Message-Passing Automata are expressively equivalent to EMSO Logic

Benedikt Bollig *,[1]

Lehrstuhl für Informatik II
RWTH Aachen, Germany

Martin Leucker

Institut für Informatik
TU München, Germany

Abstract

We study the expressiveness of finite message-passing automata with a priori unbounded FIFO channels and show them to capture exactly the class of MSC languages that are definable in existential monadic second-order logic interpreted over MSCs. Furthermore, we prove the monadic quantifier-alternation hierarchy over MSCs to be infinite and conclude that the class of MSC languages accepted by message-passing automata is not closed under complement.

Key words: concurrency, message-passing automata, message sequence charts, monadic second-order logic

1 Introduction

A common design practice when developing communicating systems is to start with drawing scenarios showing the intended interaction of the system to be. The standardized notion of *message sequence charts* (MSCs, [16]) is widely used in industry to formalize such typical behaviors.

* Corresponding author
Email addresses: `bollig@informatik.rwth-aachen.de` (Benedikt Bollig), `leucker@in.tum.de` (Martin Leucker).

[1] Part of this work was done while the author was on leave at the School of Computer Science, University of Birmingham, United Kingdom, and supported by the German Academic Exchange Service (DAAD).

An MSC depicts a single partially-ordered execution sequence of a system. It defines a set of processes interacting with one another by communication actions. In the visual representation of an MSC, processes are drawn as vertical lines that are interpreted as time axes, while a labeled arrow from one line to a second corresponds to the communication events of sending and receiving a message. Collections of MSCs are then used to capture the scenarios that a designer might want the system to follow or to avoid. In this respect, several specification formalisms have been considered, such as *high-level MSCs* or *MSC graphs* [3,26]. The next step in the design process usually is to derive an implementation of the system to develop [11], preferably automatically. In other words, we are interested in generating a distributed automaton *realizing* the behavior given in form of scenarios. This problem asks for the study of automata models that are suited for accepting the system behavior described by MSC specifications.

A common model that reflects the partially-ordered execution behavior of MSCs in a natural manner are *message-passing automata*, MPAs for short. They consist of several components that communicate using reliable FIFO channels. Several variants of MPAs have been studied in the literature: automata with a single or multiple initial states, with finitely or infinitely many states, bounded or unbounded channels, and systems with a global or local acceptance condition.

In this paper, we will focus on MPAs with a priori unbounded FIFO channels and a global acceptance condition where each component employs a finite state space. Thus, our model subsumes the one studied in [11] where a local acceptance condition is used. It coincides with the one used in [15,17], although these papers characterize the fragment of channel-bounded automata. It extends the setting of [1,24] in so far as we provide synchronization messages and a global acceptance condition to have the possibility to coordinate rather autonomous processes. Altogether, our version covers most existing models of communicating automata for MSCs.

A fruitful way to study properties of automata is to establish logical characterizations. For example, finite word automata are known to be expressively equivalent to monadic second-order (MSO) logic over words [6,8]. More precisely, the set of words satisfying some MSO formula can be defined by a finite automaton and vice versa. Those results then initiated the study of automata models for generalized structures such as graphs or, more specifically, labeled partial orders and their relation to MSO logic has been a research area of great interest aiming at a deeper understanding of their logical and algorithmic properties (see [29,7] for overviews).

In this paper, we show that MPAs accept exactly those MSC languages that are definable within the existential fragment of MSO, abbreviated by EMSO.

We recall that emptiness for MPAs is undecidable and conclude that so is satisfiability for EMSO logic. Furthermore, we show that MSO is strictly more expressive than EMSO. More specifically, the monadic quantifier-alternation hierarchy turns out to be infinite. Thus, MPAs do *not* necessarily accept a set of MSCs defined by an MSO formula. We use this result to conclude that the class of MSC languages that corresponds to MPAs is not closed under complementation, answering the question posed in [17].

Previous work deals with MPAs and sets of MSCs that make use only of a bounded part of the actually unbounded channel [15,17,9]. When restricting to sets of *bounded* MSCs (no matter if universally- or existentially-bounded), MSO corresponds to the class of MPAs and is as expressive as its existential fragment [13,17,10]. However, an algebraic or logical characterization of the whole class of MPAs has been unknown.

The next two sections introduce some basic notions and recall the definitions of MSCs and (existential) MSO logic, respectively. Section 4 deals with MPAs and their expressive equivalence to EMSO logic, while Section 5 studies the gap between MSO formulas and their existential fragment.

Acknowledgment We would like to thank Dietrich Kuske for valuable remarks on a previous version of this paper.

