

Logic for Communicating Automata with Parameterized Topology

Benedikt Bollig

LSV, ENS Cachan, CNRS & Inria, France

Email: bollig@lsv.ens-cachan.fr

Abstract—We introduce parameterized communicating automata (PCA) as a model of systems where finite-state processes communicate through FIFO channels. Unlike classical communicating automata, a given PCA can be run on any network topology of bounded degree. The topology is thus a parameter of the system. We provide various Büchi-Elgot-Trakhtenbrot theorems for PCA, which roughly read as follows: Given a logical specification φ and a class of topologies \mathfrak{T} , there is a PCA that is equivalent to φ on all topologies from \mathfrak{T} . We give uniform constructions which allow us to instantiate \mathfrak{T} with concrete classes such as pipelines, ranked trees, grids, rings, etc. The proofs build on a locality theorem for first-order logic due to Schwentick and Barthelmann, and they exploit concepts from the non-parameterized case, notably a result by Genest, Kuske, and Muscholl.

I. INTRODUCTION

The Büchi-Elgot-Trakhtenbrot theorem states that finite automata and monadic second-order (MSO) logic over words are expressively equivalent [6], [9], [24]. 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 [22], asynchronous automata [25], and graph acceptors [23], to mention just a few.

In recent years, an analogous question has been studied for communicating automata (CA). A CA consists of several finite-state processes that 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, whose nodes represent the *events* that are observed during an execution. Its edge relation $\triangleleft = \triangleleft_{\text{proc}} \cup \triangleleft_{\text{msg}}$ visualizes causal dependencies between events. Edges from $\triangleleft_{\text{proc}}$ connect consecutive events performed by a process, and edges from $\triangleleft_{\text{msg}}$ connect send events with their corresponding receives. Logical characterizations have been established for unrestricted CA [3] and channel-bounded CA [17], [20], [13]. All these results require the underlying communication topology, which provides a set of processes and channels between them, to be *fixed*.

Now, it is a natural question to ask for an automaton that realizes a given logical specification for a *class* of topologies (for example, all grid topologies, no matter what the size of the grid is). This is what this paper is about, i.e., we aim for Büchi-Elgot-Trakhtenbrot theorems in a setting with non-fixed, *parameterized* topology.

In a first step, we introduce parameterized communicating automata (PCA). Unlike classical CA, a given PCA can be run on any network topology of *bounded degree* (such as pipelines, ranked trees, grids, or rings). PCA are a conservative extension of CA and, as such, also recognize sets of MSCs. Our study is centered around the following question, which depends on a given logical specification φ and a given class \mathfrak{T} of topologies:

Is there a PCA \mathcal{A} that is equivalent to φ on all topologies $\mathcal{T} \in \mathfrak{T}$ (when \mathcal{A} is run on \mathcal{T} , it accepts precisely the MSCs over \mathcal{T} that satisfy φ)?

If the answer is affirmative, then we say that formula φ is *realizable* for \mathfrak{T} . This paper investigates realizability wrt. several logics and instances of \mathfrak{T} in a unifying framework. We consider standard first-order and existential MSO logic, $\text{FO}[\sigma]$ and $\text{EMSO}[\sigma]$, respectively. Here, $\sigma \subseteq \{\triangleleft_{\text{proc}}, \triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}, \triangleleft^*, \sim\}$ is the collection of binary relation symbols that are available in the logic. All symbols are self-explanatory apart from \sim , which allows us to say that two events are executed by the same process.

Our first results settle the limits of what we can hope for:

- (i) There is an $\text{FO}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}]$ -formula that is *not* realizable for the class of “ring forests” (unions of ring topologies).
- (ii) There is an $\text{FO}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}, \triangleleft^*]$ -formula that is *not* realizable already for the class of binary trees.

This shows that we have to restrict both, the topologies and the logic, and that we are left with only a small margin for positive results. However, we are able to provide various Büchi-Elgot-Trakhtenbrot theorems:

- (iii) Every $\text{EMSO}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}, \sim]$ -formula is realizable for the classes of pipelines, ranked trees, grids, and rings.
- (iv) When we suppose that the channels of a PCA are bounded, then every $\text{EMSO}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$ -formula is realizable for the classes of pipelines, ranked trees, grids, and rings.

In fact, we obtain (iii) and (iv) as *corollaries* of more general, uniform statements: it is shown that every EMSO formula is realizable for \mathfrak{T} whenever \mathfrak{T} is *unambiguous*. Intuitively, this rules out cycle patterns that a PCA is not able to detect on its own. Indeed, the classes of pipelines, ranked trees, and grids are all unambiguous so that we get realizability as a direct corollary. To capture also rings, some additional arguments are needed.

Note that, for (iii) and (iv) to apply, a class \mathfrak{T} of topologies has to be *fixed in advance*. The construction of a PCA \mathcal{A}

from a formula φ is uniform, but it crucially depends on \mathfrak{T} . That is, though \mathcal{A} can still be run on any other topology of bounded degree, it is only guaranteed to be equivalent to φ when it is applied to a topology from \mathfrak{T} . Now, when we fix \mathfrak{T} , we have in mind that we run \mathcal{A} only on topologies $\mathcal{T} \in \mathfrak{T}$. For that reason, \mathcal{A} does not have to check membership of \mathcal{T} in \mathfrak{T} . However, it will have to collect some topological information to identify bounded subtopologies among those from \mathfrak{T} . In fact, the translation from logic to PCA builds on a locality theorem for first-order logic due to Schwentick and Barthelmann, which states that satisfaction of a formula in a structure can be reduced to satisfaction of a normal-form formula in bounded portions of the same structure [21]. This allows us to apply notions from the setting with fixed topologies, notably a result by Genest, Kuske, and Muscholl [13]. Hereby, the assumption that topologies are of bounded degree is crucial.

For $\text{EMSO}[\prec_{\text{proc}}, \prec_{\text{msg}}]$ (without process order), we provide a variation of the theme: Every formula φ can be translated into a PCA that is equivalent to φ on all *prime* topologies. In that case, the construction is *independent* of a concrete class of topologies (once the bound on the degree has been fixed). Note that pipelines, trees, and grids are all prime.¹

Finally, every PCA \mathcal{A} can be transformed into a formula from $\text{EMSO}[\prec_{\text{proc}}, \prec_{\text{msg}}]$ that is equivalent to \mathcal{A} on *all* topologies of bounded degree. Thus, overall, we indeed establish a variety of Büchi-Elgot-Trakhtenbrot theorems for PCA.

Related Work: It seems that neither PCA nor expressiveness of parameterized systems in general in terms of logic have been considered in the literature.

In [18], Jacobs and Bloem study parameterized synthesis, where a temporal-logic specification is transformed into a system of processes that are arranged in a token ring of arbitrary size. Building on [10], the idea is to reduce parameterized synthesis to distributed synthesis over a bounded architecture. Though we also use a reduction to a bounded case, our framework differs from [18] in the model (asynchronous rather than token communication), in the topologies, and in the logic.

In parameterized verification, one aims at showing that a *given* system is correct independently of the number of processes or the communication topology [5], [15], [1], [4], [8]. Our approach is different, since we *generate* a system model from a high-level specification.

There have been a variety of automata constructions that exploit normal forms of first-order logic [23], [21], [11]. We actually borrow a technique from [11], but the overall framework is quite different.

Finally, our contribution intersects the area of distributed algorithms. Indeed, the way a PCA evaluates a (sub)topology is similar to constructing a map of an *anonymous graph* [7]. In particular, our notion of unambiguous classes of topologies is in the spirit of universal sequences. There are also methods to evaluate graphs versus logical specifications [16]. Though

¹“Prime” is a property of *single* topologies, while “unambiguous” refers to *sets* of topologies.

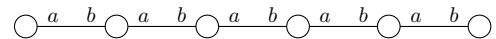
all those techniques do not seem to be directly applicable, it will be worthwhile to explore possible connections further.

Outline: Sections II–IV settle basic notions such as topologies, MSCs, PCA, and MSO logic. In Section V, we argue that we will have to restrict both topologies and logic. Sections VI and VII present the above-mentioned Büchi-Elgot-Trakhtenbrot theorems, respectively. We conclude in Section VIII. Due to space constraints, proofs are only sketched or omitted. All details can be found in the appendix.

II. PRELIMINARIES

A. Communication Topologies

A (communication) topology² is made up of single entities such as $\frac{b}{\circ} \frac{a}{\circ}$. Here, a process (represented by the circle) is equipped with two interfaces, a and b . The interfaces allow the process to communicate with its environment. When they are connected to interfaces of other processes, we obtain a topology. A simple pipeline topology is depicted below.



Thus, a topology is essentially a graph, whose nodes are processes that can communicate with adjacent processes via their interfaces. The pipeline, for example, will allow a process to execute actions $!a$ and $?a$ in order to send a message to (receive a message from, respectively) its right neighbor, if it exists. Accordingly, $!b$ and $?b$ refer to the left neighbor.

Let us define topologies formally. Throughout the paper, unless stated otherwise, we fix a nonempty finite set $\mathcal{N} = \{a, b, c, \dots\}$ of (*interface*) *names*. When we talk about a concrete process, we may also say *interface* instead of name.

Definition 1. A topology over \mathcal{N} is a pair $\mathcal{T} = (P, \dashv)$ where

- P is the nonempty finite set of processes, and
- $\dashv \subseteq P \times \mathcal{N} \times \mathcal{N} \times P$ is the edge relation.

We write $p \xrightarrow{a, b} q$ for $(p, a, b, q) \in \dashv$, which signifies that the a -interface of p points to q , and the b -interface of q points to p . We require that, whenever $p \xrightarrow{a, b} q$, the following hold:

- $p \neq q$,
- $q \xrightarrow{b, a} p$, and
- for all $a', b' \in \mathcal{N}$ and $q' \in P$ such that $p \xrightarrow{a', b'} q'$, we have $a = a'$ iff $q = q'$.

By (a), a topology does not contain self-loops. Condition (b) says that two adjacent processes are *mutually* connected (in other words, a topology is “undirected”). By (c), a name points to at most one process, and two distinct names point to distinct processes.

We usually consider topologies up to isomorphism. The set of all topologies over \mathcal{N} is denoted by $\mathbb{T}_{\mathcal{N}}$.

²What we call topology is sometimes termed *architecture*. It seems that, however, in a parameterized setting, the term topology is more custom. Actually, our definition does not quite correspond to architectures from the literature, since processes are not connected by *channels* but *interfaces*. The latter are more appropriate in our setting, as they support the view that a process is specified independently of a concrete topology.

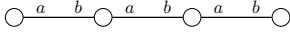


Fig. 1. Pipeline $\mathcal{T}_{\text{lin}}^4$

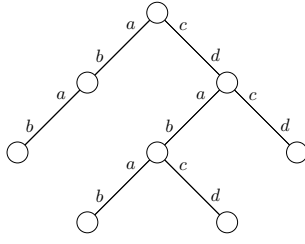


Fig. 2. Tree topology

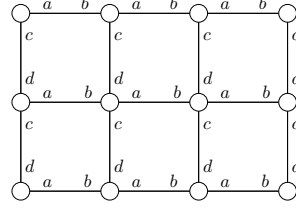


Fig. 3. Grid $\mathcal{T}_{\text{grid}}^{3,4}$

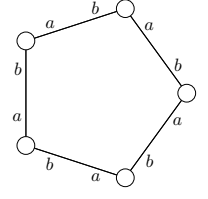


Fig. 4. Ring $\mathcal{T}_{\text{ring}}^5$

Given a topology $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$, a PCA will run identical subautomata on processes of the same *type*. We define $\text{type}_{\mathcal{T}} : P \rightarrow 2^{\mathcal{N}}$ by $\text{type}_{\mathcal{T}}(p) := \{a \in \mathcal{N} \mid \text{there are } b \in \mathcal{N} \text{ and } q \in P \text{ such that } p \dashv_{a,b} q\}$. Thus, $\text{type}_{\mathcal{T}}(p)$ contains those interfaces of p that are connected to some other process.

Remark 1. One can also define types independently of \mathcal{N} , in terms of an extra finite set *Types*, and include a mapping $\text{type} : P \rightarrow \text{Types}$ in the topology. In our setting, this can be encoded by choosing $\mathcal{N} \times \text{Types}$ as new set of names. In a topology, an edge $p \dashv_{a,b} q$ is then replaced by an edge with names $(a, \text{type}(p))$ and $(b, \text{type}(q))$. All definitions can be adapted easily and all results hold verbatim in this alternative setting.

Example 1. Let us identify some typical topology classes. We give an informal description. The precise definitions are as expected and can be found in Appendix IX.

Pipelines: A pipeline is a topology over $\{a, b\}$, as already indicated above. Recall that interface a points to the right neighbor of a process (if it exists), while b is connected to the left neighbor. Accordingly, the leftmost process has type $\{a\}$, the rightmost process has type $\{b\}$, and all inner processes have type $\{a, b\}$. The pipeline of length $n \geq 2$ is denoted by $\mathcal{T}_{\text{lin}}^n$. Figure 1 depicts $\mathcal{T}_{\text{lin}}^4$. We let $\mathfrak{T}_{\text{lin}} = \{\mathcal{T}_{\text{lin}}^n \mid n \geq 2\} \subseteq \mathbb{T}_{\{a,b\}}$ denote the set of all pipelines.

Trees: We suppose that trees are binary, but we could consider arbitrary ranked alphabets. An example tree is depicted in Figure 2. Hence, a tree is a topology over $\{a, b, c, d\}$ where interface a points to the left son, and c to the right son of a process, while b and d are their respective “dual” interfaces. We suppose that a tree has at least two processes. The type of a leaf is either $\{b\}$ or $\{d\}$. The type of the root is either $\{a\}$, $\{c\}$, or, as is the case in Figure 2, $\{a, c\}$. The set of all tree topologies is denoted by $\mathfrak{T}_{\text{tree}} \subseteq \mathbb{T}_{\{a,b,c,d\}}$. Note that pipelines can be seen as a special case of trees.

Grids: A grid is a topology over $\{a, b, c, d\}$ where processes are arranged in a matrix. It is uniquely given by its number $m \geq 1$ of rows and its number $n \geq 1$ of columns. Again, there should be at least two processes so that we suppose $\max\{m, n\} \geq 2$. A process that is not located on the border has a right and a left neighbor (following a and b , respectively), but also adjacent nodes below and above (following c and d). Let $\mathcal{T}_{\text{grid}}^{m,n}$ denote the grid with m rows and n columns. An example is illustrated in Figure 3. By $\mathfrak{T}_{\text{grid}} = \{\mathcal{T}_{\text{grid}}^{m,n} \mid m, n \geq 1 \text{ with } \max\{m, n\} \geq 2\} \subseteq \mathbb{T}_{\{a,b,c,d\}}$, we denote the set of all grids. Again, a pipeline is a special case of a grid.

Rings: A ring can be seen as a pipeline where the endpoints are glued together. Thus, it is a topology over $\{a, b\}$ in which every process has type $\{a, b\}$. The ring with $n \geq 3$ processes is denoted by $\mathcal{T}_{\text{ring}}^n$. Figure 4 illustrates $\mathcal{T}_{\text{ring}}^5$. We denote the set of all rings by $\mathfrak{T}_{\text{ring}} = \{\mathcal{T}_{\text{ring}}^n \mid n \geq 3\} \subseteq \mathbb{T}_{\{a,b\}}$. \square

Remark 2. For many concrete topology classes of bounded degree such as pipelines, grids, or rings, the names are canonical so that fixing them in advance is not a restriction. Moreover, in most cases considered in the literature, a few (sometimes even one) process types will do. In particular, it is a common assumption that processes in a ring are indistinguishable [5], [10], [18]. However, one could also assume a distinguished leader process (and add another interface name just for the purpose of identifying the leader; cf. also Remark 1).

B. 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 system. It is formalized as a labeled directed acyclic graph whose nodes, the *events*, are associated with processes from a given communication topology. Events are linked by process edges $\triangleleft_{\text{proc}}$ and message edges $\triangleleft_{\text{msg}}$. The process edges connect consecutive events of one process, and message edges connect send events with their corresponding receives according to a FIFO policy.

Definition 2. An MSC over $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ is a triple $M = (E, \triangleleft, \ell)$ where

- E is the nonempty finite set of events,
- $\triangleleft \subseteq E \times E$ is the acyclic edge relation, which is partitioned into $\triangleleft_{\text{proc}}$ and $\triangleleft_{\text{msg}}$, and
- $\ell : E \rightarrow P$ determines the location of an event in the topology; for $p \in P$, we let $E_p := \{e \in E \mid \ell(e) = p\}$.

We require that the following hold:

- $\triangleleft_{\text{proc}}$ is a union $\bigcup_{p \in P} \triangleleft_p$ where each $\triangleleft_p \subseteq E_p \times E_p$ is the direct-successor relation of some total order on E_p ,
- there are a partition $E = E_1 \uplus E_2$ and a bijection $\mu : E_1 \rightarrow E_2$ such that $\triangleleft_{\text{msg}} = \{(e, \mu(e)) \mid e \in E_1\}$,
- for all $(e, f) \in \triangleleft_{\text{msg}}$, there are $a, b \in \mathcal{N}$ such that $\ell(e) \dashv_{a,b} \ell(f)$ (communication is restricted to adjacent processes), and
- for all $(e, f), (e', f') \in \triangleleft_{\text{msg}}$ such that $\ell(e) = \ell(e')$ and $\ell(f) = \ell(f')$, we have $e \triangleleft_{\text{proc}}^* e'$ iff $f \triangleleft_{\text{proc}}^* f'$ (FIFO).

