tau_{v_0} = (\mathcal{W} = (W, \twoheadrightarrow, \pi), v_0)$ and $\theta_{v_0} = (\mathcal{U} = (U, \rightsquigarrow, \xi), u_0)$. For $d \in \{0, \ldots, R\}$, let $((W_d, \twoheadrightarrow_d, \pi_d), v_0)$ be the restriction of τ_{v_0} to elements of distance at most d from v_0 in $\mathcal{W}$. Similarly, let $((U_d, \rightsquigarrow_d, \xi_d), u_0)$ be the restriction of θ_{v_0} to elements of distance at most d from u_0 in $\mathcal{U}$.

The following claim implies (2):

Claim 1. For all $d \in \{0, \ldots, R\}$, there is an isomorphism

$$h_d : ((W_d, \twoheadrightarrow_d, \pi_d), v_0) \to ((U_d, \rightsquigarrow_d, \xi_d), u_0)$$

such that, for all $v \in W_d$, we have

$$(\mathcal{U}, u_0, h_d(v)) \in \rho(v)\,.$$

Proof. The claim holds for $d = 0$, since $(\mathcal{U}, u_0, u_0) \in \rho(v_0)$.

Now, suppose the claim holds for some $d < R$ so that we have an isomorphism $h_d : ((W_d, \twoheadrightarrow_d, \pi_d), v_0) \to ((U_d, \rightsquigarrow_d, \xi_d), u_0)$. Towards h_{d+1}, we extend the domain of h_d to elements $v' \in W_{d+1} \setminus W_d$. So let $v, v' \in W$ such that $dist_{\mathcal{W}}(v_0, v) = d$, $dist(v_0, v') = d + 1$, and v and v' are connected by an edge in τ_{v_0}. Moreover, set $u = h_d(v)$.

Suppose $v \overset{a}{\twoheadrightarrow} v'$ with $a \in \mathcal{N}$. Since $dist_{\mathcal{W}}(v_0, v) < R$, we also have, by induction hypothesis, $dist_{\mathcal{U}}(u_0, u) < R$. Suppose $(e, e') \in \lhd_{\mathsf{msg}} \cup \lhd_{\mathsf{msg}}^{-1}$ such that e is located on v and e' is located on v'. By induction hypothesis, we have $(\mathcal{U}, \gamma, u, s, H) \in \rho(e)$ for some s, H. By (ST) and (RT), there is $u' \in U$ such that $(\mathcal{U}, \gamma, u', s, H) \in \rho(e')$ and $u \overset{a}{\rightsquigarrow} u'$. Note that, since h_d is an isomorphism, $u' \in U_{d+1} \setminus U_d$. Set $h_{d+1}(v') = u'$. This is well-defined and does not depend on the concrete choice of v or a: if we obtained another, distinct element u'', we would have $(\mathcal{U}, \gamma, u'', s, H) \in \rho(e')$, which is a contradiction to the definition of the set of states of $\mathcal{A}$. We define h_{d+1} to agree with h_d on W_d.

It remains to show that h_{d+1} is an isomorphism. First, we show that h_{d+1} is surjective. Let $u, u' \in U$ with distance d and $d + 1$, respectively, from u_0 such that $u \overset{a}{\rightsquigarrow} u'$. Let $v = h_d^{-1}(u)$. By induction hypothesis, we have $(\mathcal{U}, u_0, u) \in \rho(v)$. As $dist_{\mathcal{U}}(u_0, u) < R$ and ρ is an accepting run, there is $v' \in W$ satisfying $v \overset{a}{\twoheadrightarrow} v'$ and $(\mathcal{U}, u_0, u') \in \rho(v')$. Thus, $h_{d+1}(v') = u'$ so that h_{d+1} is surjective.

Now, let us show that h_{d+1} is indeed an isomorphism. Take $v_1, v_2 \in W_d$ and $v_1', v_2' \in W_{d+1} \setminus W_d$ as well as $w_1, w_2 \in \mathcal{N}^d$ and $a_1, a_2 \in \mathcal{N}$ such that

- $v_0 \overset{w_1}{\twoheadrightarrow} v_1 \overset{a_1}{\twoheadrightarrow} v_1'$, and
- $v_0 \overset{w_2}{\twoheadrightarrow} v_2 \overset{a_2}{\twoheadrightarrow} v_2'$.

For $i = 1, 2$, let $u_i = h_{d+1}(v_i)$ and $u_i' = h_{d+1}(v_i')$. We will show that

- $v_1' = v_2'$ iff $u_1' = u_2'$ (so that h_{d+1} is injective), and
- $v_1' \overset{a}{\twoheadrightarrow} v_2'$ iff $u_1' \overset{a}{\rightsquigarrow} u_2'$, for all $a \in \mathcal{N}$.