2 Preliminaries

Let us first recall some basic definitions and notions. A *partially ordered set* (also called *poset* for short) is a pair $(E, \leq)$ such that E is a nonempty finite set and $\leq$ is a binary relation on E that is reflexive, transitive, and antisymmetric. In this context, the relation $\leq$ is called a *partial order*. A *totally ordered set* is a poset $(E, \leq)$ such that, for any $e, e' \in E$, $e \leq e'$ or $e' \leq e$. Accordingly, we then call the relation $\leq$ a *total order*. Let $\mathcal{P} = (E, \leq)$ be a poset. By $<$, we denote $\leq \setminus \{(e, e) \mid e \in E\}$. Moreover, for $e, e' \in E$, let us write $e \lessdot e'$ if both $e < e'$ and, for any $e'' \in E$, $e < e'' \leq e'$ implies $e'' = e'$. Then, $(E, \lessdot)$ and $\lessdot$ are called the *Hasse diagram* of $\mathcal{P}$ and, respectively, the *covering relation* of $\leq$. For $e \in E$, we furthermore say that e is *minimal/maximal* in $\mathcal{P}$ (we may also say minimal/maximal in $(E, <)$) if there is no $e' \in E$ such that $e' < e/e < e'$, respectively.

Directed acyclic labeled graphs can be seen as the most general structure we consider in this paper. Message sequence charts can be embedded into acyclic graphs or at least have a corresponding one-to-one graph representation.

Let in the following Σ and C be *alphabets*, i.e., nonempty finite sets, which contain the elements the components of a graph are labeled with.

Definition 2.1 ((Directed) Graph)
A *(directed) graph* over (Σ, C) is a structure $G = (E, \{\lhd_c\}_{c \in C}, \lambda)$ where E is its nonempty finite set of *nodes*, the $\lhd_c \subseteq E \times E$ are disjoint binary relations on E, and $\lambda : E \to \Sigma$ is a *(node-)labeling function.*

In the sequel, we call $\lhd := \bigcup_{c \in C} \lhd_c$ the *edge relation* or the set of *edges* of G. Moreover, we sometimes write $\leq_c$ for $(\lhd_c)^*$, abbreviate $(\lhd_c)^+$ by $<_c$, set $\leq$ to be the relation $\lhd^*$, and abbreviate $\lhd^+$ by $<$. The *cardinality* of G, denoted by $|G|$, is actually meant to be the cardinality of E. Moreover, for a subset Σ' of Σ, we set $|G|_{\Sigma'}$ to be $|\lambda^{-1}(\Sigma')|$. For $a \in \Sigma$, we then abbreviate $|G|_{\{a\}}$ by $|G|_a$.

Graphs will primarily serve as a convenient representation of partial orders, which, in turn, are a general model for the behavior of a distributed system. Thus, we assume in the sequel a graph $(E, \{\lhd_c\}_{c \in C}, \lambda)$ to generate a partial order, which means that $(E, \lhd^*)$ is supposed to be a poset. We furthermore require $\lhd$ to be irreflexive. The set of all those *acyclic* graphs is denoted by $\mathbb{DG}(\Sigma, C)$. A useful subclass of $\mathbb{DG}(\Sigma, C)$, denoted by $\mathbb{DG}_H(\Sigma, C)$, is the set of graphs $(E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ such that $\lhd = \lessdot$, i.e., $(E, \lhd)$ is the Hasse diagram of some poset. Throughout the paper, the nodes of a graph are called *events* executing *actions*, which are given by their node labeling.

It may be the case that the set of node labelings or the set of edge labelings is a singleton so that we do not need to explicitly refer to Σ and C, respectively. In that case, we speak of graphs over $(\Sigma, -)$ or over $(-, C)$ and, for example, write $\mathbb{DG}(\Sigma, -)$. Moreover, if the labeling alphabets are clear from the context, we often omit the reference to Σ and C completely.

An important concept of partially ordered sets and their associated graphs is their characterization in terms of *linear extensions* or *linearizations*, which establishes a relationship between posets (or their associated graphs) and words. So let $G = (E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ be a graph. A graph $w = (E', \lhd', \lambda') \in \mathbb{DG}(\Sigma, -)$ is called a *linearization* of G if $E' = E$, $\lhd'$ is the covering relation of some total order containing $\lhd^*$, and $\lambda' = \lambda$. Thus, w can be considered to be a word from Σ^*. The set of linearizations of G is denoted by $Lin(G)$. This notion is extended to sets L of graphs according to $Lin(L) := \bigcup_{G \in L} Lin(G)$.

For $G = (E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$, a nonempty subset Σ' of Σ with $\lambda^{-1}(\Sigma') \neq \emptyset$, and $c \in C$, we denote by $G \upharpoonright (\Sigma', \{c\})$ (we may write $G \upharpoonright \Sigma'$ if C is a singleton) the *projection* $(E', \lhd'_c, \lambda') \in \mathbb{DG}(\Sigma', \{c\})$ of G onto Σ' and c where $E' = \lambda^{-1}(\Sigma')$, $\lhd'_c$ is the union of $\lhd_c \cap (E' \times E')$ and the covering relation of the partial order $(\lhd_c)^* \cap (E' \times E')$, and $\lambda' = \lambda_{|E'}$, i.e., λ' is the restriction of λ to E'. For $e \in E$, let furthermore $G \Downarrow e$ stand for the *downwards closure* of G wrt. e, i.e., for $(E', \{\lhd'_c\}_{c\in C}, \lambda') \in \mathbb{DG}(\Sigma, C)$ where $E' = \{e' \in E \mid e' \lhd^* e\}$, $\lhd'_c = \lhd_c \cap (E' \times E')$, and $\lambda' = \lambda_{|E'}$.