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

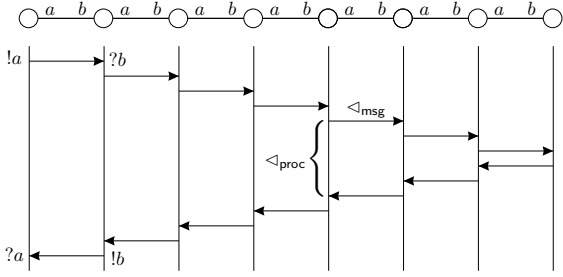


Fig. 5. MSC M_{lin}^s over topology $\mathcal{T}_{\text{lin}}^s$

Given an MSC $M = (E, \triangleleft, \ell)$, we define a mapping $act_M : E \rightarrow \{!a, ?a \mid a \in \mathcal{N}\}$ that associates with an event the action that it executes: for $(e, f) \in \triangleleft_{\text{msg}}$ and $a, b \in \mathcal{N}$ such that $\ell(e) \xrightarrow{a} \ell(f)$, we let $act_M(e) = !a$ and $act_M(f) = ?b$.

Example 2. Figure 5 illustrates an MSC, call it M_{lin}^s , over topology $\mathcal{T}_{\text{lin}}^s \in \mathfrak{T}_{\text{lin}}$. The behavior of each process is represented by a top-down process line. Arrows between process lines determine the relation $\triangleleft_{\text{msg}}$, connecting send events with their receive events. For illustration, some events are labeled with the actions that they execute. We may consider M_{lin}^s as the execution of a P2P protocol: a request from the leftmost process is forwarded by $n - 2$ inner processes of type $\{a, b\}$, until it reaches the rightmost process. An acknowledgement is then relayed back to the first process along the same way backwards. Figure 8 depicts an example MSC over $\mathcal{T}_{\text{grid}}^{2,5}$. \square

Our main result will deal with systems that have (existentially) B -bounded channels, for some $B \geq 1$ [13], [14]. 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 $M = (E, \triangleleft, \ell)$ over $\mathcal{T} = (P, \xrightarrow{\quad})$ is any total order $\preceq \subseteq E \times E$ satisfying $\triangleleft^* \subseteq \preceq$. Then, \preceq is called B -bounded if, for all $f \in E$, $p, q \in P$, and $a, b \in \mathcal{N}$ such that $p \xrightarrow{a} q$, we have $|\{e \in E \mid e \preceq f, \ell(e) = p, \text{ and } act_M(e) = !a\}| - |\{e \in E \mid e \preceq f, \ell(e) = q, \text{ and } act_M(e) = ?b\}| \leq B$. In other words, in any prefix of \preceq , there are no more than B pending messages, in every “channel” (p, q) . Now, we say that MSC M is B -bounded if it has a B -bounded linearization. For example, for all $n \geq 2$, the MSC M_{lin}^s (cf. Figure 5) is 1-bounded, because its (only) linearization is 1-bounded.

III. PARAMETERIZED COMMUNICATING AUTOMATA

Next, we introduce PCA. Their definition does not depend on a topology, but only on \mathcal{N} . The language of a PCA, a set of MSCs, is then parameterized by a topology.

Definition 3. A parameterized communicating automaton (PCA) over \mathcal{N} is a tuple $\mathcal{A} = (S, \text{Msg}, \Delta, I, F)$ where

- S is the finite set of states,
- Msg is the finite set of messages,
- $I : (2^{\mathcal{N}} \setminus \{\emptyset\}) \rightarrow 2^S$ assigns to each nonempty process type its initial states,

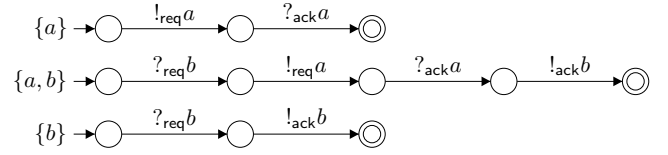


Fig. 6. PCA \mathcal{A} over $\{a, b\}$

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

The class of PCA over \mathcal{N} is denoted by $\mathbb{PCA}_{\mathcal{N}}$. A PCA over \mathcal{N} can be run on any topology $\mathcal{T} = (P, \xrightarrow{\quad}) \in \mathbb{T}_{\mathcal{N}}$. Suppose $p \xrightarrow{a} q$ for processes $p, q \in P$ and names $a, b \in \mathcal{N}$. When p executes a transition $(s, !_m a, s') \in \Delta$, it changes its local state from s to s' and writes m into the FIFO channel (p, q) . The message m can then be received by process q executing a transition with action $?_m b$. However, 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 a 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 IV). Let $\mathcal{T} = (P, \xrightarrow{\quad}) \in \mathbb{T}_{\mathcal{N}}$ be a topology and $M = (E, \triangleleft, \ell)$ be an MSC over \mathcal{T} . A run of \mathcal{A} on M will be a mapping $\rho : E \rightarrow S$. Intuitively, $\rho(e)$ is the state that process $\ell(e)$ reaches after executing $e \in E$. To define when ρ is a run, we will need some more notation.

Set $P_M := \{p \in P \mid E_p \neq \emptyset\}$, which is the set of active processes of M . A (global) initial state of \mathcal{A} for M is a tuple $\iota = (\iota_p)_{p \in P_M}$ where $\iota_p \in I(\text{type}_{\mathcal{T}}(p))$ for all $p \in P_M$. Given ι and $\rho : E \rightarrow S$ (a possible MSC run), we define another mapping $\rho_{\iota}^- : E \rightarrow S$, which returns the source state of a transition. For $(f, e) \in \triangleleft_{\text{proc}}$, we let $\rho_{\iota}^-(e) = \rho(f)$; for a $\triangleleft_{\text{proc}}$ -minimal event $e \in E$, we let $\rho_{\iota}^-(e) = \iota_{\ell(e)}$.

Now, a mapping $\rho : E \rightarrow S$ is called a run of \mathcal{A} on M if there is an initial state $\iota = (\iota_p)_{p \in P_M}$ for M such that, for all $(e, f) \in \triangleleft_{\text{msg}}$, there are $a, b \in \mathcal{N}$ and $m \in \text{Msg}$ satisfying $\ell(e) \xrightarrow{a} \ell(f)$, $\rho_{\iota}^-(e) \xrightarrow{!_m a} \rho(e)$, and $\rho_{\iota}^-(f) \xrightarrow{?_m b} \rho(f)$.

To determine if ρ is accepting, we define a multiset $h_{\rho} : S \rightarrow \mathbb{N}$ over S that counts how often each state occurs as the terminal state of an active process. For $s \in S$, we let $h_{\rho}(s) = |\{e \in E \mid e \text{ is } \triangleleft_{\text{proc}}\text{-maximal and } \rho(e) = s\}|$. We say that 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} . For a topology \mathcal{T} , the set of MSCs over \mathcal{T} that are accepted by \mathcal{A} is denoted by $L_{\mathcal{T}}(\mathcal{A})$. Finally, we let $L_{\mathcal{T}}^B(\mathcal{A})$ be the restriction of $L_{\mathcal{T}}(\mathcal{A})$ to B -bounded MSCs.

Example 3. Consider the PCA \mathcal{A} over $\{a, b\}$ from Figure 6. The acceptance condition F is simply the conjunction of formulas $\neg(\#(s) \geq 1)$ with s ranging over the states *without* double circle. Recall 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. Note that MSC M_{lin}^8 is the only MSC that is accepted by \mathcal{A} over $\mathcal{T}_{\text{lin}}^8$ (cf. Example 2). We actually have $L_{\mathcal{T}_{\text{lin}}^n}(\mathcal{A}) = L_{\mathcal{T}_{\text{lin}}^1}(\mathcal{A}) = \{M_{\text{lin}}^n\}$ for all $n \geq 2$. \square

Remark 3. The multiset h_ρ defined to evaluate the acceptance condition of a PCA does not include any states of non-active (i.e., idle) processes. So, a PCA cannot express “the topology has at least 5 processes”, but only “at least 5 processes are active”. In principle, one could include idle processes as well. However, this has to be reflected in the logic (cf. Theorem 9). One possibility is to consider processes as single events. But, apart from involving a more technical presentation, this does not seem to be natural. Alternatively, one could consider a two-sorted logic to reason about both events and processes. In that case, one very quickly exceeds the capability of PCA to evaluate a topology, as their runs rely on the messages that occur in an MSC. A two-valued logic also goes against the intuition that PCA accept behaviors rather than topologies.

The section concludes with some closure properties of PCA.

Theorem 1. *PCA are closed under union and intersection: For all $\mathcal{A}_1, \mathcal{A}_2 \in \text{PCA}_{\mathcal{N}}$, there are PCA \mathcal{A} and \mathcal{B} over \mathcal{N} such that, for all topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\mathcal{A}_1) \cup L_{\mathcal{T}}(\mathcal{A}_2)$ and $L_{\mathcal{T}}(\mathcal{B}) = L_{\mathcal{T}}(\mathcal{A}_1) \cap L_{\mathcal{T}}(\mathcal{A}_2)$.*

The construction of \mathcal{A} and \mathcal{B} is easy. The only (minor) subtle point is the acceptance condition (see Appendix X).

Theorem 2. *PCA are not closed under complementation: There is $\mathcal{A} \in \text{PCA}_{\{a,b\}}$ such that, for all $\mathcal{B} \in \text{PCA}_{\{a,b\}}$, we have $L_{\mathcal{T}_{\text{lin}}^2}(\mathcal{B}) \neq \{M \mid M \text{ MSC over } \mathcal{T}_{\text{lin}}^2\} \setminus L_{\mathcal{T}_{\text{lin}}^2}(\mathcal{A})$.*

Theorem 2 is an immediate consequence of the fact that fixed-topology CA over two processes are not completable [3]. Finally, non-deterministic PCA are strictly more expressive than deterministic ones (we do not give the formal definitions). This already holds over 1-bounded MSCs, which follows from the case of fixed-topology CA and requires a topology with five processes and five interface names [14].

IV. 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.

A. Monadic Second-Order Logic

The set $\text{MSO}_{\mathcal{N}}$ of MSO formulas over \mathcal{N} is given by:

$$\begin{aligned} \varphi ::= & \\ & act(x) = !a \mid act(x) = ?a \mid a \in type(x) \mid \\ & x \triangleleft_{\text{proc}} y \mid x \triangleleft_{\text{proc}}^* y \mid x \triangleleft_{\text{msg}} y \mid x \triangleleft^* y \mid x \sim y \mid \\ & x = y \mid x \in X \mid \neg \varphi \mid \varphi \vee \psi \mid \exists x \varphi \mid \exists X \varphi \end{aligned}$$

where $a \in \mathcal{N}$, 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 $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi \equiv \neg\varphi \vee \psi$, and $\forall x \varphi \equiv \neg\exists x \neg\varphi$.

The set $\text{FO}_{\mathcal{N}}$ of *first-order* formulas is the fragment of $\text{MSO}_{\mathcal{N}}$ without second-order quantification $\exists X$. Moreover, $\text{EMSO}_{\mathcal{N}}$ (*existential MSO*) is the set of formulas of the form $\exists X_1 \dots \exists X_n \varphi$ with $\varphi \in \text{FO}_{\mathcal{N}}$.

A formula is evaluated wrt. an MSC $M = (E, \triangleleft, \ell)$ over some topology $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$. Free variables x and X are interpreted by a mapping \mathcal{I} as an event $\mathcal{I}(x) \in E$ and a set of events $\mathcal{I}(X) \subseteq E$, respectively. For η of the form $!a$ or $?a$, the atomic formula $act(x) = \eta$ is true if $act_M(\mathcal{I}(x)) = \eta$. Formula $a \in type(x)$ is true if $a \in type_{\mathcal{T}}(\ell(\mathcal{I}(x)))$, i.e., a is contained in the type of the process where $\mathcal{I}(x)$ is located. Formula, $x \triangleleft_{\text{proc}}^* y$ is satisfied if $\mathcal{I}(x) \triangleleft_{\text{proc}}^* \mathcal{I}(y)$. Moreover, $x \sim y$ holds true if $\ell(\mathcal{I}(x)) = \ell(\mathcal{I}(y))$, i.e., $\mathcal{I}(x)$ and $\mathcal{I}(y)$ are located on the same process. Other formulas are interpreted as expected. Though, even in $\text{FO}_{\mathcal{N}}$, some binary predicates are mutually expressible in terms of others (e.g., $\triangleleft_{\text{proc}}$ and \sim in terms of $\triangleleft_{\text{proc}}^*$, and $\triangleleft_{\text{proc}}^*$ in terms of \triangleleft^* and \sim), we include all of them explicitly in the logic. They will be used in fragments in which they would no longer be expressible.

Let $\sigma \subseteq \{\triangleleft_{\text{proc}}, \triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}, \triangleleft^*, \sim\}$ be a nonempty set of relation symbols. The logics $\text{FO}_{\mathcal{N}}[\sigma]$ and $\text{EMSO}_{\mathcal{N}}[\sigma]$ restrict $\text{FO}_{\mathcal{N}}$ and $\text{EMSO}_{\mathcal{N}}$, respectively: instead of $\{\triangleleft_{\text{proc}}, \triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}, \triangleleft^*, \sim\}$, we can only access the relation symbols from σ . Our main (positive) result will concern the logic $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$ (recall that $\triangleleft_{\text{proc}}$ and \sim can be expressed in terms of $\triangleleft_{\text{proc}}^*$).

Let $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$ be a topology, and let $\varphi \in \text{MSO}_{\mathcal{N}}$ 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 . Let $L_{\mathcal{T}}^B(\varphi)$ be the restriction of $L_{\mathcal{T}}(\varphi)$ to B -bounded MSCs.

Example 4. We will consider two $\text{FO}_{\{a,b\}}$ -sentences. First, $\varphi_1 = \forall x (act(x) = ?b \rightarrow \exists y (x \triangleleft_{\text{proc}}^* y \wedge act(y) = !b))$ says that every process that receives a message from its b -interface, eventually sends a message through b . Note that $M_{\text{lin}}^n \in L_{\mathcal{T}_{\text{lin}}^n}(\varphi_1)$ for all $n \geq 2$ (cf. Example 2). Next, let $\varphi_2 = \exists x \exists y (b \notin type(x) \wedge a \notin type(y) \wedge x \triangleleft_{\text{msg}} y)$. Interpreted over pipelines, φ_2 says that the leftmost process sends a message to the rightmost process. We have $M_{\text{lin}}^n \in L_{\mathcal{T}_{\text{lin}}^n}(\varphi_2)$ iff $n = 2$. \square

B. Locality of FO Logic

Next, we state a locality theorem due to Schwentick and Barthelmann [21].³ It formalizes the intuition that first-order logic can only reason about local *neighborhoods*, which include elements whose distance from a given center is bounded by a parameter that depends on the formula.

³Gaifman’s normal form appears to be more difficult to deal with in our context.

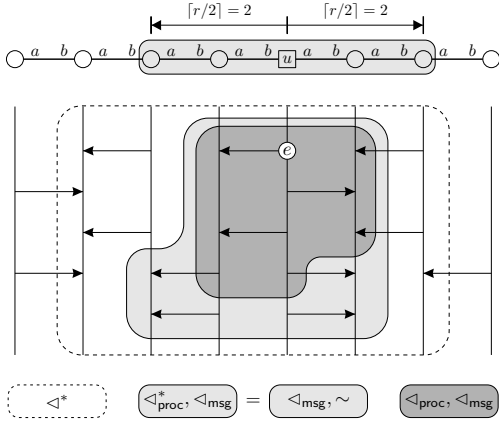


Fig. 7. Distance $r = 3$ in an MSC depending on the signature

Fix a nonempty set $\sigma \subseteq \{\langle_{\text{proc}}, \langle_{\text{proc}}^*, \langle_{\text{msg}}, \langle^*, \sim\}$ of relation symbols. Let $M = (E, \langle, \ell)$ be an MSC over some topology $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$. The distance $\text{dist}_M^\sigma(e, f)$ 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, let $\sigma = \{\langle_{\text{proc}}^*, \langle_{\text{msg}}\}$. Then, $\text{dist}_M^\sigma(e, f)$ refers to the distance in the (undirected) graph $(E, \langle_{\text{proc}}^* \cup (\langle_{\text{proc}}^*)^{-1} \cup \langle_{\text{msg}} \cup \langle_{\text{msg}}^{-1})$ so that $\text{dist}_M^\sigma(e, f) \leq 1$ for all $e, f \in E$ such that $\ell(e) = \ell(f)$, and $\text{dist}_M^\sigma(e, f) = \text{dist}_M^\sigma(f, e) = 1$ for all $(e, f) \in \langle_{\text{msg}}$.

Example 5. Let M be the MSC over $\mathcal{T}_{\text{lin}}^8$ from Figure 7 and let e be the distinguished event given by the white circle. The set of events f such that $\text{dist}_M^\sigma(e, f) \leq 3$ depends on σ , and is illustrated for $\{\langle^*\}$, $\{\langle_{\text{proc}}^*, \langle_{\text{msg}}\}$ (inducing the same set as $\{\langle_{\text{msg}}, \sim\}$), and $\{\langle_{\text{proc}}, \langle_{\text{msg}}\}$. \square

Let $r \geq 1$. A formula $\chi \in \text{FO}_{\mathcal{N}}[\sigma]$ 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 $\text{FO}_{\mathcal{N}}[\sigma]$ -formula by replacing each subformula of the form $\exists z\psi$ with $\exists z(\text{dist}^\sigma(y, z) < r \wedge \psi)$, and each subformula of the form $\forall z\psi$ with $\forall z(\text{dist}^\sigma(y, z) < r \rightarrow \psi)$. Here, $\text{dist}^\sigma(y, z) < r$ denotes the obvious $\text{FO}_{\mathcal{N}}[\sigma]$ -formula. We use strict inequality for technical reasons (cf. [11]). Adapted to our setting, [21] yields the following:

Theorem 3 (Schwentick & Barthelmann, [21]). *Let $\varphi \in \text{FO}_{\mathcal{N}}[\sigma]$. There are $r \geq 1$ and $\varphi' = \exists x_1 \dots \exists x_n \forall y \chi \in \text{FO}_{\mathcal{N}}[\sigma]$ such that χ is (r, σ) -local around y and, for all topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\varphi')$.*

V. NEGATIVE RESULTS

Recall that we are interested in realizability of formulas φ for a class \mathfrak{T} of topologies: Is there a PCA \mathcal{A} such that $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$ for all $\mathcal{T} \in \mathfrak{T}$. In this section, we show that such a PCA does not always exist.