First, suppose $v_1' = v_2'$. Let $w = \overline{a_1 w_1} w_2 a_2$. Then, $v_1' \xrightarrow{w} v_1'$. As $\mathfrak{T}$ is $(2R+1)$-unambiguous, $\tau_{v_0}, \theta_{v_0} \in R\text{-}\mathit{Sub}(\mathfrak{T})$, $|w| \leq 2R$, and $u_1' \overset{w}{\rightsquigarrow} u_2'$, this implies $u_1' = u_2'$. The same argument applies when we start with $u_1' = u_2'$.

Next, suppose $v_1' \xrightarrow{a} v_2'$, for some $a \in \mathcal{N}$. Since $\rightarrow$ is the relation belonging to τ_{v_0}, a message is exchanged between two events located on v_1' and v_2', respectively. It follows that there is $e \in E_{v_1'}$ such that a is enabled in $\rho(e)$. Let $w = \overline{a_2 w_2} w_1 a_1$. Then, $|w| \leq 2R$ and wa is circular. As a is enabled in $\rho(e)$, $(\mathcal{U}, u_0, u_1') \in \rho(v_1')$, and $u_2' \overset{w}{\rightsquigarrow} u_1'$, we have $u_1' \overset{a}{\rightsquigarrow} u_2'$.

Finally, suppose $u_1' \overset{a}{\rightsquigarrow} u_2'$. Let $w = \overline{a_2 w_2} w_1 a_1$. As ρ is an accepting run, a message has to be exchanged between v_1' and a process v such that $v_1' \xrightarrow{a} v$. We have $u_2' \overset{w}{\rightsquigarrow} u_1' \overset{a}{\rightsquigarrow} u_2'$ and $v_2' \xrightarrow{w} v_1' \xrightarrow{a} v$. As $\mathfrak{T}$ is $(2R+1)$-unambiguous and $|wa| = 2R+1$, we have $v = v_2'$. Thus, $v_1' \xrightarrow{a} v_2'$.

This concludes the proof of Claim 1. □

C Proof of Theorem 5

Theorem 5. *There exists a sentence $\varphi \in \mathrm{FO}[\lhd_{\mathsf{proc}}^*, \lhd^*]$ such that, for all PCA $\mathcal{A}$ over $\mathcal{P}$ and $\mathcal{N}$, there is $\mathcal{T} \in \mathfrak{T}_{\mathrm{tree}}$ with $L^1_{\mathcal{T}}(\mathcal{A}) \neq L^1_{\mathcal{T}}(\varphi)$.*

Proof. A picture over the set $\Sigma = \{\circ, \odot, \bullet\}$ of colors is a rectangular matrix with $m \geq 1$ rows and $n \geq 1$ columns, and with entries in Σ. An example picture of dimension $(m = 3, n = 8)$ is

$$P = \begin{pmatrix} \circ & \circ & \odot & \bullet & \odot & \circ & \odot & \circ \\ \odot & \odot & \circ & \bullet & \circ & \odot & \circ & \odot \\ \circ & \odot & \circ & \bullet & \circ & \odot & \circ & \circ \end{pmatrix}$$

The coordinates of a picture can be ordered by relations $\leq_1$ (for columns) and $\leq_2$ (for rows). We let $(i,j) \leq_1 (i',j')$ if $i \leq i'$ and $j = j'$, and $(i,j) \leq_2 (i',j')$ if $i = i'$ and $j \leq j'$. Accordingly, FO logic over pictures uses the binary predicates $x \leq_1 y$ and $x \leq_2 y$, as well as unary predicates $\eta(x)$ with $\eta \in \Sigma$. Let $\mathfrak{P}_=$ be the set of those pictures that are the concatenation $P_1 Q P_2$ of pictures of the same height, where Q is a single column with entries $\bullet$, and P_1 and P_2 are pictures over $\{\circ, \odot\}$ whose sets of column labelings coincide. The example picture P above is contained in $\mathfrak{P}_=$. Note that $\mathfrak{P}_=$ is FO-definable by a sentence that requires that, for all coordinates x in the first row, there has to be a coordinate y on the opposite side of the picture (i.e., beyond column Q) such that their respective column labelings coincide. This can indeed be expressed using $\leq_1$ and $\leq_2$. In [17], Thomas exploits the picture language $\mathfrak{P}_=$ to show that FO over pictures using $\leq_1$ and $\leq_2$ is incomparable with EMSO using the direct successor relations of $\leq_1$ and $\leq_2$, which is equivalent to *graph acceptors*.