Let B be a natural. For $G = (E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$, we say that the degree of G is *bounded* by B if, for any $e \in E$, $|\{e' \in E \mid e \lhd e' \text{ or } e' \lhd e\}| \leq B$. Given $\mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$, the *degree* of $\mathcal{K}$ is said to be *bounded* by B if, for any $G \in \mathcal{K}$, the degree of G is bounded by B. We say that $\mathcal{K}$ has *bounded degree* if its degree is bounded by some B.

Let Q be a nonempty and finite set. A (Q-)*extended graph* over (Σ, C) is a graph $(E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{DG}(\Sigma \times Q, C)$, i.e, λ is a mapping $E \to \Sigma \times Q$. Note that λ can be seen as a pair (λ', ρ) of mappings $E \to \Sigma$ and $E \to Q$, respectively. Given a class $\mathcal{K}$ of graphs over (Σ, C), the corresponding set of Q-extended graphs over (Σ, C) is denoted by $\mathcal{K}^Q$.

2.2 Monadic Second-Order Logic over Graphs

Throughout the paper, we fix supplies $\mathrm{Var} = \{x, y, \ldots, x_1, x_2, \ldots\}$ of *individual variables* and $\mathrm{VAR} = \{X, Y, \ldots, X_1, X_2, \ldots\}$ of *set variables*.

Definition 2.2 (Monadic Second-Order Logic over Graphs)
Formulas from $\mathrm{MSO}(\Sigma, C)$, the set of *monadic second-order formulas* over the class $\mathbb{DG}(\Sigma, C)$, are built up from the atomic formulas $\lambda(x) = a$ (for $a \in \Sigma$), $x \lhd_c y$ (for $c \in C$), $x \in X$, and $x = y$ (where $x, y \in \mathrm{Var}$ and $X \in \mathrm{VAR}$) and, furthermore, allow the boolean connectives $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$ and the quantifiers $\exists$, $\forall$, which can be applied to either kind of variable.

Let $G = (E, \{\lhd_c\}_{c\in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ be a graph. Given an interpretation function $\mathcal{I}$, which assigns to an individual variable x an event $\mathcal{I}(x) \in E$ and to a set variable X a set of events $\mathcal{I}(X) \subseteq E$, the satisfaction relation $G \models_{\mathcal{I}} \varphi$ for a formula $\varphi \in \mathrm{MSO}(\Sigma, C)$ is given by $G \models_{\mathcal{I}} \lambda(x) = a$ if $\lambda(\mathcal{I}(x)) = a$, $G \models_{\mathcal{I}} x \lhd_c y$ if $\mathcal{I}(x) \lhd_c \mathcal{I}(y)$, $G \models_{\mathcal{I}} x \in X$ if $\mathcal{I}(x) \in \mathcal{I}(X)$, and $G \models_{\mathcal{I}} x = y$ if $\mathcal{I}(x) = \mathcal{I}(y)$, while the remaining operators are defined as usual. If we consider *sentences*, i.e., formulas without free variables, we replace $\models_{\mathcal{I}}$ with $\models$.

For an $\mathrm{MSO}(\Sigma, C)$-formula φ, the notation $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ shall indicate that at most $x_1, \ldots, x_m, X_1, \ldots, X_n$ occur free in φ. The fragment of $\mathrm{MSO}(\Sigma, C)$ that does not make use of any set-variable quantifier is the set of *first-order formulas* over $\mathbb{DG}(\Sigma, C)$ and denoted by $\mathrm{FO}(\Sigma, C)$. An $\mathrm{MSO}(\Sigma, C)$-formula is called *existential* if it is of the form $\exists X_1 \ldots \exists X_n \varphi(X_1, \ldots, X_n, \overline{Y})$ where $\overline{Y}$ is a block of second-order variables and $\varphi(X_1, \ldots, X_n, \overline{Y}) \in \mathrm{FO}(\Sigma, C)$. Let $\mathrm{EMSO}(\Sigma, C)$ denote the class of existential $\mathrm{MSO}(\Sigma, C)$-formulas. In general, we would like to distinguish formulas by their quantifier-alternation depth. So $\Sigma_k(\Sigma, C)$ ($k \geq 1$) shall contain the $\mathrm{MSO}(\Sigma, C)$-formulas of the form $\exists \overline{X_1} \forall \overline{X_2} \ldots \exists/\forall \overline{X_k} \varphi(\overline{X_1}, \ldots, \overline{X_k}, \overline{Y})$ with first-order kernel $\varphi(\overline{X_1}, \ldots, \overline{X_k}, \overline{Y})$ ($\overline{X_i}$ and $\overline{Y}$ are blocks of second-order variables). Note that $\Sigma_1(\Sigma, C)$ and $\mathrm{EMSO}(\Sigma, C)$ coincide. Let us furthermore introduce a variant of $\mathrm{MSO}(\Sigma, C)$: choosing our atomic entities to be $\lambda(x) = a$ (for $a \in \Sigma$), $x \leq y$, $x \in X$, and $x = y$ yields the logics $\mathrm{MSO}(\Sigma, C)[\leq]$, $\mathrm{EMSO}(\Sigma, C)[\leq]$, and $\Sigma_k(\Sigma, C)[\leq]$. The semantics of $x \leq y$ wrt. a graph $G = (E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ and an interpretation function $\mathcal{I}$ is determined by $G \models_{\mathcal{I}} x \leq y$ if $\mathcal{I}(x) \lhd^* \mathcal{I}(y)$.