A. Restrictions on Topologies Are Necessary

In fact, there is a sentence from $\text{FO}_{\{a,b\}}[\langle_{\text{proc}}, \langle_{\text{msg}}]$ that is not realizable for the class of *all* topologies over $\{a, b\}$. This

even holds for the class $\mathfrak{T}_{\text{ring}}^* \subseteq \mathbb{T}_{\{a,b\}}$ of *ring forests* and when we restrict to 1-bounded MSCs. A ring forest is a disjoint union of an arbitrary number of rings (possibly containing several copies of one and the same ring).

Theorem 4. *There exists a sentence $\varphi \in \text{FO}_{\{a,b\}}[\langle_{\text{proc}}, \langle_{\text{msg}}]$ such that, for all PCA $\mathcal{A} \in \text{PCA}_{\{a,b\}}$, there is $\mathcal{T} \in \mathfrak{T}_{\text{ring}}^*$ with $L_{\mathcal{T}}^1(\mathcal{A}) \neq L_{\mathcal{T}}^1(\varphi)$.*

The proof of Theorem 4 reveals that PCA have limited ability to “detect” cycles in an MSC and in a topology (cf. Appendix XI). In the following sections, we will, therefore, restrict the “cyclic behavior” of a class of topologies (of a single topology, respectively).

In Section VI, the construction of a PCA indeed requires a class \mathfrak{T} of topologies to be given in advance. We show that certain formulas are realizable for all classes \mathfrak{T} that are *unambiguous*: one can tell by looking at a sequence $w \in (\mathcal{N} \times \mathcal{N})^*$ of edge labels whether w produces a cycle or not, in any topology of \mathfrak{T} . This notion excludes the set of ring forests exploited in Theorem 4, but it captures the classes of pipelines, trees, and grids, as well as singleton rings and the class of “almost all” rings (which will finally allow us to cover the class of all rings with one single PCA).

In Section VII, the construction of a PCA from a given formula does not depend on a class of topologies, but only on \mathcal{N} . It makes sure that the synthesized PCA agrees with the formula on all *prime* topologies. This forbids cycles with a periodic labeling (as it occurs in rings), but includes all pipelines, trees, and grids.

B. Restrictions on Logic Are Necessary

Next, we argue that we have to restrict the logic, too. In fact, when we take $\text{FO}_{\mathcal{N}}$ (i.e., first-order logic with all binary predicates), then the negative result even holds for the class of trees (actually, for simple bus 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 [13].

Theorem 5. *There exists a sentence $\varphi \in \text{FO}_{\{a,b,c,d\}}$ such that, for all PCA $\mathcal{A} \in \text{PCA}_{\{a,b,c,d\}}$, there is $\mathcal{T} \in \mathfrak{T}_{\text{tree}}$ with $L_{\mathcal{T}}^1(\mathcal{A}) \neq L_{\mathcal{T}}^1(\varphi)$.*

The proof (cf. Appendix XII) uses a technique from [23], which was employed to show that FO (with reflexive transitive closure relations) and a local variant of EMSO are incomparable over *pictures*.

We briefly discuss Theorem 5. Let $\varphi \in \text{FO}_{\mathcal{N}}[\sigma]$, for some $\sigma \subseteq \{\langle_{\text{proc}}, \langle_{\text{proc}}^*, \langle_{\text{msg}}, \langle^*, \sim\}$, and suppose r is the radius associated with φ according to Theorem 3. Satisfaction of φ in an MSC M essentially depends on the σ -neighborhoods of the latter (informally, the σ -neighborhood of an event e is the substructure of M , including the *actions*, induced by all elements f such that $\text{dist}_M^\sigma(e, f) \leq r$). When σ contains \langle^* , there is no a priori bound on the size of the σ -neighborhood of e : it may feature events f that are far from e in terms of the number of messages that separate them. In other words,

$dist_M^\sigma(e, f)$ can be small for $\sigma = \{\triangleleft^*\}$, but large for $\sigma = \{\triangleleft_{proc}^*, \triangleleft_{msg}\}$. This is exemplified in Figure 7. On the other hand, the exchange of information in a PCA relies on messages and is restricted to processes that are adjacent in the topology. This intuitively explains the mismatch between automata and logic that Theorem 5 reveals.

The size of a $\{\triangleleft_{proc}^*, \triangleleft_{msg}\}$ -neighborhood is a priori unbounded as well. However, unlike a $\{\triangleleft^*\}$ -neighborhood, it has “bounded width”, since it spans over a bounded area of the *underlying topology*: Figure 7 illustrates that, provided $\triangleleft^* \notin \sigma$, a *topology* neighborhood of radius $\lceil r/2 \rceil$ is indeed sufficient to cover all events f with $dist_M^\sigma(e, f) \leq r$. As our topologies are structures of bounded degree, there are only a bounded number of such topology neighborhoods. This allows us to take advantage of results on CA over fixed topologies. We pursue this idea in Section VI to show realizability of $EMSO_{\mathcal{N}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$ - and $EMSO_{\mathcal{N}}[\triangleleft_{proc}, \triangleleft_{msg}, \sim]$ -formulas for unambiguous classes of topologies (as discussed in the previous subsection).

Any $\{\triangleleft_{proc}, \triangleleft_{msg}\}$ -neighborhood in an MSC also spans over a bounded topology (cf. Figure 7). But here, we can go one step further: up to isomorphism, there are only a bounded number of $\{\triangleleft_{proc}, \triangleleft_{msg}\}$ -neighborhoods. Using this, we will build, in Section VII, a PCA that evaluates a formula by “computing” such neighborhoods in an MSC. The construction is independent of a class of topologies. However, the resulting PCA is only guaranteed to be equivalent to the given formula when it is applied to *prime* topologies.

VI. EMSO VS. PCA OVER UNAMBIGUOUS TOPOLOGY CLASSES

A. Main Result and Consequences

Following the discussion in the previous section, we will consider logics that discard \triangleleft^* . Particular attention is paid to $EMSO_{\mathcal{N}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$, containing the process-order relation.

Recall that the $\{\triangleleft_{proc}^*, \triangleleft_{msg}\}$ -neighborhood of an event is covered by a bounded area in the underlying topology. We exploit this to translate a formula into a PCA by simulating several fixed-topology CA in parallel. In fact, in our construction of a PCA, a process will have to determine its (bounded) topology neighborhood, so that it can launch a corresponding fixed-topology CA. To some extent, it can verify whether it is the source of a w -labeled path, for a given $w \in (\mathcal{N} \times \mathcal{N})^*$. In general, however, there is no means to discover whether w forms a cycle (cf. Theorem 4). We will, therefore, consider topology classes \mathfrak{T} in which some words are *unambiguous*, i.e., they always form a cycle, or never form a cycle in \mathfrak{T} . For example, $(a, b)(c, d)(b, a)(d, c)$ is unambiguous for the class of grids: it gives rise to a cycle whenever it is applicable. As we deal with bounded topology neighborhoods, we have to consider only words up to a certain length.

Let $w = (a_1, b_1) \dots (a_n, b_n) \in (\mathcal{N} \times \mathcal{N})^*$. The length n of w is denoted by $|w|$. For a topology $(P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ and processes $p, q \in P$, we write $p \dashv^w q$ if there is a w -labeled path from p to q , i.e., there are $p_0, \dots, p_n \in P$ such that $p = p_0 \dashv_{a_1} b_1 p_1 \dashv_{a_2} b_2 \dots \dashv_{a_n} b_n p_n = q$.

Definition 4. Let $k \in \mathbb{N}$ and let $\mathfrak{T} \subseteq \mathbb{T}_{\mathcal{N}}$ be a class of topologies. We say that \mathfrak{T} is k -unambiguous if, for all $w \in (\mathcal{N} \times \mathcal{N})^*$ with $|w| \leq k$, all topologies $(P, \dashv), (P', \dashv') \in \mathfrak{T}$, and all processes $p, q \in P$ and $p', q' \in P'$ such that $p \dashv^w q$ and $p' \dashv'^w q'$, we have $p = q$ iff $p' = q'$.

In other words, if a topology from \mathfrak{T} admits a cycle of length $\leq k$ with label w , then following w (if possible) will always form a cycle, in any topology from \mathfrak{T} . Note that “unambiguous” is a property of a *class* of topologies. It captures the classes of pipelines, trees, and grids. Moreover, it will allow us to apply our results to ring topologies.

Lemma 1. The classes \mathfrak{T}_{lin} , \mathfrak{T}_{tree} , \mathfrak{T}_{grid} , and $\{\mathcal{T}_{ring}^n \mid n \geq \max\{3, k+1\}\}$ are all k -unambiguous, for every $k \in \mathbb{N}$. Moreover, $\{\mathcal{T}\}$ is k -unambiguous for all $\mathcal{T} \in \mathfrak{T}_{ring}$ and $k \in \mathbb{N}$.

For $\varphi \in EMSO_{\mathcal{N}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$, let $r_\varphi \geq 1$ denote the radius associated with the first-order kernel of φ according to Theorem 3. Note that an exponential upper bound for r_φ was given in [19]. We now present our main result:

Theorem 6. Let $\varphi \in EMSO_{\mathcal{N}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$ be a sentence, $B \geq 1$, and $\mathfrak{T} \subseteq \mathbb{T}_{\mathcal{N}}$ be an $(r_\varphi + 2)$ -unambiguous set of topologies. There is a PCA $\mathcal{A} \in \mathbb{PCA}_{\mathcal{N}}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\varphi)$.

Before we prove Theorem 6, let us discuss it and state some consequences. First, note that we have to commit to a class \mathfrak{T} of topologies before constructing the PCA. This is also what usually happens in practice: one has a “concrete” class of topologies in mind when writing a formula. For example, a specifier may want to synthesize a PCA that is equivalent to the given formula on all grids.⁴ Though, a priori, different topology classes give rise to different PCA, we give a uniform construction and proof. By Lemma 1, we can then instantiate \mathfrak{T} in Theorem 6 with various classes so that we obtain the following corollary (for simplicity, we consider pipelines and rings as topologies over $\{a, b, c, d\}$):

Corollary 1. Let $\varphi \in EMSO_{\{a,b,c,d\}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$ be a sentence, $B \geq 1$, and \mathfrak{T} be any of the following:

- the set of pipeline topologies,
- the set of grid topologies,
- the set of tree topologies,
- the set of ring topologies with at least $r_\varphi + 3$ nodes, or
- the singleton set $\{\mathcal{T}\}$ where \mathcal{T} is any ring topology.

Then, there is a PCA $\mathcal{A} \in \mathbb{PCA}_{\{a,b,c,d\}}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\varphi)$.

The construction where one single ring topology is given is *not* an immediate consequence of a corresponding result from the setting with fixed topologies [13]. The reason is that CA over fixed topologies have an initial state per *process*, while PCA have initial states per *process type*, which is a priori weaker. Now, given a formula φ , Corollary 1 gives us a way

⁴The synthesized PCA can still be run on any other topology (over the same set of names). However, it is sufficient to know that it is *equivalent to the formula when we run it on a grid (respectively, topology from \mathfrak{T})*.

of “covering” all rings in terms of a *finite* collection of PCA: For those rings with at most $r_\varphi + 2$ processes, we can construct tailor-made PCA, i.e., one PCA for each ring. All other rings can be covered by one single PCA. Using this observation, we can even do better than that and show that one single PCA is enough. The idea is to “approximate” the size of the given ring in terms of the number of active processes and to launch a corresponding PCA according to Corollary 1. As we already know that we will run the PCA on a *ring*, we in fact only have to “determine” its size (see Appendix XV).

Theorem 7. *Let $\varphi \in \text{EMSO}_{\{a,b\}}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$ be a sentence and $B \geq 1$. There is a PCA $\mathcal{A} \in \text{PCA}_{\{a,b\}}$ such that, for all $\mathcal{T} \in \mathfrak{T}_{\text{ring}}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\varphi)$.*

Let us come back to the generic result of Theorem 6. Its proof uses the logical characterization of fixed-topology CA [13]. But it works similarly for $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}, \sim]$ when we take [3] instead of [13]. In the setting of fixed topologies, \sim reduces to a local comparison of event labels so that [3] is indeed applicable. The logic $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}, \sim]$ is a priori weaker than $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$, but allows us to drop the channel restriction.

Theorem 8. *Let $\varphi \in \text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}, \sim]$ be a sentence and $\mathfrak{T} \subseteq \mathbb{T}_{\mathcal{N}}$ be an $(r_\varphi + 2)$ -unambiguous set of topologies. There is a PCA $\mathcal{A} \in \text{PCA}_{\mathcal{N}}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.*

From Theorem 8 and Lemma 1, we can derive statements analogous to Corollary 1 and Theorem 7 (which we omit).

We do not know whether Theorem 8 holds for $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$ (or, equivalently, whether Theorem 6 holds without channel bound). The answer is affirmative if $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}]$ is equivalent to unbounded CA over *fixed* topologies. But this is an open problem.

The translation of PCA to $\text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}]$ is not restricted to topologies of a particular form. The proof of the following theorem is by a standard construction.

Theorem 9. *Let $\mathcal{A} \in \text{PCA}_{\mathcal{N}}$ be a PCA. There is a sentence $\varphi \in \text{EMSO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}]$ such that, for all topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\mathcal{A})$.*

B. Proof Sketch for Main Result (Theorem 6)

The rest of this section is devoted to the proof of Theorem 6 (which works for Theorem 8 with only minor changes).

Set $\sigma^* = \{\triangleleft_{\text{proc}}^*, \triangleleft_{\text{msg}}\}$ and let $\varphi \in \text{EMSO}_{\mathcal{N}}[\sigma^*]$ be a sentence. According to Theorem 3, there are a radius $r = r_\varphi \geq 1$ and $\varphi' = \exists X_1 \dots \exists X_m \exists x_1 \dots \exists x_n \forall y \chi \in \text{EMSO}_{\mathcal{N}}[\sigma^*]$ such that $\chi \in \text{FO}_{\mathcal{N}}[\sigma^*]$ is (r, σ^*) -local around y and, for all $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\varphi) = L_{\mathcal{T}}(\varphi')$. The free variables of $\forall y \chi$ 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.

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 [11, page 806], 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.

For simplicity, we henceforth suppose that y is the only free variable of χ (and write $\chi(y)$). Thus, for the rest of the proof, we fix $r, B \geq 1$, an $(r + 2)$ -unambiguous set $\mathfrak{T} \subseteq \mathbb{T}_{\mathcal{N}}$ of topologies, and a sentence $\forall y \chi(y) \in \text{FO}_{\mathcal{N}}[\sigma^*]$ such that $\chi(y)$ is (r, σ^*) -local around y . We will build a PCA $\mathcal{A} \in \text{PCA}_{\mathcal{N}}$ such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\forall y \chi(y))$. We sketch the construction and try to give some intuition. The formal definition of \mathcal{A} is technical and requires a lot of additional notation. All details can be found in Appendix XIV.

We exploit locality of $\chi(y)$: to know whether $M, e \models \chi(y)$, i.e., MSC M satisfies $\chi(y)$ when y is interpreted as e , it is sufficient to look at the neighborhood of e with radius r (as y is the only free variable, $r - 1$ actually would be enough).

Example 6. Consider the MSC M over $\mathcal{T} = \mathcal{T}_{\text{grid}}^{2,5}$ depicted in Figure 8. Take any event e that is located on process p . All events f such that $\text{dist}_M^{\sigma^*}(e, f) \leq r = 3$ lie on a process in the gray-shaded *topology neighborhood* of p with radius $R = \lceil r/2 \rceil = 2$, which has p as a distinguished center. We call this neighborhood a *sphere* and denote it by $R\text{-Sph}(\mathcal{T}, p)$. One major task of \mathcal{A} is to identify spheres in the topology it is run on. But it has to rely on the messages that are predetermined by M . Therefore, \mathcal{A} can actually only detect a substructure of $R\text{-Sph}(\mathcal{T}, p)$. Figure 9 depicts its restriction $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ (gray-shaded) to those edges that are “covered” by a message of M and to those nodes that one can reach from p with at most R such edges. However, every process preserves its complete type information. Observe that process $(2, 5)$ is not part of $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ anymore. Moreover, the edge between $(1, 3)$ and $(1, 4)$ is removed, since it is not covered by a message. Let $R\text{-Sph}(M, p)$ be the restriction of M to $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ (cf. again Figure 9). We call $R\text{-Sph}(M, p)$ a *partial MSC*, since it has some unmatched events. In fact, satisfaction $M, e \models \chi(y)$ only depends on $R\text{-Sph}(M, p)$, for all events e on p . \square

The example illustrates that, essentially, we have to cope with spheres and partial MSCs, i.e., structures of bounded “width”. For a fixed sphere, every $\text{MSO}_{\mathcal{N}}$ -formula can be translated to a (fixed-topology) CA that accepts exactly the partial MSCs that are a model of the formula [13]. For [13] to apply, we need to restrict to B -bounded MSCs (unless we prove Theorem 8, which is based on [3]). Thus, given a sphere θ , we can construct a CA \mathcal{B}_θ that recognizes the partial MSCs over θ that satisfy $\chi(y)$ for all events e located on the sphere center. Up to isomorphism, there are only finitely many spheres θ of radius R and, thus, finitely many CA \mathcal{B}_θ . To obtain the PCA \mathcal{A} running on MSCs over arbitrarily large topologies, we will *glue* these CA together. Note that, to exploit unambiguity, it is crucial that we restrict to those spheres that arise from topologies in \mathfrak{T} .