To transfer that result and its proof to our setting, we use MSCs to encode pictures. An MSC encoding of a picture is based on a tree topology of a particular form. Let $\mathfrak{T}_{\mathrm{pict}}$ be the set of topologies $\mathcal{T}^n_{\mathrm{pict}} \in \mathfrak{T}_{\mathrm{tree}}$, with $n \geq 1$, as depicted

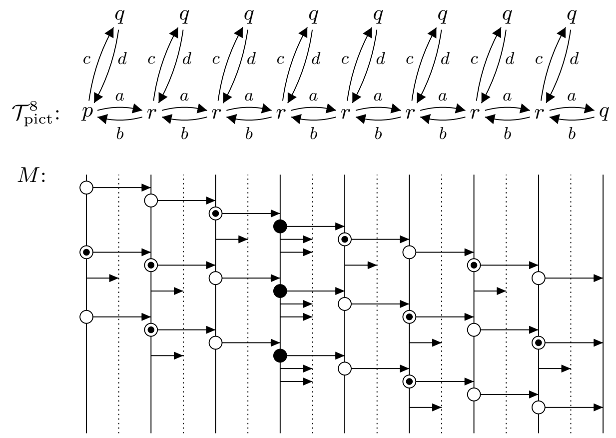

Fig. 8. The communication topology $\mathcal{T}^8_{\mathrm{pict}}$ over $\mathcal{P} = \{p, q, r\}$ and $\mathcal{N} = \{a, b, c, d\}$, as well as an MSC that encodes a picture

in Figure 8 for $n = 8$. Note that $\mathcal{T}^n_{\mathrm{pict}}$ has $2n + 1$ vertices. As $\mathfrak{T}_{\mathrm{pict}}$ is a subset of $\mathfrak{T}_{\mathrm{tree}}$, it is k-unambiguous, for all $k \in \mathbb{N}$, and contains only prime topologies. The MSC M that encodes picture P (see above) is shown in Figure 8. An event performing $!a$ corresponds to a picture coordinate. If it is immediately followed, on the same process line, by *one* event performing $!c$, then its entry is $\odot$. If it is immediately followed by *two* events performing $!c$, then its entry is $\bullet$. Otherwise, it is $\circ$. Note that M is 1-bounded.

The set of all valid picture encodings is definable by a formula Ψ from $\mathrm{FO}[\lhd^*_{\mathsf{proc}}, \lhd^*]$, i.e., for all $n \geq 1$, $L_{\mathcal{T}^n_{\mathrm{pict}}}(\Psi)$ is the set of MSCs that correspond to pictures with n columns. Let $\mathfrak{L}^n_=$ be the set of MSCs over $\mathcal{T}^n_{\mathrm{pict}}$, that encode a picture from $\mathfrak{P}_=$ with n columns. In $\mathrm{FO}[\lhd^*_{\mathsf{proc}}, \lhd^*]$, we can define formulas *is-coordinate*(x) and $\eta(x)$ for all $\eta \in \Sigma$ in the obvious way. Moreover, $x \leq_1 y$ (which corresponds to walking down in a picture) is given, for MSCs, by $x \lhd^*_{\mathsf{proc}} y$. Finally, $x \leq_2 y$ (which corresponds to walking rightwards in a picture) is given by $x \lhd^* y \wedge \forall x'(x \lhd_{\mathsf{proc}} x' \rightarrow \neg(x' \lhd^* y))$. It follows that there is a sentence $\varphi \in \mathrm{FO}[\lhd^*_{\mathsf{proc}}, \lhd^*]$ such that $\mathfrak{L}^n_= = L_{\mathcal{T}^n_{\mathrm{pict}}}(\varphi) = L^1_{\mathcal{T}^n_{\mathrm{pict}}}(\varphi)$, for all $n \geq 1$.

Now suppose that there is a PCA $\mathcal{A} = (S, Msg, \Delta, I, F)$ such that, for all $\mathcal{T} \in \mathfrak{T}_{\mathrm{pict}}$, we have $L^1_{\mathcal{T}}(\mathcal{A}) = L^1_{\mathcal{T}}(\varphi)$. In any accepting run of $\mathcal{A}$ on an MSC $M \in \mathfrak{L}^n_=$ encoding picture $P_1 Q P_2$, say, of height m, all the information needed to compare P_1 and P_2 has to be present in the constant number of equivalence classes induced by F and in the assignments of states and messages to the events located on Q, i.e., on the $\bullet$-labeled column. There are $(|S| \times |Msg|)^m$ such assignments. On the other hand, there are $2^{2^m} - 1$ nonempty sets of words of length m over $\{\circ, \odot\}$, which exceeds $(|S| \times |Msg|)^m$ for sufficiently large m. Thus, there is an accepting run, possibly wrt. a different topology, on the encoding of $P'_1 Q P'_2$ for some P'_1, P'_2 that induce distinct sets of column labelings, a contradiction. $\square$