Let $\mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$. For an $\mathrm{MSO}(\Sigma, C)$-sentence φ, the *language of* φ relative to $\mathcal{K}$, denoted by $L_{\mathcal{K}}(\varphi)$, is the set of graphs $G \in \mathcal{K}$ with $G \models \varphi$. However, as a formula $\varphi(X_1, \ldots, X_n) \in \mathrm{MSO}(\Sigma, C)$ (with free variables) can be considered to define a language of graphs whose labelings are enriched by tuples from $\{0,1\}^n$, we may accordingly denote the corresponding language of φ relative to $\mathcal{K}$ by $L_{\mathcal{K}}(\varphi)$, too, which is then a subset of $\mathcal{K}^{\{0,1\}^n}$. More precisely, an extended graph $G = (E, \{\lhd_c\}_{c \in C}, (\lambda, \rho)) \in \mathcal{K}^{\{0,1\}^n}$ *satisfies* φ if we have $(E, \{\lhd_c\}_{c \in C}, \lambda) \models_{\mathcal{I}_G} \varphi$ where, for any $e \in E$, $e \in \mathcal{I}_G(X_i)$ if $\rho(e)[i] = 1$ (where $\rho(e)[i]$ yields the i-th component of $\rho(e)$).

For $\mathfrak{F} \subseteq \mathrm{MSO}(\Sigma, C)$ and sets $L, \mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$, L is called $\mathfrak{F}_{\mathcal{K}}$-*definable* if $L = L_{\mathcal{K}}(\varphi)$ for some sentence $\varphi \in \mathfrak{F}$. Moreover, the language classes of $\mathrm{MSO}(\Sigma, C)_{\mathcal{K}}$-, $\mathrm{EMSO}(\Sigma, C)_{\mathcal{K}}$-, and $\Sigma_k(\Sigma, C)_{\mathcal{K}}$-definable sets are denoted by $\mathcal{MSO}(\Sigma, C)_{\mathcal{K}}$, $\mathcal{EMSO}(\Sigma, C)_{\mathcal{K}}$, and $\mathcal{L}_{\mathcal{K}}(\Sigma_k(\Sigma, C))$, respectively. Similarly, wrt. the alternative predicate symbol $\leq$, we obtain further classes of graph languages, for example $\mathcal{MSO}(\Sigma, C)[\leq]_{\mathcal{K}}$ and $\mathcal{EMSO}(\Sigma, C)[\leq]_{\mathcal{K}}$.

For $\mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$, we say that the *monadic quantifier-alternation hierarchy* over $\mathcal{K}$ is infinite if the sets $\mathcal{L}_{\mathcal{K}}(\Sigma_k(\Sigma, C))$, $k = 1, 2, \ldots$, form an infinite strict hierarchy. Recall that, in general, the classes of $\Sigma_k(\Sigma, C)_{\mathbb{DG}(\Sigma, C)}$-definable languages form an infinite hierarchy [23,22].

2.3 Graph Acceptors

Besides formulas, graphs themselves may provide a framework to specify graph properties. For instance, we might be interested in the set of those graphs in which a given pattern occurs at least, say, $n \in \mathbb{N}$ times. A pattern H hereby specifies the local neighborhood around a distinguished center γ where the size of the neighborhood is constituted by a natural $R \in \mathbb{N}$, the *radius* of H,

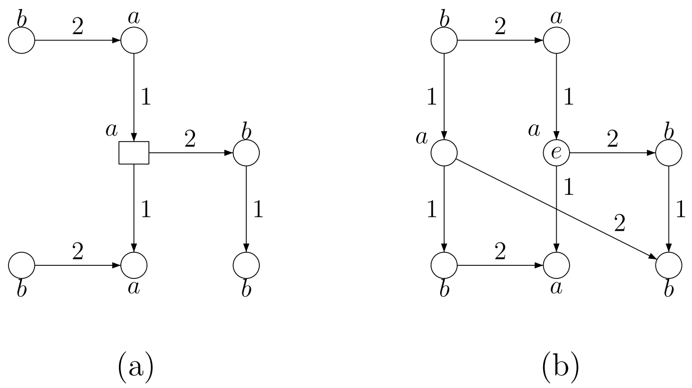


Fig. 1. A 2-sphere over $(\{a, b\}, \{1, 2\})$

which restricts the distance of any node of H to γ.