We proceed as follows. Every process p guesses a sphere θ , supposing that its topology neighborhood looks like θ , and

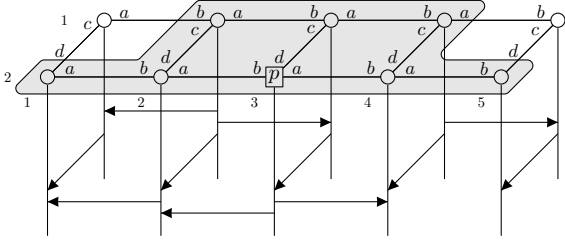


Fig. 8. MSC M over $\mathcal{T}_{\text{grid}}^{2,5}$, and $2\text{-Sph}(\mathcal{T}_{\text{grid}}^{2,5}, p)$

runs a copy of \mathcal{B}_θ to make sure that the partial MSC in its neighborhood is accepted by \mathcal{B}_θ . Whenever p communicates with neighboring processes, the guess is forwarded in terms of messages. Processes receiving the guess also have to simulate \mathcal{B}_θ . Since neighboring processes have to verify their own guess as well, a process will actually have to run several CA simultaneously. The main difficulty, however, is to verify that a guess is correct so that the right CA is applied. The procedure of guessing and forwarding spheres is not able to check by itself whether a cycle in a sphere is correctly simulated in an MSC, and vice versa. It is only correct by the fact that the underlying set \mathfrak{T} of topologies is $(r+2)$ -unambiguous. Indeed, $2R+1 = 2\lceil r/2 \rceil + 1 \leq r+2$ is the maximal length of a cycle through a sphere center that is needed to cover a given edge in the sphere.

Example 7. We resume Example 6 to illustrate the functioning of \mathcal{A} . So, assume $\mathcal{T} = \mathcal{T}_{\text{grid}}^{2,5} \in \mathfrak{T}$, $r = 3$, and $R = \lceil r/2 \rceil = 2$. In an accepting run of \mathcal{A} on the MSC M from Figure 8, process $p = (2, 3)$ will guess the sphere $\theta = R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ illustrated in Figure 9. Accordingly, it will launch the fixed-topology CA \mathcal{B}_θ , which accepts the partial MSC $R\text{-Sph}(M, p)$ (cf. again Figure 9). Actually, \mathcal{B}_θ consists of several local automata $\mathcal{B}_\theta[q]$, one per sphere process q (rather than process type). Thus, p simulates the local automaton $\mathcal{B}_\theta[(2', 3')]$. In doing so, it eventually sends θ to process $(2, 2)$, together with its current position $(2', 3')$ in θ . Receiving the message through interface a , $(2, 2)$ can infer its own position $(2', 2')$ in θ , and so it learns that it has to simulate $\mathcal{B}_\theta[(2', 2')]$. Similarly, $(2, 2)$ has to receive a message over interface d that confirms that $(1, 2)$ has launched $\mathcal{B}_\theta[(1', 2')]$, and so on. There are subtle arguments and technical issues in \mathcal{A} that guarantee that \mathcal{T} and θ simulate each other. We sketch only the direction “ \mathcal{T} simulates θ ”. Let $w = (b, a)(d, c)(a, b)(c, d)$. Our construction makes sure that, starting from p , \mathcal{T} exhibits a w -labeled path. A priori, this does not imply that the path returns to p . But as θ arises from the $(r+2)$ -unambiguous class \mathfrak{T} , $|w| = 4 \leq r+2$, and w forms a cycle in θ , w forms a cycle in \mathcal{T} as well. Recall that every process has to run several CA simultaneously, which we did not take into account in the example. \square

Remark 4. The normal form stated in Theorem 3 can be computed effectively (it builds on Gaifman’s effective normal form). Thus, Theorem 6 is constructive if the spheres $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ with $\mathcal{T} \in \mathfrak{T}$ are effectively representable

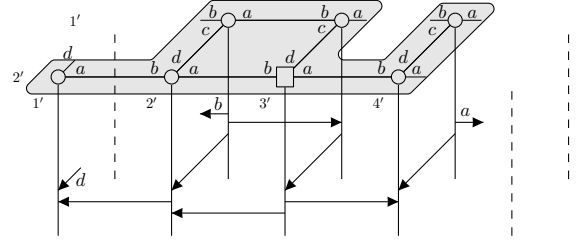


Fig. 9. MSC $2\text{-Sph}(M, p)$ over $2\text{-Sph}(\mathcal{T}_{\text{grid}}^{2,5}, p) \upharpoonright M$

(which holds for all standard classes). It follows from the word case that the PCA cannot be computed in elementary time.

Remark 5. We cannot exploit [17], [20], dealing with universally bounded CA, instead of [13], even if we restrict to universally B -bounded MSCs M (all linearizations are B -bounded): while $R\text{-Sph}(M, p)$ is guaranteed to be (existentially) B -bounded, it is not necessarily universally B -bounded.

VII. EMSO VS. PCA OVER PRIME TOPOLOGIES

Next, we present an orthogonal approach to realizability, where the construction of a PCA is independent of a concrete class of topologies. For this, we will restrict the logic further.

Recall that satisfaction of an $\text{FO}_{\mathcal{N}}[\langle \cdot \rangle_{\text{proc}}, \langle \cdot \rangle_{\text{msg}}]$ -formula essentially depends on the $\{\langle \cdot \rangle_{\text{proc}}, \langle \cdot \rangle_{\text{msg}}\}$ -neighborhoods that occur in an MSC⁵ (Theorem 3). Up to isomorphism, there are only finitely many such neighborhoods, for a fixed radius. We slightly *modify* the construction from [3] (which we had used as a *black-box* for Theorem 8) and define a PCA that, when running on an MSC, outputs the neighborhood of each event. However, this automaton can, a priori, not detect cycles (cf. Theorem 4) and “needs some help” from the underlying topology. It is only guaranteed to compute neighborhoods correctly when it is run on *prime* topologies. “Prime” is in the spirit of “unambiguous”, but on a lower level, since it is tailored to detecting neighborhoods in MSCs rather than in topologies. While “prime” discards all ring topologies and ring forests, it includes all pipelines, trees, and grids. We give more intuition in the proof sketch for Theorem 10 below.

Definition 5. A topology $(P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ is called *prime* if, for all $p \in P$, $w \in (\mathcal{N} \times \mathcal{N})^*$, and $n \geq 1$, we have that $p \dashv^{w^n} p$ implies $p \dashv^w p$.

In other words, a prime topology satisfies the following monotonicity property: If $p \dashv^w q$ with $p \neq q$, then starting from p and “applying” w several times will never lead back to p . Note that “prime” is a property of a *single* topology, while “unambiguous” refers to a *class* of topologies.

Lemma 2. All topologies in $\mathfrak{T}_{\text{lin}}$, $\mathfrak{T}_{\text{tree}}$, and $\mathfrak{T}_{\text{grid}}$ are prime, while none of the topologies in $\mathfrak{T}_{\text{ring}}$ is prime.

We build a PCA that is equivalent to a given formula φ on all prime topologies. Unlike in Theorem 6, it does not depend on a class of topologies, but only on φ and \mathcal{N} .

⁵In a neighborhood, we only keep the process types, not the processes.

Theorem 10. *Let $\varphi \in \text{EMSO}_{\mathcal{N}}[\langle \downarrow_{\text{proc}}, \downarrow_{\text{msg}} \rangle]$ be a sentence. There is a PCA $\mathcal{A} \in \mathbb{PCAN}$ such that, for all prime topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\varphi)$.*

Proof: We sketch the idea, details can be found in Appendix XVI. Thanks to Theorem 3, the problem reduces to constructing a PCA from a formula $\forall y \chi \in \text{FO}_{\mathcal{N}}[\langle \downarrow_{\text{proc}}, \downarrow_{\text{msg}} \rangle]$ where χ is $(r, \{\downarrow_{\text{proc}}, \downarrow_{\text{msg}}\})$ -local around y , for some $r \geq 1$ (cf. proof of Theorem 6). To translate $\forall y \chi$ into a PCA, 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 [3] 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 XVI) so that we obtain the desired sphere automaton. As a last step, the latter is restricted to states that signal that χ holds.

To compute neighborhoods, the PCA has to guarantee that certain sequences w of “directions” form a cycle in an MSC. While there is no direct way to enforce that w forms a cycle, a PCA can make sure that, for all $n \in \mathbb{N}$, an event admits a w^n -labeled path. As MSCs inherit the prime-property from topologies, this implies that w indeed gives rise to a cycle. ■

Remark 6. Using Hanf’s normal form (which we did not do to avoid additional notation), the PCA can be built in elementary time [2]. This is a priori not guaranteed using Theorem 3.

VIII. CONCLUSION

In this paper, we developed a framework for communicating systems with parameterized network topology. In particular, we provided various characterizations of PCA in terms of EMSO logic. Our constructions and proofs are uniform and capture typical cases such as pipelines, trees, grids, and rings.

Theorem 7 reveals that the notion “unambiguous” is not optimal: all $\text{EMSO}_{\{a,b\}}[\langle \downarrow_{\text{proc}}, \downarrow_{\text{msg}} \rangle]$ -formulas are realizable for the class of rings, which is not k -unambiguous for all $k \geq 3$. It seems difficult to characterize exactly those classes for which all formulas from a given logic are realizable, but it will be worthwhile to examine if classes with comparably simple characterizations exist that generalize our results.

Our constructions crucially rely on the bounded-degree property. An obvious question would be a framework including topologies of unbounded degree such as star topologies or, more generally, unranked trees. However, it is not clear how a parameterized automaton should look like in that case. One possibility is to employ registers so that, at any time, a process can remember “some” of its unboundedly many neighbors.

Our framework may carry over to Zielonka’s asynchronous automata [25] with binary actions. These automata have been considered in [12] over tree architectures to get decidability of the controller-synthesis problem. This also raises the question about a parameterized formulation of the control problem.

It is important to study also 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.

Acknowledgment: The author thanks Dietrich Kuske for stimulating discussions, and Nicolas Baudru for the pointer to [7].

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 D. Kuske. An optimal construction of Hanf sentences. *Journal of Applied Logic*, 10(2):179–186, 2012.
- [3] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theor. Comput. Sci.*, 358(2-3):150–172, 2006.
- [4] 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.
- [5] 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.
- [6] J. Büchi. Weak second order logic and finite automata. *Z. Math. Logik, Grundlag. Math.*, 5:66–62, 1960.
- [7] J. Chalopin, S. Das, and A. Kosowski. Constructing a map of an anonymous graph: Applications of universal sequences. In *Proc. of OPODIS’10*, volume 6490 of *LNCS*, pages 119–134. Springer, 2010.
- [8] G. Delzanno, A. Sangnier, and G. Zavattaro. Parameterized verification of ad hoc networks. In *Proc. of CONCUR’10*, volume 6269 of *LNCS*. Springer, 2010.
- [9] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–52, 1961.
- [10] E. A. Emerson and K. S. Namjoshi. On reasoning about rings. *Int. J. Found. Comput. Sci.*, 14(4):527–550, 2003.
- [11] P. Gastin and D. Kuske. Uniform satisfiability problem for local temporal logics over Mazurkiewicz traces. *Inf. Comput.*, 208(7):797–816, 2010.
- [12] 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.
- [13] 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.
- [14] B. Genest, D. Kuske, and A. Muscholl. On communicating automata with bounded channels. *Fundam. Inform.*, 80(1-3):147–167, 2007.
- [15] S. M. German and A. P. Sistla. Reasoning about systems with many processes. *J. ACM*, 39(3):675–735, 1992.
- [16] 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.
- [17] 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.
- [18] S. Jacobs and R. Bloem. Parameterized synthesis. In *Proc. of TACAS’12*, volume 7214 of *LNCS*, pages 362–376. Springer, 2012.
- [19] H. J. Keisler and W. B. Lutfallah. Shrinking games and local formulas. *Ann. Pure Appl. Logic*, 128(1-3):215–225, 2004.
- [20] D. Kuske. Regular sets of infinite message sequence charts. *Inf. Comput.*, 187:80–109, 2003.
- [21] 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.
- [22] 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.
- [23] W. Thomas. Elements of an automata theory over partial orders. In *Proc. of POMIV 1996*, volume 29 of *DIMACS*. AMS, 1996.
- [24] 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.
- [25] W. Zielonka. Notes on finite asynchronous automata. *R.A.I.R.O. — Informatique Théorique et Applications*, 21:99–135, 1987.

APPENDIX

IX. FORMAL DEFINITION OF TOPOLOGY CLASSES

We give the formal definitions of pipelines, trees, grids, and rings. Recall that we do not distinguish isomorphic topologies. Given $n \in \mathbb{N}$, we let in the following $[n] := \{1, \dots, n\}$.

For $n \geq 2$, the pipeline $\mathcal{T}_{\text{lin}}^n$ is defined as the topology $(\{1, \dots, n\}, \dashv)$ over $\{a, b\}$ where

$$\dashv = \begin{aligned} & \{(i, a, b, i+1) \mid i \in [n-1]\} \\ & \cup \{(i+1, b, a, i) \mid i \in [n-1]\}. \end{aligned}$$

A tree topology is a topology (P, \dashv) over $\{a, b, c, d\}$ where P is a prefix-closed subset of $\{0, 1\}^*$ such that $|P| \geq 2$, and

$$\dashv = \begin{aligned} & \{(u, a, b, u0) \mid u, u0 \in P\} \\ & \cup \{(u0, b, a, u) \mid u, u0 \in P\} \\ & \cup \{(u, c, d, u1) \mid u, u1 \in P\} \\ & \cup \{(u1, d, c, u) \mid u, u1 \in P\}. \end{aligned}$$

For $n \geq 3$, the ring $\mathcal{T}_{\text{ring}}^n$ is defined as the topology $(\{1, \dots, n\}, \dashv)$ over $\{a, b\}$ where

$$\dashv = \begin{aligned} & \{(i, a, b, (i+1) \bmod n) \mid i \in [n]\} \\ & \cup \{((i+1) \bmod n, b, a, i) \mid i \in [n]\}. \end{aligned}$$

For $m, n \geq 1$ such that $\max\{m, n\} \geq 2$, we define $\mathcal{T}_{\text{grid}}^{m,n}$ as the topology $([m] \times [n], \dashv)$ over $\{a, b, c, d\}$ where

$$\dashv = \begin{aligned} & \{((i, j), a, b, (i, j+1)) \mid i \in [m] \text{ and } j \in [n-1]\} \\ & \cup \{((i, j+1), b, a, (i, j)) \mid i \in [m] \text{ and } j \in [n-1]\} \\ & \cup \{((i, j), c, d, (i+1, j)) \mid i \in [m-1] \text{ and } j \in [n]\} \\ & \cup \{((i+1, j), d, c, (i, j)) \mid i \in [m-1] \text{ and } j \in [n]\}. \end{aligned}$$

X. PROOF OF THEOREM 1

Theorem 1. *PCA are closed under union and intersection: For all $\mathcal{A}_1, \mathcal{A}_2 \in \text{PCA}_{\mathcal{N}}$, there are PCA \mathcal{A} and \mathcal{B} over \mathcal{N} such that, for all topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$, we have $L_{\mathcal{T}}(\mathcal{A}) = L_{\mathcal{T}}(\mathcal{A}_1) \cup L_{\mathcal{T}}(\mathcal{A}_2)$ and $L_{\mathcal{T}}(\mathcal{B}) = L_{\mathcal{T}}(\mathcal{A}_1) \cap L_{\mathcal{T}}(\mathcal{A}_2)$.*

Proof: The proof is by a standard construction with one subtlety, which is the acceptance condition. Suppose $\mathcal{A}_1 = (S_1, \text{Msg}_1, \Delta_1, I_1, F_1)$ and $\mathcal{A}_2 = (S_2, \text{Msg}_2, \Delta_2, I_2, F_2)$. We assume that $S_1 \cap S_2 = \emptyset$.

For union, $\mathcal{A} = (S, \text{Msg}, \Delta, I, F)$ chooses and simulates non-deterministically \mathcal{A}_1 or \mathcal{A}_2 . The acceptance condition makes sure that this choice is consistent throughout a run. So, we let:

- $S = S_1 \cup S_2$,
- $\text{Msg} = \text{Msg}_1 \cup \text{Msg}_2$,
- $\Delta = \Delta_1 \cup \Delta_2$,
- $I(t) = I_1(t) \cup I_2(t)$ for all $t \in 2^{\mathcal{N}} \setminus \{\emptyset\}$, and
- $F = (F_1 \wedge \bigwedge_{s \in S_2} \neg(\#(s) \geq 1)) \vee (F_2 \wedge \bigwedge_{s \in S_1} \neg(\#(s) \geq 1))$.

For intersection, we use a standard product construction and define $\mathcal{B} = (S, \text{Msg}, \Delta, I, F)$ as follows:

- $S = S_1 \times S_2$,

- $\text{Msg} = \text{Msg}_1 \times \text{Msg}_2$,
- $I(t) = I_1(t) \times I_2(t)$ for all $t \in 2^{\mathcal{N}} \setminus \{\emptyset\}$.