Let us make this idea more precise and let R be a natural. Given a graph $G = (E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ and nodes $e, e' \in E$, the *distance* $d_G(e', e)$ from e' to e in G is ∞ if it holds $(e, e') \notin (\lhd \cup \lhd^{-1})^*$ and, otherwise, the minimal natural number k such that there is a sequence of elements $e_0, \ldots, e_k \in E$ with $e_0 = e$, $e_k = e'$, and $e_i \lhd e_{i+1}$ or $e_{i+1} \lhd e_i$ for each $i \in \{0, \ldots, k-1\}$. Sometimes, if it is clear from the context, we omit the subscript G just writing $d(e', e)$. An *R-sphere* over (Σ, C) is a graph $H = (E, \{\lhd_c\}_{c \in C}, \lambda, \gamma)$ over (Σ, C) together with a designated *sphere center* $\gamma \in E$ such that, for any $e \in E$, $d_H(e, \gamma) \leq R$ (in abuse of notation, the distance from one node to another will be given wrt. a sphere as well). For a graph $G = (E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ and $e \in E$, let the *R-sphere of G around e*, denoted by $R\text{-Sph}(G, e)$, be given by $(E', \{\lhd'_c\}_{b \in C}, \lambda', e)$ where $E' = \{e' \in E \mid d_G(e', e) \leq R\}$, $\lhd'_c = \lhd_c \cap (E' \times E')$ for each $c \in C$, and λ' is the restriction of λ to E'. A 2-sphere over $(\{a, b\}, \{1, 2\})$ is shown in Figure 1 (a) where the sphere center is depicted as a rectangle. It precisely deals with the 2-sphere of the graph aside around e.

Graph acceptors [27,29] are a generalization of finite automata to graphs. They are known to be expressively equivalent to EMSO logic wrt. graphs of bounded degree. A graph acceptor works on a graph as follows: it first assigns to each node one of its control states and then checks if the local neighborhood of each node (incorporating the state assignment) corresponds to a pattern from a finite supply of spheres.

Definition 2.3 (Graph Acceptor [27,29])
A *graph acceptor* over (Σ, C) is a structure $\mathcal{B} = (Q, R, \mathfrak{S}, Occ)$ where

- Q is its nonempty finite set of *states*,
- $R \in \mathbb{N}$ is the *radius*,
- $\mathfrak{S}$ is a finite set of R-spheres over $(\Sigma \times Q, C)$, and
- Occ is a boolean combination of *conditions* of the form "sphere $H \in \mathfrak{S}$ occurs at least n times" where $n \in \mathbb{N}$.

A *run* of $\mathcal{B}$ on a graph $G = (E, \{\lhd_c\}_{c \in C}, \lambda) \in \mathbb{DG}(\Sigma, C)$ is a mapping $\rho : E \to Q$ such that, for each $e \in E$, the R-sphere of $(E, \{\lhd_c\}_{c \in C}, (\lambda, \rho))$ around e is isomorphic to some $H \in \mathfrak{S}$. We call ρ *accepting* if the tiling of G with spheres from $\mathfrak{S}$, which is uniquely determined by ρ, satisfies the constraints imposed by Occ. (In the tiling induced by ρ, sphere $H \in \mathfrak{S}$ occurs $|\{e \in E \mid H \cong R\text{-Sph}((E, \{\lhd_c\}_{c \in C}, (\lambda, \rho)), e)\}|$ times.) The language of $\mathcal{B}$ relative to a class $\mathcal{K} \subseteq \mathbb{DG}(\Sigma, C)$, denoted by $L_{\mathcal{K}}(\mathcal{B})$, is the set of graphs $G \in \mathcal{K}$ on which there is an accepting run of $\mathcal{B}$. Moreover, we denote by $\mathcal{GA}(\Sigma, C)_{\mathcal{K}}$ ($\mathcal{GA}_{\mathcal{K}}$ if Σ and C are clear from the context) the class $\{L \subseteq \mathcal{K} \mid L = L_{\mathcal{K}}(\mathcal{B})$ for some graph acceptor $\mathcal{B}$ over $(\Sigma, C)\}$. An interesting class of graph languages distinguishes those sets that are recognized by some graph acceptor that employs only 1-spheres [28]. We denote by 1-$\mathcal{GA}(\Sigma, C)_{\mathcal{K}}$ or 1-$\mathcal{GA}_{\mathcal{K}}$ the class $\{L \subseteq \mathcal{K} \mid L = L_{\mathcal{K}}(\mathcal{B})$ for some graph acceptor $\mathcal{B} = (Q, R, \mathfrak{S}, Occ)$ over (Σ, C) with $R = 1\}$.