Moreover, Δ is the classical transition product: there is a transition

$$(s_1, s_2) \xrightarrow{!(m_1, m_2)^a} (s'_1, s'_2)$$

whenever $(s_1, !m_1 a, s'_1) \in \Delta_1$ and $(s_2, !m_2 a, s'_2) \in \Delta_2$. Receive transitions are defined analogously. Note that the acceptance condition F is a finite boolean combination of statements $\langle \#((s_1, s_2)) \geq k \rangle$ with $(s_1, s_2) \in S_1 \times S_2$. It is given as $F = \llbracket F_1 \rrbracket_1 \wedge \llbracket F_2 \rrbracket_2$. Here, $\llbracket \cdot \rrbracket_1$ replaces every atomic formula $\langle \#(s_1) \geq k \rangle$ by

$$\bigvee_{m \in \mathbf{M}_k(S_2)} \bigwedge_{s \in S_2} \langle \#((s_1, s_2)) \geq m(s_2) \rangle$$

where $\mathbf{M}_k(S_2)$ is the set of multisets over S_2 of size k , i.e., the set of mappings $m : S_2 \rightarrow \mathbb{N}$ such that $\sum_{s_2 \in S_2} m(s_2) = k$. The transformation $\llbracket \cdot \rrbracket_2$ is defined analogously. ■

XI. PROOF OF THEOREM 4

Theorem 4. *There exists a sentence $\varphi \in \text{FO}_{\{a,b\}}[\prec_{\text{proc}}, \prec_{\text{msg}}]$ such that, for all PCA $\mathcal{A} \in \text{PCA}_{\{a,b\}}$, there is $\mathcal{T} \in \mathfrak{T}_{\text{ring}}^*$ with $L_{\mathcal{T}}^1(\mathcal{A}) \neq L_{\mathcal{T}}^1(\varphi)$.*

Proof: The sentence φ will say that every event is part of the cycle pattern that is depicted in Figure 10:

$$\varphi = \forall x \exists x_1, \dots, x_6 (x \in \{x_1, \dots, x_6\} \wedge \text{cycle}(x_1, \dots, x_6))$$

where $\text{cycle}(x_1, \dots, x_6)$ is defined as

$$\begin{aligned} & x_1 \prec_{\text{msg}} x_2 \prec_{\text{proc}} x_3 \prec_{\text{msg}} x_4 \prec_{\text{proc}} x_5 \prec_{\text{msg}} x_6 \\ & \wedge x_1 \prec_{\text{proc}} x_6 \wedge \bigwedge_{i \in \{1,3,5\}} \text{act}(x_i) = !a \end{aligned}$$

Towards a contradiction, suppose there is a PCA \mathcal{A} such that, for all $\mathcal{T} \in \mathfrak{T}_{\text{ring}}^*$, $L_{\mathcal{T}}(\mathcal{A})$ and $L_{\mathcal{T}}(\varphi)$ agree on all 1-bounded MSCs. Without loss of generality, we assume that send transitions in \mathcal{A} are of the form $(s, !s' a, s')$, i.e., the message coincides with the target state.

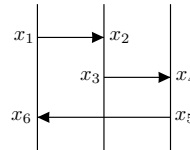
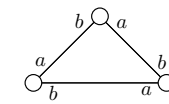


Fig. 10. Cycle

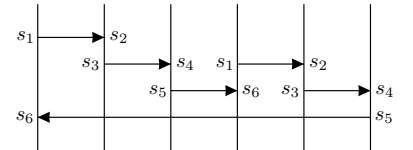
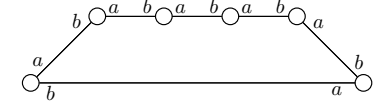


Fig. 11. Fusing two cycles

Consider the MSC M_n that consists of $n \geq 1$ disjoint copies of the “atomic” MSC from Figure 10. That is, M_n is an MSC over the topology $\mathcal{T}_n = \mathcal{T}_{\text{ring}}^3 \uplus \dots \uplus \mathcal{T}_{\text{ring}}^3 \in \mathfrak{T}_{\text{ring}}^*$ with n disjoint copies of $\mathcal{T}_{\text{ring}}^3$. Obviously, $M_n \in L_{\mathcal{T}_n}^1(\varphi) = L_{\mathcal{T}_n}^1(\mathcal{A})$ for all $n \geq 1$. When we choose n large enough, then there is an accepting run ρ of \mathcal{A} on M_n that behaves the same on

at least two disjoint copies of atomic MSCs. More precisely, M_n is an MSC over $\mathcal{T} \uplus \mathcal{T}_{\text{ring}}^3 \uplus \mathcal{T}_{\text{ring}}^3 \in \mathfrak{T}_{\text{ring}}^*$, for some \mathcal{T} , such that ρ assigns states s_1, \dots, s_6 to the events x_1, \dots, x_6 of the atomic MSCs over the last two copies of $\mathcal{T}_{\text{ring}}^3$, respectively. We will now replace these two atomic MSCs with the larger one from Figure 11, over $\mathcal{T}_{\text{ring}}^6$. The resulting MSC, call it M'_n , is a 1-bounded MSC over $\mathcal{T} \uplus \mathcal{T}_{\text{ring}}^6$. There is still a run on M'_n , using the assignment shown in Figure 11. As the multiset of terminal states does not change, the run is accepting. But this is a contradiction, since M'_n does not satisfy φ . ■

XII. PROOF OF THEOREM 5

Recall that we consider tree topologies from $\mathfrak{T}_{\text{tree}}$ over $\{a, b, c, d\}$.

Theorem 5. *There exists a sentence $\varphi \in \text{FO}_{\{a,b,c,d\}}$ such that, for all PCA $\mathcal{A} \in \text{PCA}_{\{a,b,c,d\}}$, there is $\mathcal{T} \in \mathfrak{T}_{\text{tree}}$ with $L_{\mathcal{T}}^1(\mathcal{A}) \neq L_{\mathcal{T}}^1(\varphi)$.*

Proof: A picture over the set $\Omega = \{\circ, \odot, \bullet\}$ of colors is a rectangular matrix with $m \geq 1$ rows and $n \geq 1$ columns, and with entries in Ω . 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 $\eta \in \Omega$. Let $\mathfrak{P}_=$ be the set of those pictures that are the concatenation P_1QP_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 [23], 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 $\mathfrak{T}_{\text{pict}}$ be the set of topologies $\mathcal{T}_{\text{pict}}^n \in \mathfrak{T}_{\text{tree}}$, with $n \geq 1$, as depicted in Figure 12 for $n = 8$. Note that $\mathcal{T}_{\text{pict}}^n$ has $2n + 1$ vertices. As $\mathfrak{T}_{\text{pict}}$ is a subset of $\mathfrak{T}_{\text{tree}}$, 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 12. 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

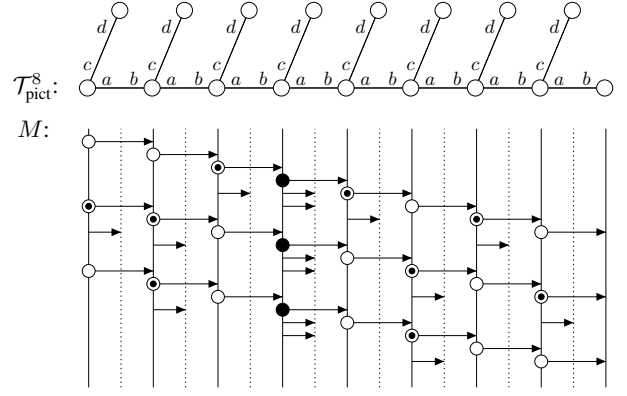


Fig. 12. The communication topology $\mathcal{T}_{\text{pict}}^8$ over $\mathcal{N} = \{a, b, c, d\}$, as well as an MSC that encodes a picture

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 $\text{FO}_{\{a,b,c,d\}}$, i.e., for all $n \geq 1$, $L_{\mathcal{T}_{\text{pict}}^n}(\Psi)$ is the set of MSCs that correspond to pictures with n columns. Let $\mathfrak{L}_=^n$ be the set of MSCs over $\mathcal{T}_{\text{pict}}^n$, that encode a picture from $\mathfrak{P}_=$ with n columns. In $\text{FO}_{\{a,b,c,d\}}$, we can define formulas *is-coordinate*(x) and $\eta(x)$ for all $\eta \in \Omega$ in the obvious way. Moreover, $x \leq_1 y$ (which corresponds to walking down in a picture) is given, for MSCs, by $x \triangleleft_{\text{proc}}^* y$. Finally, $x \leq_2 y$ (which corresponds to walking rightwards in a picture) is given by $x \triangleleft^* y \wedge \forall x'(x \triangleleft_{\text{proc}} x' \rightarrow \neg(x' \triangleleft^* y))$. It follows that there is a sentence $\varphi \in \text{FO}_{\{a,b,c,d\}}$ such that $\mathfrak{L}_=^n = L_{\mathcal{T}_{\text{pict}}^n}(\varphi) = L_{\mathcal{T}_{\text{pict}}^n}^1(\varphi)$, for all $n \geq 1$.

Now suppose that there is a PCA $\mathcal{A} = (S, \text{Msg}, \Delta, I, F)$ such that, for all $\mathcal{T} \in \mathfrak{T}_{\text{pict}}$, we have $L_{\mathcal{T}}^1(\mathcal{A}) = L_{\mathcal{T}}^1(\varphi)$. In any accepting run of \mathcal{A} on an MSC $M \in \mathfrak{L}_=^n$ encoding picture P_1QP_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 $(|S| \times |\text{Msg}|)^m$ such assignments. On the other hand, there are $2^{2^m} - 1$ nonempty sets of words of length m over $\{\circ, \odot\}$, which exceeds $(|S| \times |\text{Msg}|)^m$ for sufficiently large m . Thus, there is an accepting run, possibly wrt. a different topology, on the encoding of $P'_1QP'_2$ for some P'_1, P'_2 that induce distinct sets of column labelings, a contradiction. ■

Note that the proof reveals that Theorem 5 already holds for the logic $\text{FO}_{\{a,b,c,d\}}[\triangleleft^*, \sim]$.

XIII. PROOFS OF LEMMA 1 AND LEMMA 2

Lemma 1. *The classes $\mathfrak{T}_{\text{lin}}$, $\mathfrak{T}_{\text{tree}}$, $\mathfrak{T}_{\text{grid}}$, and $\{\mathcal{T}_{\text{ring}}^n \mid n \geq \max\{3, k + 1\}\}$ are all k -unambiguous, for every $k \in \mathbb{N}$. Moreover, $\{\mathcal{T}\}$ is k -unambiguous for all $\mathcal{T} \in \mathfrak{T}_{\text{ring}}$ and $k \in \mathbb{N}$.*

Proof: Note that the claim for pipelines follows from the result for trees or grids.

Trees: Set $\bar{a} = b$, $\bar{b} = a$, $\bar{c} = d$, and $\bar{d} = c$. We say that a word $w \in (\{a, b, c, d\} \times \{a, b, c, d\})^*$ is *well-formed* if it is generated

by the grammar $A \rightarrow (a, \bar{a})A(\bar{a}, a)A \mid \varepsilon$ where $a \in \{a, b, c, d\}$. We show that, for all $w \in (\{a, b, c, d\} \times \{a, b, c, d\})^*$, all $(P, \dashv) \in \mathfrak{T}_{\text{tree}}$, and all $p, q \in P$ such that $p \xrightarrow{w} q$, we have

$$p = q \iff w \text{ is well-formed.}$$

This will imply the claim from the lemma. We proceed by induction on the structure of w . The claim is immediate for $w = \varepsilon$. So suppose that $w \neq \varepsilon$. First, assume $p = q$. Then, w is of the form $(a, \bar{a})w_1(\bar{a}, a)w_2$ such that $p \xrightarrow{a, \bar{a}} p' \xrightarrow{w_1} p' \xrightarrow{\bar{a}, a} p$, $p \xrightarrow{w_2} p$ for some p' . Using the induction hypothesis, we obtain that w_1 and w_2 are both well-formed. Thus, w is well-formed as well. Conversely, suppose that w is well-formed. Then, it is of the form $(a, \bar{a})w_1(\bar{a}, a)w_2$ such that w_1 and w_2 are well-formed. By the induction hypothesis, we get $p = q$.

Grids: Since the grid $\mathcal{T}_{\text{grid}}^{m,n} \in \mathfrak{T}_{\text{grid}}$ is isomorphic to (P, \dashv) where $P = \{1, \dots, m\} \times \{1, \dots, n\}$ are coordinates in the plane, ‘‘applying’’ pairs $(a, b), (b, a), (c, d), (d, c)$ to a coordinate (i, j) corresponds to adding, component-wise, $(0, 1), (0, -1), (1, 0), (-1, 0)$, respectively. In the expected manner, every word $w \in (\{a, b, c, d\} \times \{a, b, c, d\})^*$ determines a pair $\hat{w} \in \mathbb{Z} \times \mathbb{Z}$. For all $p, q \in P$, we have

$$p \xrightarrow{w} q \implies p + \hat{w} = q.$$

Now, $p \xrightarrow{w} p$ implies $\hat{w} = (0, 0)$, which implies the claim.

Rings: The set $\{\mathcal{T}_{\text{ring}}^n \mid n \geq \max\{3, k+1\}\}$ is k -unambiguous, since a word of length $\leq k$ is not sufficient to circumnavigate a ring with at least $\max\{3, k+1\}$ nodes. Thus, the same argument as for pipelines (grids) applies. Finally, $\{\mathcal{T}\}$ is k -unambiguous for every ring $\mathcal{T} \in \mathfrak{T}_{\text{ring}}$ and $k \in \mathbb{N}$, since every node in \mathcal{T} ‘‘looks the same’’. ■

Lemma 2. *All topologies in $\mathfrak{T}_{\text{lin}}$, $\mathfrak{T}_{\text{tree}}$, and $\mathfrak{T}_{\text{grid}}$ are prime, while none of the topologies in $\mathfrak{T}_{\text{ring}}$ is prime.*

Proof: Again, the claim for pipelines follows from the result for trees or grids.

Trees: The claim follows from the fact that, given a word $w \in (\{a, b, c, d\} \times \{a, b, c, d\})^*$ that is not well-formed (cf. proof of Lemma 1), w^n is not well-formed either, for all $n \geq 1$ (like for words that are not well-bracketed).

Grids: For grids, we also apply the argument from the proof of Lemma 1. Suppose that $p \xrightarrow{w} q$ with $p \neq q$. This implies $\hat{w} \neq (0, 0)$. Since addition is monotonous, we have that $p \xrightarrow{w^n} q'$ implies $p \neq q'$, for all $n \geq 1$.

Rings: Let $\mathcal{T}_{\text{ring}}^n = (P, \dashv) \in \mathfrak{T}_{\text{ring}}$, $p \in P$, and $w = (a, b)^n$. Then, we have $p \xrightarrow{w} p$. Thus, $\mathcal{T}_{\text{ring}}^n$ is not prime. ■

XIV. MISSING DETAILS FOR PROOF OF THEOREM 6

Recall that we fixed $r, B \geq 1$, a $(r+2)$ -unambiguous set $\mathfrak{T} \subseteq \mathbb{T}_{\mathcal{N}}$ of topologies, and a sentence $\forall y \chi(y) \in \text{FO}_{\mathcal{N}}[\sigma^*]$ such that $\chi(y)$ is (r, σ^*) -local around y . Moreover, we set $R = \lceil r/2 \rceil$. We will build a PCA \mathcal{A} such that, for all $\mathcal{T} \in \mathfrak{T}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\forall y \chi(y))$.

A. Communicating Automata over Sphere Topologies

Let $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$. For $p, q \in P$, set $\text{dist}_{\mathcal{T}}(p, q)$ to be the distance (i.e., the minimal length of a path, or ∞ if such a path does not exist) between p and q in \mathcal{T} . By $R\text{-Sph}(\mathcal{T}, p)$, we denote the R -sphere of \mathcal{T} around p , i.e., the substructure of $(P, \dashv, \text{type}_{\mathcal{T}})$ induced by the vertices $q \in P$ such that $\text{dist}_{\mathcal{T}}(p, q) \leq R$, with p as an additional constant called *center*. Note that $R\text{-Sph}(\mathcal{T}, p)$ is always a topology (when we ignore the type mapping and the constant). For example, the shaded area in Figure 8 captures the sphere $2\text{-Sph}(\mathcal{T}_{\text{grid}}^{2,5}, p)$.

Let $R\text{-Spheres}(\mathbb{T}_{\mathcal{N}}) := \{R\text{-Sph}(\mathcal{T}, p) \mid \mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}} \text{ and } p \in P\}$. Any element from $R\text{-Spheres}(\mathbb{T}_{\mathcal{N}})$ is called an (R) -sphere. We (mostly) do not distinguish isomorphic spheres so that the number of R -spheres is finite (thanks to the fact that topologies have bounded degree).

Let $\theta = (\mathcal{U}, \gamma) \in R\text{-Spheres}(\mathbb{T}_{\mathcal{N}})$ be an R -sphere, with $\mathcal{U} = (Q, \sim, \xi)$. Recall that $\xi : Q \rightarrow 2^{\mathcal{N}}$ and $\gamma \in Q$ is the center. Similarly to topologies, we may also write $\text{dist}_{\mathcal{U}}(p, q)$ or $\text{dist}_{\theta}(p, q)$ for $\text{dist}_{(Q, \sim)}(p, q)$. We will define (partial) MSCs over θ . These MSCs are *connected*, are empty or have at least one event on γ , and may have unmatched events, namely those whose communication partners are beyond \mathcal{U} .

Definition 6. *An MSC over θ is a tuple $M = (E, \triangleleft, \ell, \lambda)$ where E is the finite (possibly empty) set of events, $\triangleleft = \triangleleft_{\text{proc}} \uplus \triangleleft_{\text{msg}} \subseteq E \times E$ is the acyclic edge relation, $\ell : E \rightarrow Q$, and $\lambda : E \rightarrow \{!a, ?a \mid a \in \mathcal{N}\}$. For $p \in Q$, let $E_p := \{e \in E \mid \ell(e) = p\}$. We require that the following hold:*

- $\triangleleft_{\text{proc}}$ is a union $\bigcup_{p \in P} \triangleleft_p$ where each $\triangleleft_p \subseteq E_p \times E_p$ is the direct-successor relation of some total order on E_p ,
- there are a partition $E = E_1 \uplus E_2 \uplus E_{\text{unm}}$ and a bijection $\mu : E_1 \rightarrow E_2$ such that $\triangleleft_{\text{msg}} = \{(e, \mu(e)) \mid e \in E_1\}$ (events from E_{unm} are called *unmatched*),
- for all $(e, f) \in \triangleleft_{\text{msg}}$, there are $a, b \in \mathcal{N}$ such that $\ell(e) \stackrel{a}{\sim} \ell(f)$ and $(\lambda(e), \lambda(f)) = (!a, ?b)$,
- for all $(e, f), (e', f') \in \triangleleft_{\text{msg}}$ such that $\ell(e) = \ell(e')$ and $\ell(f) = \ell(f')$, we have $e \triangleleft_{\text{proc}}^* e'$ iff $f \triangleleft_{\text{proc}}^* f'$ (FIFO),
- for all $e \in E_{\text{unm}}$ and $a \in \mathcal{N}$ with $\lambda(e) = \{!a, ?a\}$, we have both $\text{dist}_{\theta}(\gamma, \ell(e)) = R$ and there is no pair $(b, p) \in \mathcal{N} \times Q$ such that $\ell(e) \stackrel{b}{\sim} p$.

Moreover, we require that, if $E \neq \emptyset$, then the graph (E, \triangleleft) is connected and $E_{\gamma} \neq \emptyset$.

Note that this definition actually depends on R , which we had fixed. Again, we do not distinguish isomorphic partial MSCs.

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 2-sphere is depicted in Figure 9.

Definition 7. *Suppose $\theta = (\mathcal{U}, \gamma)$ to be an R -sphere with $\mathcal{U} = (Q, \sim, \xi)$. A communicating automaton (CA) over θ is a tuple $\mathcal{B} = (S, \Delta, \iota, F)$ where*

- S is the finite set of states,
- $\iota : Q \rightarrow S$ associates with each process an initial state,
- $F : Q \rightarrow 2^S$ determines the local final states,
- $\Delta \subseteq S \times \Sigma_{\mathcal{B}} \times S$ 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.

Runs of CA are defined similarly to PCA, but we have to consider unmatched events. Let $M = (E, \triangleleft, \ell, \lambda)$ be an MSC over θ , and let $\rho : E \rightarrow S$ be a mapping. We define $\rho^- : E \rightarrow S$ as follows: For $(f, e) \in \triangleleft_{\text{proc}}$, let $\rho^-(e) = \rho(f)$; for a $\triangleleft_{\text{proc}}$ -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 $(e, f) \in \triangleleft_{\text{msg}}$ and names $a, b \in \mathcal{N}$ with $\ell(e) \stackrel{a}{\sim} \ell(f)$, we have both $(\rho^-(e), !_m a, \rho(e)) \in \Delta$ and $(\rho^-(f), ?_{\rho(e)} b, \rho(f)) \in \Delta$,
- for all $e \in E_{\text{unm}}$ and $a \in \mathcal{N}$ with $\lambda(e) = !a$, we have $(\rho^-(e), !_m a, \rho(e)) \in \Delta$,
- for all $e \in E_{\text{unm}}$ 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 $\triangleleft_{\text{proc}}$ -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 $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ be a topology, $p \in P$, and $M = (E, \triangleleft, \ell)$ be an MSC over \mathcal{T} . Set $H = \{(q, a, b, q') \in \dashv \mid q = \ell(e) \text{ and } q' = \ell(f) \text{ for some } (e, f) \in \triangleleft_{\text{msg}} \cup \triangleleft_{\text{msg}}^{-1}\}$. Let Q be the set of processes $q \in P$ such that p and q are connected in the graph (P, H) by a path using at most R edges. By $R\text{-Sph}(M, p)$ (somewhat abusing notation), we denote the restriction of $(E, \triangleleft, \ell, \text{act}_M)$ to events in $\ell^{-1}(Q)$. We also define $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M := ((Q, H|_Q, \text{type}_{\mathcal{T}|_Q}), p)$ where $H|_Q = \{(q, a, b, q') \in H \mid q, q' \in Q\}$ and $\text{type}_{\mathcal{T}|_Q}$ is the restriction of $\text{type}_{\mathcal{T}}$ to Q . Then, $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ is an R -sphere and $R\text{-Sph}(M, p)$ is an MSC over $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$. For example, consider the MSC M over $\mathcal{T}_{\text{grid}}^{2,5}$ from Figure 8. Figure 9 shows the MSC $2\text{-Sph}(M, p)$ over the sphere $2\text{-Sph}(\mathcal{T}_{\text{grid}}^{2,5}, p) \upharpoonright M$.

The next theorem is due to a result by Genest, Kuske, and Muscholl. It allows us to evaluate $\forall y \chi(y)$ in terms of CA over bounded topologies.

Theorem 10 (cf. [13], Theorem 4.1). *There is a collection $(\mathcal{B}_{\theta})_{\theta \in R\text{-Spheres}(\mathbb{T}_{\mathcal{N}})}$ of CA \mathcal{B}_{θ} over θ such that the following holds, for all topologies $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$, all $p \in P$, and all B -bounded MSCs $M = (E, \triangleleft, \ell)$ over \mathcal{T} :*

$$\begin{aligned} M, e \models \chi(y) \text{ for all } e \in E_p \\ \iff R\text{-Sph}(M, p) \in L(\mathcal{B}_{R\text{-Sph}(\mathcal{T}, p) \upharpoonright M}). \end{aligned}$$

The result from [13] applies, since $R\text{-Sph}(M, p)$ is B -bounded and, given $e \in E_p$, all events $f \in E$ such that $\text{dist}_M^+(e, f) \leq r$ are covered by $R\text{-Sph}(M, p)$ (cf. Figures 7, 8, and 9). Finally, as MSCs over spheres are connected, local final states in CA are enough.

Remark 7. When $\chi(y)$ has more free variables than just y , \mathcal{B}_{θ} also depends on an assignment of truth values to propositions in χ over these variables.

Remark 8. To prove Theorem 8, we use [3] instead of [13]. In fact, Theorem 10 holds verbatim (even without the restriction to B -bounded MSCs) when $\forall y \chi(y) \in \text{FO}_{\mathcal{N}}[\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}, \sim]$. Note that, in the fixed-topology setting, \sim reduces to the comparison of sphere-process labels that are added to events so that [3] can indeed be applied.

B. The Construction of \mathcal{A}

According to Theorem 10, it will be sufficient that each process that is run by the PCA \mathcal{A} identifies a subsphere of its actual topology neighborhood. So, we set $R\text{-Sub}(\mathfrak{T}) := \{R\text{-Sph}(\mathcal{T}, p) \upharpoonright M \mid \mathcal{T} = (P, \dashv) \in \mathfrak{T}, p \in P, \text{ and } M \text{ an MSC over } \mathcal{T}\}$, which is finite up to isomorphism. Note that, for $\mathcal{T} \in \mathfrak{T}$ and a process p of \mathcal{T} , $R\text{-Sub}(\mathfrak{T})$ includes $R\text{-Sph}(\mathcal{T}, p)$ as well as some spheres that consist only of one single node. We fix the finitely many CA $(\mathcal{B}_{\theta})_{\theta \in R\text{-Sub}(\mathfrak{T})}$ according to Theorem 10, where $\mathcal{B}_{\theta} = (S_{\theta}, \Delta_{\theta}, \iota_{\theta}, F_{\theta})$ such that the sets S_{θ} are pairwise disjoint.

We say that $w \in (\mathcal{N} \times \mathcal{N})^*$ is *circular* (wrt. \mathfrak{T}) if there are $\mathcal{T} = (P, \dashv) \in \mathfrak{T}$ and $p \in P$ such that $p \xrightarrow{w} p$.

The PCA $\mathcal{A} = (S, \text{Msg}, \Delta, I, F) \in \text{PCA}_{\mathcal{N}}$ with $S = \text{Msg}$ is defined as follows (\mathcal{U} will always refer to (Q, \sim, ξ) and \mathcal{U}' to (Q', \sim', ξ')):

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, $\alpha \in Q$ 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:

- $\xi(\alpha) = \xi'(\alpha')$,
- if $\gamma = \alpha$ and $\gamma' = \alpha'$, then $\kappa = \kappa'$,
- if $(\mathcal{U}, \gamma, \alpha) = (\mathcal{U}', \gamma', \alpha')$, then $\kappa = \kappa'$, and
- for all $a, b, b' \in \mathcal{N}$, $p \in Q$, and $p' \in Q'$ such that $\alpha \stackrel{a}{\sim} p$ and $\alpha' \stackrel{a'}{\sim} p'$, we have $b = b'$.

Let $t \in S$ and $a, b \in \mathcal{N}$. We let t_{ab} and ${}_{ba}t$ both denote the set of tuples $(\mathcal{U}, \gamma, \alpha, s, H) \in t$ such that there is $p \in Q$ satisfying $\alpha \stackrel{a}{\sim} p$.

We say that (a, b) is *enabled* in t if, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in t$, $q \in Q$, and $w \in (\mathcal{N} \times \mathcal{N})^{\leq 2R}$ such that $w(a, b)$ is circular and $q \stackrel{w}{\sim} \alpha$, we have $\alpha \stackrel{a}{\sim} q$.

Initial and Final States: For $t \in 2^{\mathcal{N}} \setminus \{\emptyset\}$, a state $t \in S$ is contained in $I(t)$ if, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in t$, we have $\xi(\alpha) = t$, $s = \iota_{(\mathcal{U}, \gamma)}(\alpha)$, 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 $\{a \in \mathcal{N} \mid \alpha \stackrel{a}{\sim} p \text{ for some } b \in \mathcal{N} \text{ and } p \in Q\} \subseteq H$. The latter means that H contains

all verification obligations imposed by the guessed topology \mathcal{U} . Then, F is defined as $\bigwedge_{t \in S \setminus G} \neg(\#(t) \geq 1)$.

Send Transitions (ST): The triple $(t^-, !_m a, t) \in S \times \Sigma_{\mathcal{A}} \times S$ is contained in the transition relation Δ if $m = t$, there is $b \in \mathcal{N}$ such that $\underline{t}_{ab} \neq \emptyset$ and (a, b) 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 $\text{dist}_{\mathcal{U}}(\gamma, \alpha) = R$, and
- 4) $H = H^- \cup \{a\}$, which marks interface a as “checked”.

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

- (a) $\Phi(\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-) = (\mathcal{U}, \gamma, \alpha, s, H) =: \kappa$ implies
 1. $(\mathcal{U}^-, \gamma^-, \alpha^-) = (\mathcal{U}, \gamma, \alpha)$,
 2. $(s^-, ?_k b, s) \in \Delta_{(\mathcal{U}, \gamma)}$ for some k such that, if $\kappa \in \underline{ab}t$, then $k = \mu(\kappa)$,
 3. α has a b -successor in \mathcal{U} or $\text{dist}_{\mathcal{U}}(\gamma, \alpha) = R$, and
 4. $H = H^- \cup \{b\}$.
- (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 $\mathcal{U} = (Q, \rightsquigarrow, \xi)$).
- (c) For all $(\mathcal{U}, \gamma, \alpha, s, H) \in (m \setminus m_{\underline{ab}}) \cup (t \setminus \underline{ab}t)$, we have $\text{dist}_{\mathcal{U}}(\gamma, \alpha) = R$.

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

This concludes the construction of the PCA \mathcal{A} . It remains to show its correctness in the sense of the following lemma:

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 $\mathcal{T} = (P, \dashv) \in \mathfrak{T}$ and let $M = (E, \triangleleft, \ell)$ be a B -bounded MSC over \mathcal{T} . For a process $p \in P$, we let $M_p = (Ev_p, \triangleleft_p, \ell_p, \lambda_p) = R\text{-Sph}(M, p)$, and we let $\tau_p = ((Q_p, \rightsquigarrow_p, \xi_p), p)$ denote $R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$. Note that, since M is B -bounded, M_p is B -bounded, too. Finally, let $\mathcal{B}_p = (S_p, \Delta_p, \iota_p, F_p) := \mathcal{B}_{\tau_p}$, which is a CA over τ_p .

“ \Rightarrow ”: Suppose $M \models \forall y \chi(y)$. By Theorem 10, we have $M_p \in L(\mathcal{B}_p)$, for all $p \in P$. Thus, for all $p \in P$, there is an accepting run $\rho_p : Ev_p \rightarrow S_p$ of \mathcal{B}_p on M_p . From the

collection $(\rho_p)_{p \in P}$, we define a mapping $\rho : E \rightarrow S$ by

$$\rho(e) = \{(\tau_p, \ell(e), \rho_p(e), H_e) \mid p \in P \text{ with } e \in Ev_p\} \quad (1)$$

where $H_e = \{a \in \mathcal{N} \mid \text{there is } f \in E \text{ such that } f \triangleleft_{\text{proc}}^* e \text{ and } \text{act}_M(f) \in \{!a, ?a\}\}$. Towards an appropriate initial state of \mathcal{A} for M , we let $\zeta = (\zeta_q)_{q \in P_M}$ where

$$\zeta_q = \{(\tau_p, q, \iota_p(q), \emptyset) \mid p \in P \text{ such that } q \in Q_p\}.$$

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

Clearly, $\zeta_p \in S$, for all $p \in P_M$. So, let us prove that $\rho(e) \in S$ for all $e \in E$. Suppose $\kappa = (\mathcal{U} = (Q, \rightsquigarrow, \xi), \gamma, \alpha, s, H) \in \rho(e)$ and $\kappa' = (\mathcal{U}' = (Q', \rightsquigarrow', \xi'), \gamma', \alpha', s', H') \in \rho(e)$.

- (a) By (1), we have $\xi(\alpha) = \xi'(\alpha') = \ell(e)$.
- (b) Assume that we have both $\gamma = \alpha$ and $\gamma' = \alpha'$. By (1), we have $\kappa = (\tau_{\ell(e)}, \ell(e), \rho_{\ell(e)}(e), H_e)$ 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, $\ell(e) \rightsquigarrow \gamma$ iff $\ell(e) \rightsquigarrow' \gamma'$ for all $w \in (\mathcal{N} \times \mathcal{N})^*$, which is impossible if $\gamma \neq \gamma'$). We deduce $\kappa = \kappa'$.
- (d) Suppose $a, b, b' \in \mathcal{N}$, $p \in Q$, and $p' \in Q'$ such that $\ell(e) \overset{a}{\rightsquigarrow} b$ p and $\ell(e) \overset{a'}{\rightsquigarrow} b'$ p' . Similarly to (c), we obtain $p = p'$, which implies $b = b'$.

Next, we show that ρ is a run of \mathcal{A} on M . Let $e \in E$, $a, b \in \mathcal{N}$, and $p' \in P$ such that $\text{act}_M(e) \in \{!a, ?a\}$ and $\ell(e) \overset{a}{\rightsquigarrow} b$ p' . Clearly, we have $\rho(e)_{\underline{ab}} \neq \emptyset$. Let us show that, moreover, (a, b) is enabled in $\rho(e)$. Let $p \in P$ such that $e \in Ev_p$. Moreover, let $q \in Q_p$ and $w \in (\mathcal{N} \times \mathcal{N})^{\leq 2R}$ such that $w(a, b)$ is circular and $q \rightsquigarrow_p^w \ell(e)$. Since, $\tau_p = R\text{-Sph}(\mathcal{T}, p) \upharpoonright M$ and $\mathcal{T} \in \mathfrak{T}$ with \mathfrak{T} being $(r+2)$ -unambiguous, we have $\ell(e) \overset{a}{\rightsquigarrow} b$ q . Thus, (a, b) is enabled in $\rho(e)$.

Let $(e, f) \in \triangleleft_{\text{msg}}$ and $a, b \in \mathcal{N}$ such that $\ell(e) \overset{a}{\rightsquigarrow} b$ $\ell(f)$. We show that $\rho_{\zeta}^-(e) \xrightarrow{\rho(e)a} \rho(e)$ and $\rho_{\zeta}^-(f) \xrightarrow{\rho(e)b} \rho(f)$. We define a bijection $\Phi_e : \rho_{\zeta}^-(e) \rightarrow \rho(e)$ as follows:

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

$$\rho_{\zeta}^-(e) = \{(\tau_p, \ell(e), \iota_p(\ell(e)), \emptyset) \mid p \in P \text{ such that } e \in Ev_p\}.$$

We set $\Phi_e(\tau_p, \ell(e), \iota_p(\ell(e)), \emptyset) = (\tau_p, \ell(e), \rho_p(e), H_e)$.

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

$$\rho_{\zeta}^-(e) = \{(\tau_p, \ell(e^-), \rho_p(e^-), H_{e^-}) \mid p \in P \text{ such that } e \in Ev_p\}.$$

We set $\Phi_e(\tau_p, \ell(e^-), \rho_p(e^-), H_{e^-}) = (\tau_p, \ell(e), \rho_p(e), H_e)$.