Note that, considering a graph acceptor relative to the class $\mathbb{DG}$ of all graphs, its spheres themselves are contained in $\mathbb{DG}$. It might be worth noting that such a coincidence does not necessarily hold for arbitrary classes of graphs, i.e., applying graph acceptors to a subclass $\mathcal{K}$ of $\mathbb{DG}$, their spheres might still require a more general structure than $\mathcal{K}$ admits. But obviously, it always suffices to restrict to those spheres that can be *embedded* into some graph from $\mathcal{K}$ in a sense made precise below. Those considerations will play a role when we address the issue of graph acceptors over message sequence charts.

Graph acceptors can be characterized logically as follows:

Theorem 2.4 ([28,29]) For any class $\mathcal{K} \subseteq \mathbb{DG}$ of bounded degree, it holds $\mathcal{EMSO}_{\mathcal{K}} = \mathcal{GA}_{\mathcal{K}}$.

The proof relies on Hanf's Theorem [12], which basically states that any first-order sentence can be rephrased as a boolean combination of conditions "R-sphere H occurs at least $n \in \mathbb{N}$ times".

2.4 Grids

An important class of graphs is provided by *grids*, which, once more, are a special case of graphs. However, while the node-labeling is a singleton and will therefore be omitted, an edge of a grid is labeled with either 1 or 2. Let in the following $\mathbb{N}_{\geq 1}$ stand for $\mathbb{N} \setminus \{0\}$ and, given $n \in \mathbb{N}_{\geq 1}$, $[n]$ for $\{1, \ldots, n\}$.

Definition 2.5 (Grid)
Given $n, m \in \mathbb{N}_{\geq 1}$, the (n, m)-*grid* is the graph $G(n, m) := ([n] \times [m], S_1, S_2) \in \mathbb{DG}_H(-, \{1, 2\})$ where $S_1, S_2 \subseteq ([n] \times [m])^2$ contain the pairs $((i, j), (i+1, j)) \in ([n] \times [m])^2$ and $((i, j), (i, j+1)) \in ([n] \times [m])^2$, respectively.

Note that, in the context of grids, we use S_1 and S_2 rather than $\lhd_1$ and, respectively, $\lhd_2$ to denote the edge relations, because this is more common. The set of grids is denoted by $\mathbb{GR}$. A relation $R \subseteq \mathbb{N}_{\geq 1} \times \mathbb{N}_{\geq 1}$ may be represented by the grid language $\{G(n,m) \mid (n,m) \in R\}$. As a unary function $f : \mathbb{N}_{\geq 1} \to \mathbb{N}_{\geq 1}$ can be considered as a binary relation, we define the *grid language* $\mathcal{G}(f)$ of f to be the set $\{G(n, f(n)) \mid n \in \mathbb{N}_{\geq 1}\}$.

By means of grids, Matz and Thomas showed that quantifier alternation of second-order variables in MSO logic over graphs forms an infinite hierarchy.

Theorem 2.6 ([23,22]) The monadic quantifier-alternation hierarchy over $\mathbb{GR}$ is infinite.

3 Message Sequence Charts

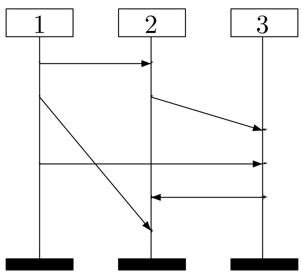
Forthcoming definitions will be made wrt. a fixed finite set P of at least two *processes*. We denote by $Ch(P)$ the set $\{(p,q) \mid p,q \in P,\ p \neq q\}$ of reliable FIFO *channels*. Thus, a message exchange is allowed between distinct processes only. Let $Act^!(P)$ denote the set $\{p!q \mid (p,q) \in Ch(P)\}$ of *send actions* while $Act^?(P)$ denotes the set $\{q?p \mid (p,q) \in Ch(P)\}$ of *receive actions*. Hereby, $p!q$ and $q?p$ are to be read as *p sends a message to q* and *q receives a message from p*, respectively. They are related in the sense that they will label communicating events of an MSC, which are joint by a message arrow in its graphical representation. Accordingly, we set $Com(P) := \{(p!q, q?p) \mid (p,q) \in Ch(P)\}$. Observe that an action $p\theta q$ ($\theta \in \{!, ?\}$) is performed by process p, which is indicated by $P(p\theta q) = p$. We let $Act(P)$ stand for the union of $Act^!(P)$ and $Act^?(P)$ and, for $p \in P$, set $Act(P)_p$ to be the set $\{\sigma \in Act(P) \mid P(\sigma) = p\}$. Moreover, we use P_{c} as a shorthand for $P \uplus \{\mathrm{c}\}$ (the symbol c will be subsequently used to label message arrows in an MSC, while a process will label the successor relation of the corresponding process line). As P will be clear from the context, we take the liberty of omitting the reference to P and just write Ch, $Act^!$, $Act^?$, Act, and Com.

Definition 3.1 (Message Sequence Chart)
A *message sequence chart* (over P) is a graph $M = (E, \{\lhd_p\}_{p \in P}, \lhd_{\mathrm{c}}, \lambda) \in \mathbb{DG}(Act, P_{\mathrm{c}})$ such that

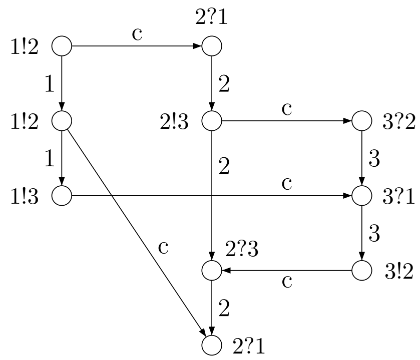
- $\lhd_p$ is the covering relation of some total order on $E_p := \lambda^{-1}(Act_p)$ (recall that this total order is then denoted by $\leq_p$),
- $\lhd_{\mathrm{c}} \subseteq E \times E$ such that, for any $e, e' \in E$, $e \lhd_{\mathrm{c}} e'$ iff
  - $(\lambda(e), \lambda(e')) \in Com$ and
  - $|M \Downarrow e|_{\lambda(e)} = |M \Downarrow e'|_{\lambda(e')}$, and
- $|M|_{p!q} = |M|_{q?p}$ for each $(p,q) \in Ch$.

Recall that λ is a labeling function of type $E \to Act$ and $\lhd^* = (\lhd_c \cup \bigcup_{p \in P} \lhd_p)^*$ is required to be a partial order. Moreover, E is a nonempty finite set of events. Events on one and the same process line are totally ordered and events on distinct process lines that are immediately concerned with each other (wrt. $\lhd_c$) are labeled with actions related by Com.

Given an MSC $(E, \{\lhd_p\}_{p \in P}, \lhd_c, \lambda)$ and $e \in E$, $P(e)$ will serve as a shorthand for $P(\lambda(e))$. The set of MSCs over P is denoted by $\mathbb{MSC}(P)$ or just $\mathbb{MSC}$. Summarizing, we model an MSC as a graph, adopting the view taken in [20,4] rather than considering partial orders [14,24,17]. As we will discuss in more detail, this does not affect our main results.



(a)



(b)

Fig. 2. An MSC

An MSC is depicted in Figure 2 (b). However, to illustrate an MSC, one mostly represents it by a diagram such as shown in Figure 2 (a), which is more intuitive and provides enough information to infer the corresponding graph. This example shows that it would be too restrictive if we confined ourselves to graphs from $\mathbb{DG}_H(Act, P_c)$, as the edge representing the second message from process 1 to process 2 is already implicitly present.

To be able to apply Theorem 2.4, the following remark will prove important.

Remark 3.2 The degree of $\mathbb{MSC}$ is bounded by 3.

Note that, for clarity, an MSC does not carry any information about the concrete messages to be sent. However, forthcoming results can be easily extended towards MSCs that are equipped with message information, as they are provided in [1,3,11], for example.

4 Message-Passing Automata and Their Expressiveness

In this section, we introduce and study *message-passing automata* (MPAs), a model of computation that is close to a real-life implementation of a communicating system.

4.1 Message-Passing Automata

An MPA is a collection of state machines that share one global initial state and several global final states. The machines are connected pairwise with a priori unbounded reliable FIFO buffers. The transitions of each component are labeled with send or receive actions. Hereby, a send action $p!q$ puts a message at the end of the channel from p to q. A receive action can be taken provided the requested message is found in the channel. To extend the expressive power, MPAs can send certain synchronization messages.

Definition 4.1 (Message-Passing Automaton)
A *message-passing automaton* (over P) is a structure $\mathcal{A} = ((\mathcal{A}_p)_{p\in P}, \mathcal{D}, \overline{s}^{in}, F)$ such that

- $\mathcal{D}$ is a nonempty finite set of *synchronization messages* (or *data*),
- for each $p \in P$, $\mathcal{A}_p$ is a pair (S_p, Δ_p) where
  - S_p is a nonempty finite set of (p-)*local states* and
  - $\Delta_p \subseteq S_p \times Act_p \times \mathcal{D} \times S_p$ is the set of (p-)*local transitions*,
- $\overline{s}^{in} \in \prod_{p\in P} S_p$ is the *global initial state*, and
- $F \subseteq \prod_{p\in P} S_p$ is the set of *global final states.*

By $S_\mathcal{A}$, we denote the set $\prod_{p\in P} S_p$ of *global states* of $\mathcal{A}$. For $\overline{s} = (s_p)_{p\in P} \in S_\mathcal{A}$, $\overline{s}[p]$ will henceforth refer to s_p.