Note that $|\rho_{\zeta}^-(e)| = |\rho(e)| = |\{p \in P \mid e \in Ev_p\}|$. 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 $e^- \triangleleft_{\text{proc}} e$ and $\Phi_e(\tau_p, \ell(e^-), \rho_p(e^-), H_{e^-}) = (\tau_p, \ell(e), \rho_p(e), H_e)$. We show (ST).

- 1) We have $(\tau_p, \ell(e^-)) = (\tau_p, \ell(e))$.
- 2) As ρ_p is a run, $(\rho_p(e^-), !_p(e^-)a, \rho_p(e)) \in \Delta_p$.

- 3) We have $e \in Ev_p$. By the definition of $M_p = R\text{-Sph}(M, p)$, we have that $\ell(e)$ has an a -successor in τ_p , or $dist_{\tau_p}(p, \ell(e)) = R$.
- 4) Clearly, $H_e = H_{e^-} \cup \{a\}$.

We define the bijection $\Phi_f : \rho_{\bar{\zeta}}(f) \rightarrow \rho(f)$ and verify (RTa) accordingly. It remains to define a bijection $\hat{\Phi}_{(e,f)} : \rho(e)_{\underline{ab}} \rightarrow \underline{ab}\rho(f)$ as well as a mapping $\mu : \underline{ab}\rho(f) \rightarrow \bigcup_{p \in P} S_p$.

We have

$$\rho(e)_{\underline{ab}} = \{(\tau_p, \ell(e), \rho_p(e), H_e) \mid p \in P \text{ such that } e \in Ev_p \text{ and } \ell(e) \stackrel{a}{\rightsquigarrow}_p q \text{ for some } q \in Q_p\},$$

$$\underline{ab}\rho(f) = \{(\tau_p, \ell(f), \rho_p(f), H_f) \mid p \in P \text{ such that } f \in Ev_p \text{ and } q \stackrel{b}{\rightsquigarrow}_p \ell(f) \text{ for some } q \in Q_p\}.$$

Note that $\rho(e)_{\underline{ab}}$ and $\underline{ab}\rho(f)$ are both nonempty. For $p \in P$ with $e \in Ev_p$ and $\ell(e) \stackrel{a}{\rightsquigarrow}_p q$ for some $q \in Q_p$, we have $f \in Ev_p$. This follows from the definition of M_p . We set $\hat{\Phi}_{(e,f)}(\tau_p, \ell(e), \rho_p(e), H_e) = (\tau_p, \ell(f), \rho_p(f), H_f)$. Note that $\hat{\Phi}_{(e,f)}$ is bijective. Similarly, for $p \in P$ with $f \in Ev_p$ and $q \stackrel{b}{\rightsquigarrow}_p \ell(f)$ for some $q \in Q_p$, we have $e \in Ev_p$. In that case, we set $\mu(\tau_p, \ell(f), \rho_p(f), H_f) = \rho_p(e)$. With this definition, (RTb) is directly verified. In (RTc), we have to show that, for all $(\mathcal{U}, \gamma, \alpha, s, H) \in (\rho(e) \setminus \rho(e)_{\underline{ab}}) \cup (\rho(f) \setminus \underline{ab}\rho(f))$, we have $dist_{\mathcal{U}}(\gamma, \alpha) = R$. So, consider $(\tau_p, \ell(e), \rho_p(e), H_e)$ with $p \in P$ such that $e \in Ev_p$ and $\ell(e)$ does not have an a -successor in τ_p . Since, then, e is an unmatched event of M_p , we have $dist_{\tau_p}(p, \ell(e)) = R$ by the definition of MSC M_p . The reasoning for $(\mathcal{U}, \gamma, \alpha, s, H) \in \rho(f) \setminus \underline{ab}\rho(f)$ is analogous.

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

So, let $e \in E$ be $\triangleleft_{\text{proc}}$ -maximal. Suppose $p \in P$ such that $e \in Ev_p$. We have to show that $\rho_p(e) \in F_p(\ell(e))$ and $\{a \in \mathcal{N} \mid \ell(e) \stackrel{a}{\rightsquigarrow}_p q \text{ for some } b \in \mathcal{N} \text{ and } q \in Q_p\} \subseteq H_e$, i.e., for all $a \in \mathcal{N}$ such that $\ell(e)$ has an outgoing a -edge in τ_p , there is $f \in \mathcal{A}_{\text{proc}}^* e$ such that $act_M(f) \in \{!a, ?a\}$. The former holds since ρ_p is an accepting run. The latter holds since, by definition, the MSC M_p “covers” the a -labeled edge.

“ \Leftarrow ”: Now, suppose $M \in L_{\mathcal{T}}(\mathcal{A})$. There is an accepting run $\rho : E \rightarrow S$ of \mathcal{A} on M , say, with initial state $\zeta = (\zeta)_{p \in P_M}$ for M . In particular, for all $(e, f) \in \triangleleft_{\text{msg}}$ and $a, b \in \mathcal{N}$ such that $\ell(e) \stackrel{a}{\rightsquigarrow} \ell(f)$, we have $\rho_{\bar{\zeta}}(e) \stackrel{!_{\rho(e)}a}{\rightsquigarrow} \rho(e)$ and $\rho_{\bar{\zeta}}(f) \stackrel{?_{\rho(e)}b}{\rightsquigarrow} \rho(f)$.

We will show $M \in L_{\mathcal{T}}(\forall y \chi(y))$. By Theorem 10, it is sufficient to prove $M_{p_0} \in L(\mathcal{B}_{p_0})$ for all $p_0 \in P_M$, i.e., to determine accepting runs $\rho_{p_0} : Ev_{p_0} \rightarrow S_{p_0}$ of \mathcal{B}_{p_0} on M_{p_0} . Let $e, f \in E$ such that $(e, f) \in \triangleleft_{\text{msg}}$. Suppose $a, b \in \mathcal{N}$ such that $\ell(e) \stackrel{a}{\rightsquigarrow} \ell(f)$. According to (ST) and (RT), consider the unique mappings $\Phi_e : \rho_{\bar{\zeta}}(e) \rightarrow \rho(e)$, $\Phi_f : \rho_{\bar{\zeta}}(f) \rightarrow \rho(f)$, $\mu_f : \underline{ab}\rho(f) \rightarrow \bigcup_{\theta \in R\text{-Sub}(\mathcal{T})} S_{\theta}$, and $\hat{\Phi}_{(e,f)} : \rho(e)_{\underline{ab}} \rightarrow \underline{ab}\rho(f)$.

Pick $p_0 \in P_M$. For all events $e \in E$ of M located on p_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 $\rho_{p_0}(e) = s$. Note that, by (ST) and (RT), $\theta_{p_0} := (\mathcal{U}, \gamma)$ is invariant along all events on p_0 .

We claim

$$\tau_{p_0} \cong \theta_{p_0}. \quad (2)$$

Recall that, hereby, $\tau_{p_0} = R\text{-Sph}(\mathcal{T}, p_0) \upharpoonright M_{p_0}$. In particular, (2) implies that $\rho_{p_0}(e) \in S_{p_0}$ for all events e on p_0 . Before we prove (2), we define ρ_{p_0} for all other events of M_{p_0} . In doing so, whenever ρ_{p_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_{p_0} for all events of M_{p_0} that are located on processes p with $0 \leq dist_{\tau_{p_0}}(p_0, p) \leq k < R$. Consider an event $f \in Ev_{p_0}$ that is located on some p with $dist_{\tau_{p_0}}(p_0, p) = k + 1$. There is $e \in Ev_{p_0}$ located on a node with distance k to p_0 (i.e., $\rho_{p_0}(e)$ is already defined) such that e and f form a message.

- Suppose that we have $(e, f) \in \triangleleft_{\text{msg}}$ where $\ell(e) \stackrel{a}{\rightsquigarrow} \ell(f)$, i.e., e sends a message via interface a . Assume that $\kappa_e = (\mathcal{U} = (Q, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$ (by induction, this will mean $\rho_{p_0}(e) = s$). Due to (2), we have $\kappa_e \in \rho_{p_0}(e)_{\underline{ab}}$. By, (RTb), there is $\alpha' \in Q$ such that $\alpha \stackrel{a}{\rightsquigarrow} \alpha'$ and $\hat{\Phi}_{(e,f)}(\kappa_e) = (\mathcal{U}, \gamma, \alpha', s', H')$ for some s', H' . We set $\rho_{p_0}(f) = s'$.
- Suppose that we have $(f, e) \in \triangleleft_{\text{msg}}$ where $\ell(f) \stackrel{a}{\rightsquigarrow} \ell(e)$, i.e., e receives via interface b . Assume that $\kappa_e = (\mathcal{U} = (Q, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$, i.e., $\rho_{p_0}(e) = s$. Due to (2), we have $\kappa_e \in \underline{ab}\rho_{p_0}(e)$. Again, due to (RTb), there is $\alpha' \in Q$ such that $\alpha' \stackrel{a}{\rightsquigarrow} \alpha$, and we have $\hat{\Phi}_{(f,e)}^{-1}(\kappa_e) = (\mathcal{U}, \gamma, \alpha', s', H')$ for some s', H' . We set $\rho_{p_0}(f) = s'$.

Note that ρ_{p_0} is well defined. So let us show that ρ_{p_0} is a run of \mathcal{B}_{p_0} on M_{p_0} . Pick $e \in Ev_{p_0}$.

- Suppose e is of type $!a$. We have $\rho_{\bar{\zeta}}(e) \stackrel{!_{\rho(e)}a}{\rightsquigarrow} \rho(e)$. Assume first that e is process-minimal on some process p . Suppose $\kappa_e = (\mathcal{U}, \gamma, \alpha, s, H)$. Recall that $\rho_{p_0}(e) = s$. Thanks to (ST), $\rho_{\bar{\zeta}}(e)$ contains $(\mathcal{U}, \gamma, \alpha, s^-, \emptyset)$, with $s^- = \iota_{p_0}(p)$ and $(s^-, !_s a, s) \in \Delta_{p_0}$. Moreover, $H = \{!a\}$. If e is not process-minimal, then e has a process-predecessor e^- . By (ST), κ_{e^-} is of the form $(\mathcal{U}, \gamma, \alpha, \rho_{p_0}(e^-), H^-)$, for some H^- , and we have $(\rho_{p_0}(e^-), !_s a, s) \in \Delta_{p_0}$ and $H = H^- \cup \{!a\}$.
- Suppose e executes action $?b$ and let $e^-, f \in Ev_{p_0}$ and $a \in \mathcal{N}$ be such that $e^- \triangleleft_{\text{proc}} e$, $f \triangleleft_{\text{msg}} e$, and $\ell(f) \stackrel{a}{\rightsquigarrow} \ell(e)$. Suppose $\kappa_e = (\mathcal{U} = (Q, \rightsquigarrow, \xi), \gamma, \alpha, s, H)$, $\kappa_{e^-} = (\mathcal{U}^-, \gamma^-, \alpha^-, s^-, H^-)$, and $\kappa_f = (\mathcal{U}', \gamma', \alpha', s', H')$. By (RT), $(\mathcal{U}^-, \gamma^-, \alpha^-) = (\mathcal{U}, \gamma, \alpha)$, $(\mathcal{U}', \gamma') = (\mathcal{U}, \gamma)$, and $\alpha' \stackrel{a}{\rightsquigarrow} \alpha$. Finally, $(s^-, ?_s b, s) \in \Delta_{p_0}$. The cases where e does not have a process-predecessor or where e is unmatched are similar.

Next, we show that ρ_{p_0} is accepting. Let $e \in Ev_{p_0}$ be process-maximal and suppose $\kappa_e = (\mathcal{U}, \gamma, \alpha, s, H)$. From the fact that ρ is accepting, we can deduce that $s \in F_{p_0}(\alpha)$.

To finish the proof, it remains to show (2), i.e., $\tau_{p_0} \cong \theta_{p_0}$. This is done using the H -component of a state as well as the fact that \mathfrak{T} is $(r+2)$ -unambiguous.

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

Suppose $\tau_{p_0} = (\mathcal{W}, \equiv, \pi, p_0)$ and $\theta_{p_0} = (\mathcal{U}, \rightsquigarrow, \xi, q_0)$. For $d \in \{0, \dots, R\}$, let $((W_d, \equiv_d, \pi_d), p_0)$ be the restriction of τ_{p_0} to elements of distance at most d from p_0 in \mathcal{W} . Similarly, let $((Q_d, \rightsquigarrow_d, \xi_d), q_0)$ be the restriction of θ_{p_0} to elements of distance at most d from q_0 in \mathcal{U} .

The following claim implies (2):

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

$$h_d : ((W_d, \equiv_d, \pi_d), p_0) \rightarrow ((Q_d, \rightsquigarrow_d, \xi_d), q_0)$$

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

$$(\mathcal{U}, q_0, h_d(p)) \in \rho(p).$$

Proof: The claim holds for $d = 0$, since $(\mathcal{U}, q_0, q_0) \in \rho(p_0)$.

Now, suppose the claim holds for some $d < R$ so that we have an isomorphism

$$h_d : ((W_d, \equiv_d, \pi_d), p_0) \rightarrow ((Q_d, \rightsquigarrow_d, \xi_d), q_0).$$

Towards h_{d+1} , we extend the domain of h_d to elements $p' \in W_{d+1} \setminus W_d$. So let $p, p' \in W$ such that $dist_{\mathcal{W}}(p_0, p) = d$, $dist(p_0, p') = d+1$, and p and p' are connected by an edge in τ_{p_0} . Moreover, set $q = h_d(p)$.

Suppose $p \xrightarrow{a,b} p'$ with $a, b \in \mathcal{N}$. Since $dist_{\mathcal{W}}(p_0, p) < R$, we also have, by induction hypothesis, $dist_{\mathcal{U}}(q_0, q) < R$. Suppose $(e, e') \in \triangleleft_{msg} \cup \triangleleft_{msg}^{-1}$ such that e is located on p and e' is located on p' . By induction hypothesis, we have $(\mathcal{U}, \gamma, q, s, H) \in \rho(e)$ for some s, H . By (ST) and (RT), there is $q' \in Q$ such that $(\mathcal{U}, \gamma, q', s, H) \in \rho(e')$ and $q \xrightarrow{a,b} q'$. Note that, since h_d is an isomorphism, $q' \in Q_{d+1} \setminus Q_d$. Set $h_{d+1}(p') = q'$. This is well-defined and does not depend on the concrete choice of p or a : if we obtained another, distinct element q'' , we would have $(\mathcal{U}, \gamma, q'', s, H) \in \rho(e')$, 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 $a, b \in \mathcal{N}$, and let $q, q' \in Q$ with distance d and $d+1$, respectively, from q_0 such that $q \xrightarrow{a,b} q'$. Let $p = h_d^{-1}(q)$. By induction hypothesis, we have $(\mathcal{U}, q_0, q) \in \rho(p)$. As $dist_{\mathcal{U}}(q_0, q) < R$ and ρ is an accepting run, there is $p' \in W$ satisfying $p \xrightarrow{a,b} p'$ and $(\mathcal{U}, q_0, q') \in \rho(p')$. Thus, $h_{d+1}(p') = q'$ so that h_{d+1} is surjective.

For $w \in (\mathcal{N} \times \mathcal{N})^*$, let $\overleftarrow{w} \in (\mathcal{N} \times \mathcal{N})^*$ denote its reverse, which is defined inductively by $\overleftarrow{\varepsilon} = \varepsilon$ and $\overleftarrow{w(a,b)} = (b,a)\overleftarrow{w}$.

Now, let us show that h_{d+1} is indeed an isomorphism. Take $p_1, p_2 \in W_d$ and $p'_1, p'_2 \in W_{d+1} \setminus W_d$ as well as $w_1, w_2 \in (\mathcal{N} \times \mathcal{N})^d$ and $a_1, b_1, a_2, b_2 \in \mathcal{N}$ such that

- $p_0 \xrightarrow{w_1} p_1 \xrightarrow{a_1, b_1} p'_1$ and
- $p_0 \xrightarrow{w_2} p_2 \xrightarrow{a_2, b_2} p'_2$.

For $i = 1, 2$, let $q_i = h_{d+1}(p_i)$ and $q'_i = h_{d+1}(p'_i)$. We will show that

- $p'_1 = p'_2$ iff $q'_1 = q'_2$ (so that h_{d+1} is injective), and
- $p'_1 \xrightarrow{a,b} p'_2$ iff $q'_1 \xrightarrow{a,b} q'_2$, for all $a, b \in \mathcal{N}$.

First, suppose $p'_1 = p'_2$. Let $w = (b_1, a_1)\overleftarrow{w_1}w_2(a_2, b_2)$. Then, $p'_1 \xrightarrow{w} p'_1$. As \mathfrak{T} is $(r+2)$ -unambiguous, $\tau_{p_0}, \theta_{p_0} \in R\text{-Sub}(\mathfrak{T})$, $|w| \leq 2R = 2\lceil r/2 \rceil \leq r+2$, and $q'_1 \xrightarrow{w} q'_2$, this implies $q'_1 = q'_2$. The same argument applies when we start with $q'_1 = q'_2$.

Next, suppose $p'_1 \xrightarrow{a,b} p'_2$, for some $a, b \in \mathcal{N}$. Since \equiv is the relation belonging to τ_{p_0} , a message is exchanged between two events located on p'_1 and p'_2 , respectively. It follows that there is $e \in E_{p'_1}$ such that (a, b) is enabled in $\rho(e)$. Let $w = (b_2, a_2)\overleftarrow{w_2}w_1(a_1, b_1)$. Then, $|w| \leq 2R$ and $w(a, b)$ is circular. As (a, b) is enabled in $\rho(e)$, $(\mathcal{U}, q_0, q'_1) \in \rho(p'_1)$, and $q'_2 \xrightarrow{w} q'_1$, we have $q'_1 \xrightarrow{a,b} q'_2$.

Finally, suppose $q'_1 \xrightarrow{a,b} q'_2$, for some $a, b \in \mathcal{N}$. Let $w = (b_2, a_2)\overleftarrow{w_2}w_1(a_1, b_1)$. As ρ is an accepting run, a message has to be exchanged between p'_1 and a process p such that $p'_1 \xrightarrow{a,b} p$. We have $q'_2 \xrightarrow{w} q'_1 \xrightarrow{a,b} q'_2$ and $p'_2 \xrightarrow{w} p'_1 \xrightarrow{a,b} p$. As \mathfrak{T} is $(r+2)$ -unambiguous and $|w(a, b)| \leq 2R+1 = 2\lceil r/2 \rceil + 1 \leq r+2$, we have $p = p'_2$. Thus, $p'_1 \xrightarrow{a,b} p'_2$.

This concludes the proof of Claim 1. \blacksquare

XV. PROOF OF THEOREM 7

Theorem 7. *Let $\varphi \in \text{EMSO}_{\{a,b\}}[\triangleleft_{proc}^*, \triangleleft_{msg}]$ be a sentence and $B \geq 1$. There is a PCA $\mathcal{A} \in \mathbb{P}\mathbb{C}\mathbb{A}_{\{a,b\}}$ such that, for all $\mathcal{T} \in \mathfrak{T}_{ring}$, we have $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\varphi)$.*

The rest of the section is devoted to the proof of the theorem. Intuitively, \mathcal{A} will “approximate” the size of the ring on which it is run, and launch a corresponding automaton. Set $n = r_{\varphi} + 2$. According to Corollary 1, let \mathcal{A}_{n+1} be the PCA for the class of rings, of size at least $n+1$. Moreover, for $k \in \{3, \dots, n\}$, let \mathcal{A}_k be the PCA for the single ring topology \mathcal{T}_{ring}^k . Now, \mathcal{A} is the union of several product PCA $\mathcal{A}_k \times \mathcal{B}_k$ (for the intersection of \mathcal{A}_k and \mathcal{B}_k ; cf. Theorem 1), with k ranging over $\{3, \dots, n+1\}$. Here, \mathcal{B}_k imposes a condition on the number of active processes of an MSC. First, \mathcal{B}_3 accepts those MSCs that have ≤ 3 active processes. For $k \in \{4, \dots, n\}$, \mathcal{B}_k checks that an MSC has exactly k active processes. Finally, \mathcal{B}_{n+1} requires that there are at least $n+1$ active processes.

The construction of \mathcal{A} is straightforward. To prove its correctness, we exploit that, for $k < l$, PCA \mathcal{A}_k and \mathcal{A}_l behave equivalently on MSCs M over \mathcal{T}_{ring}^l with k active processes. This is why we can apply \mathcal{A}_k to M even though the underlying ring has size larger than k .

For an MSC M (over some topology) and $k \in \{3, \dots, n+1\}$, we write $M \approx k$ if the following hold:

- M has ≤ 3 active processes if $k = 3$,
- M has exactly k active processes if $k \in \{4, \dots, n\}$, and
- M has $\geq k$ active processes if $k = n+1$.

The following observation is crucial:

Lemma 4. Let $l \geq 3$ and $k \in \{3, \dots, n+1\}$. Moreover, let \mathcal{T} be the ring topology of size l and M be an MSC over \mathcal{T} such that $M \approx k$. Then, $M \in L_{\mathcal{T}}^B(\mathcal{A}_k)$ iff $M \in L_{\mathcal{T}}^B(\mathcal{A}_{\min\{l, n+1\}})$.

Proof: If $k = l$, then there is nothing to show (the case $k = n+1 < l$ is also immediate). Suppose $k < l$. Since $M = (E, \triangleleft, \ell)$ is an MSC over \mathcal{T} such that $M \approx k$, M is *acyclic* meaning that there is an active process p of M that does not communicate through one of its interfaces:

- for all $e \in E_p$, $act_M(e) \notin \{!a, ?a\}$, or
- for all $e \in E_p$, $act_M(e) \notin \{!b, ?b\}$.

But then, according to the construction from Appendix XIV, every accepting run of \mathcal{A}_k on M is also an accepting run of $\mathcal{A}_{\min\{l, n+1\}}$ on M , and vice versa. An alternative argument is that the truth value of a formula/automaton does not change when we add or remove non-active processes in a topology. Thus, we have $M \in L_{\mathcal{T}}^B(\mathcal{A}_k)$ iff $M \in L_{\mathcal{T}}^B(\mathcal{A}_{\min\{l, n+1\}})$ ■

We will now argue that \mathcal{A} is correct. Let $l \geq 3$ and let \mathcal{T} be the ring of size l . We have to show that $L_{\mathcal{T}}^B(\mathcal{A}) = L_{\mathcal{T}}^B(\varphi)$.

We first consider the inclusion $L_{\mathcal{T}}^B(\mathcal{A}) \subseteq L_{\mathcal{T}}^B(\varphi)$. Let $M \in L_{\mathcal{T}}^B(\mathcal{A})$. There is $k \in \{3, \dots, n+1\}$ such that $M \in L_{\mathcal{T}}^B(\mathcal{A}_k \times \mathcal{B}_k)$. As $M \in L_{\mathcal{T}}^B(\mathcal{B}_k)$, we have $M \approx k$. Moreover, $M \in L_{\mathcal{T}}^B(\mathcal{A}_k)$. By Lemma 4, this implies $M \in L_{\mathcal{T}}^B(\mathcal{A}_{\min\{l, n+1\}})$. The latter equals $L_{\mathcal{T}}^B(\varphi)$. We deduce that $M \in L_{\mathcal{T}}^B(\varphi)$.

Now, we consider the inclusion $L_{\mathcal{T}}^B(\mathcal{A}) \supseteq L_{\mathcal{T}}^B(\varphi)$. Let $M \in L_{\mathcal{T}}^B(\varphi)$, and let $k \in \{3, \dots, n+1\}$ such that $M \approx k$. Note that k is uniquely determined. Clearly, we have $M \in L_{\mathcal{T}}^B(\mathcal{B}_k)$. Moreover, $M \in L_{\mathcal{T}}^B(\mathcal{A}_{\min\{l, n+1\}})$. By Lemma 4, this implies $M \in L_{\mathcal{T}}^B(\mathcal{A}_k)$. We deduce $M \in L_{\mathcal{T}}^B(\mathcal{A}_k \times \mathcal{B}_k)$, which finally implies $M \in L_{\mathcal{T}}^B(\mathcal{A})$.

XVI. MISSING DETAILS FOR PROOF OF THEOREM 10

MSCs over Prime Topologies: Recall that the construction of a PCA from a formula is based on the sphere automaton. For the construction from [3] (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.⁶ Consider the set $\mathbb{D} = \{\text{proc}, \text{proc}^{-1}\} \cup \{\text{msg}_{(a,b)}, \text{msg}_{(a,b)}^{-1} \mid a, b \in \mathcal{N}\}$ of *directions*. Let $M = (E, \triangleleft, \ell)$ be an MSC, over some topology. Every direction $\delta \in \mathbb{D}$ defines a binary relation $\llbracket \delta \rrbracket_M \subseteq E \times E$ as follows:

- $\llbracket \text{proc} \rrbracket_M = \triangleleft_{\text{proc}}$,
- $\llbracket \text{proc}^{-1} \rrbracket_M = \triangleleft_{\text{proc}}^{-1}$,
- $\llbracket \text{msg}_{(a,b)} \rrbracket_M = \{(e, f) \in \triangleleft_{\text{msg}} \mid \ell(e) \xrightarrow{a} b, \ell(f)\}$, and
- $\llbracket \text{msg}_{(a,b)}^{-1} \rrbracket_M = \{(f, e) \in \triangleleft_{\text{msg}}^{-1} \mid \ell(e) \xrightarrow{b} a, \ell(f)\}$.

This is extended to strings $w = \delta_1 \dots \delta_n \in \mathbb{D}^*$: we let $(e, f) \in \llbracket w \rrbracket_M$ if $(e, f) \in \text{id}_E \circ \llbracket \delta_1 \rrbracket_M \circ \dots \circ \llbracket \delta_n \rrbracket_M$ (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 $(e, e) \in \llbracket w \rrbracket_M$ implies

⁶The property is exploited in [3, 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.

$(e, e) \in \llbracket w \rrbracket_M$. Note that MSCs are in general not prime, i.e., for arbitrary topologies. However, we can show the following:

Lemma 5. Let $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ be a prime topology and $M = (E, \triangleleft, \ell)$ 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 $(e, e) \in \llbracket w \rrbracket_M$. We build the “projection” $\langle w \rangle \in (\mathcal{N} \times \mathcal{N})^*$ of w to the alphabet $\mathcal{N} \times \mathcal{N}$ so that it can be applied to the topology \mathcal{T} . It is defined by $\langle \text{proc} \rangle = \langle \text{proc}^{-1} \rangle = \varepsilon$ and $\langle \text{msg}_{(a,b)} \rangle = \langle \text{msg}_{(a,b)}^{-1} \rangle = (a, b)$.

From $(e, e) \in \llbracket w \rrbracket_M$, 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 $(e, e) \notin \llbracket w \rrbracket_M$ (which implies $w \neq \varepsilon$ and $n > 1$). Consider the unique event $e_1 \in E$ such that $(e, e_1) \in \llbracket w \rrbracket_M$. Due to $\ell(e) \xrightarrow{\langle w \rangle} \ell(e)$, we have either $e \triangleleft_{\text{proc}}^+ e_1$ or $e_1 \triangleleft_{\text{proc}}^+ e$. Suppose $e \triangleleft_{\text{proc}}^+ e_1$. The other case is analogous. As $(e, e) \in \llbracket w \rrbracket_M$, there are $e_2, \dots, e_n \in E$ such that $(e, e_1), (e_1, e_2), \dots, (e_{n-1}, e_n) \in \llbracket w \rrbracket_M$. Thus, $(e, e_n) \in \llbracket w \rrbracket_M$. As MSCs obey a FIFO policy, we moreover have $e \triangleleft_{\text{proc}}^+ e_1 \triangleleft_{\text{proc}}^+ e_2 \triangleleft_{\text{proc}}^+ \dots \triangleleft_{\text{proc}}^+ e_n$ and, therefore, $e \neq e_n$. But this contradicts $(e, e) \in \llbracket w \rrbracket_M$. We conclude that M is prime. ■

Sphere Automaton: Next, we adapt the construction of the sphere automaton from [3] 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. Set $\sigma = \{\triangleleft_{\text{proc}}, \triangleleft_{\text{msg}}\}$. Let $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ be a topology, $M = (E, \triangleleft, \ell)$ 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 $(E, \triangleleft, \ell, act_M)$ to events of distance at most r from e . More precisely, it is defined as the structure $\mathcal{M} = (E', \triangleleft', \pi, \lambda, e)$ where $E' = \{f \in E \mid \text{dist}_{\mathcal{M}}^{\sigma}(e, f) \leq r\}$, \triangleleft' is given by $\triangleleft'_{\text{proc}} = \triangleleft_{\text{proc}} \cap (E' \times E')$ and $\triangleleft'_{\text{msg}} = \triangleleft_{\text{msg}} \cap (E' \times E')$, and π and λ are mappings with domain E' : for all $e \in E'$, $\pi(e) = \text{type}_{\mathcal{T}}(\ell(e))$ and $\lambda(e) = act_M(e)$. Note that \mathcal{M} does not depend on a topology anymore, as an event is mapped by π to a process type rather than a process.

Let $\mathbb{S}_r := \{r\text{-Sph}(M, e) \mid M \text{ is an MSC over some topology } \mathcal{T} \in \mathbb{T}_{\mathcal{N}} \text{ and } e \text{ is an event of } M\}$ 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 VI. However, the crucial difference is that the sphere automaton is supposed to detect spheres, while the automaton from Section VI evaluates 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 6. Let $r \in \mathbb{N}$ be some radius. There are a PCA $\mathcal{A} = (S, \text{Msg}, \Delta, I, F) \in \text{PCA}_{\mathcal{N}}$ and a mapping $\nu : S \rightarrow \mathbb{S}_r$ such that the following hold, for all prime topologies $\mathcal{T} \in \mathbb{T}_{\mathcal{N}}$

and all MSCs $M = (E, \triangleleft, \ell)$ 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 $\mathcal{A} = (S, \text{Msg}, \Delta, I, F)$ with $S = \text{Msg}$ as follows:

States: A state $t \in S$ is either \emptyset_ω for some $\omega \in 2^{\mathcal{N}} \setminus \{\emptyset\}$ (the empty set with some annotated type; we set $\text{type}(\emptyset_\omega) = \omega$ and let $\nu(t)$ be an arbitrary sphere), or a nonempty set of tuples $(\mathcal{M}, \gamma, \alpha, \text{col})$ where $(\mathcal{M} = (E, \triangleleft, \pi, \lambda), \gamma) \in \mathbb{S}_r$, $\alpha \in E$ is the *active event*, and $\text{col} \in \{1, \dots, 4 \cdot \text{max}E^2 + 1\}$ is a color with $\text{max}E$ 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 $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$ such that $\gamma = \alpha$ (in that case, we set $\nu(t) = (\mathcal{M}, \gamma)$),
- for all tuples $((E, \triangleleft, \pi, \lambda), \gamma, \alpha, \text{col}) \in t$ and $((E', \triangleleft', \pi', \lambda'), \gamma', \alpha', \text{col}') \in t$, we have $\pi(\alpha) = \pi'(\alpha')$ and $\lambda(\alpha) = \lambda'(\alpha')$ (we set $\text{type}(t) = \pi(\alpha)$ and $\text{label}(t) = \lambda(\alpha)$), and
- if $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$ and $(\mathcal{M}, \gamma, \alpha', \text{col}) \in t$, then $\alpha = \alpha'$.

Initial and Final States: For all $\omega \in 2^{\mathcal{N}} \setminus \{\emptyset\}$, we let $I(\omega) = \{\emptyset_\omega\}$. Let G be the set of nonempty states $t \in S$ such that there is $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$ with α not $\triangleleft_{\text{proc}}$ -maximal in \mathcal{M} . We set $F = \bigwedge_{t \in G} \neg(\#(t) \geq 1)$.

Send Transitions: In the following, we let \mathcal{M} refer to $(E, \triangleleft, \pi, \lambda)$. 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 $\text{label}(t) = !a$,
- (2) $\text{type}(t^-) = \text{type}(t)$,
- (3) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$ and $e \in E$, we have $e \triangleleft_{\text{proc}} \alpha$ iff $(\mathcal{M}, \gamma, e, \text{col}) \in t^-$,
- (4) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t^-$ and $e \in E$, we have $\alpha \triangleleft_{\text{proc}} e$ iff $(\mathcal{M}, \gamma, e, \text{col}) \in t$,
- (5) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$, if α is $\triangleleft_{\text{proc}}$ -minimal in \mathcal{M} and $t^- \neq \emptyset$, then $\text{dist}_{\mathcal{M}}^\sigma(\gamma, \alpha) = r$, and
- (6) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t^-$, if α is $\triangleleft_{\text{proc}}$ -maximal in \mathcal{M} , then $\text{dist}_{\mathcal{M}}^\sigma(\gamma, \alpha) = r$.

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) $\text{label}(t) = ?a$,
- (8) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in t$ and $e \in E$, we have $e \triangleleft_{\text{msg}} \alpha$ iff $(\mathcal{M}, \gamma, e, \text{col}) \in m$,
- (9) for all $(\mathcal{M}, \gamma, \alpha, \text{col}) \in m$ and $e \in E$, we have $\alpha \triangleleft_{\text{msg}} e$ iff $(\mathcal{M}, \gamma, e, \text{col}) \in t$.

This concludes the construction of \mathcal{A} . The correctness proof in the sense of Lemma 6 follows the same lines as that of [3].

To show that cycles in spheres are correctly simulated by a given input MSC M , we use the fact that M is prime. Let $\mathcal{T} = (P, \dashv) \in \mathbb{T}_{\mathcal{N}}$ be a prime topology, $M = (E, \triangleleft, \ell)$ be an MSC over \mathcal{T} (i.e., according to Lemma 5, 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 $\nu(\rho(e_0)) = (\mathcal{M} = (E', \triangleleft', \pi', \lambda'), \gamma)$. We show that, then, $(\gamma, \gamma) \in \llbracket w \rrbracket_{\mathcal{M}}$ (with the obvious meaning) implies $(e_0, e_0) \in \llbracket w \rrbracket_M$.

So, suppose $(\gamma, \gamma) \in \llbracket w \rrbracket_{\mathcal{M}}$. By the construction of \mathcal{A} , we have that, for all $n \geq 1$, there is $e_n \in E$ such that $(e_0, e_n) \in \llbracket w^n \rrbracket_M$ (cf. proof of [3, Claim 4.11]). In particular, $(e_n, e_{n+1}) \in \llbracket w \rrbracket_M$ 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 $\delta \in \mathbb{D}$ such that $f_1 \neq f_2$, $(f_1, f) \in \llbracket \delta \rrbracket_M$, and $(f_2, f) \in \llbracket \delta \rrbracket_M$, which is a contradiction, as f can have at most one δ -predecessor. Continuing this scheme, we get $e_n \notin \{e_0, \dots, e_{n-1}\}$ for all $n \geq 1$. But this is a contradiction to the fact that E is finite. We deduce $(e_0, e_0) \in \llbracket w \rrbracket_M$.