An MPA $\mathcal{A} = ((\mathcal{A}_p)_{p\in P}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_p = (S_p, \Delta_p)$, is called deterministic if, for any $p \in P$, Δ_p satisfies the following conditions:

- If $(s, p!q, m_1, s_1) \in \Delta_p$ and $(s, p!q, m_2, s_2) \in \Delta_p$, then $m_1 = m_2$ and $s_1 = s_2$.
- If $(s, p?q, m, s_1) \in \Delta_p$ and $(s, p?q, m, s_2) \in \Delta_p$, then $s_1 = s_2$.

An MPA with set of synchronization messages $\{\circ, \bullet\}$, which is not deterministic, is illustrated in Figure 3. Note that its MSC language cannot be recognized by some MPA with only one synchronization message. Nevertheless, it can be recognized by some deterministic MPA. (To verify this is left to the reader as an exercise. Basically, the second component $\mathcal{A}_2$ has to be modified accordingly.) Let us define the behavior of MPAs. In doing so, we adhere to the style of [17]. In particular, an automaton will run

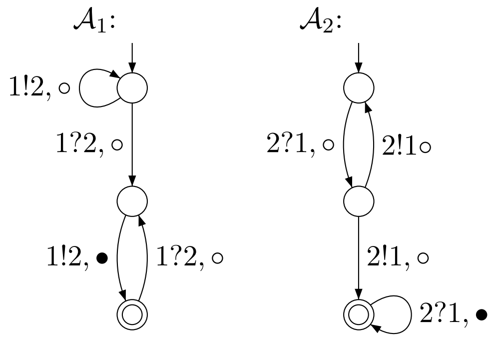


Fig. 3. A message-passing automaton

on MSCs rather than linearizations of MSCs, allowing for its distributed behavior. Let $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_p = (S_p, \Delta_p)$, be an MPA and $M = (E, \{\lhd_p\}_{p \in P}, \lhd_{\mathrm{c}}, \lambda) \in \mathbb{MSC}$ be an MSC. For a function $r : E \to \bigcup_{p \in P} S_p$, we define $r^- : E \to \bigcup_{p \in P} S_p$ to map an event $e \in E$ onto $\overline{s}^{in}[P(e)]$ if e is minimal in $(E_{P(e)}, \leq_{P(e)})$ and, otherwise, onto $r(e')$ where $e' \in E_{P(e)}$ is the unique event with $e' \lhd_{P(e)} e$. A *run* of $\mathcal{A}$ on M is a pair (r, m) of mappings $r : E \to \bigcup_{p \in P} S_p$ with $r(e) \in S_{P(e)}$ for each $e \in E$ and $m : \lhd_{\mathrm{c}} \to \mathcal{D}$ such that, for any $e, e' \in E$, $e \lhd_{\mathrm{c}} e'$ implies

- $(r^-(e), \lambda(e), m((e, e')), r(e)) \in \Delta_{P(e)}$ and
- $(r^-(e'), \lambda(e'), m((e, e')), r(e')) \in \Delta_{P(e')}$.

For $p \in P$, let f_p denote $\overline{s}^{in}[p]$ if E_p is empty. Otherwise, let f_p denote the p-local state $r(e')$ with e' maximal in $(E_p, \leq_p)$. We call (r, m) *accepting* if $(f_p)_{p \in P} \in F$. By $L(\mathcal{A}) := \{M \in \mathbb{MSC} \mid$ there is an accepting run of $\mathcal{A}$ on $M\}$, let us denote the *language* of $\mathcal{A}$. Moreover, we set $\mathcal{MPA} := \{L \subseteq \mathbb{MSC} \mid$ there is an MPA $\mathcal{A}$ such that $L = L(\mathcal{A})\}$. We also say that the languages from $\mathcal{MPA}$ are the *implementable* ones. This nomenclature is arbitrary and rather geared to the literature, where the term realizability usually refers to *locally-accepting* MPAs without any synchronization message [2,18,24].

Remark 4.2 The emptiness problem for MPAs is undecidable.

Proof Several decidability questions were studied for *communicating finite-state machines*, a slightly different variant of MPAs. Among them, the emptiness problem for communicating finite-state machines turned out to be undecidable [5]. The proof can be easily adapted towards MPAs. $\square$

Note that, for any deterministic MPA $\mathcal{A}$ and any MSC M, there is at most one run of $\mathcal{A}$ on M. However, introducing a sink state and an error message, $\mathcal{A}$ can be easily extended towards a deterministic MPA $\mathcal{A}'$ such that $L(\mathcal{A}') = L(\mathcal{A})$ and, for any MSC M, there is *exactly* one run of $\mathcal{A}'$ on M.

Consider a variant of MPAs, which allows for accepting extended MSCs, say from $\mathbb{MSC}^Q$ for some alphabet Q. Accordingly, for $p \in P$, the p-local transition relation of an MPA is henceforth a subset of $S_p \times (Act_p \times Q) \times \mathcal{D} \times S_p$. However, the type of an action (σ, q) still solely depends on σ so that, in particular, a run may allow communicating events to have different additional labelings. Such an automaton will be used in Section 5 to characterize the language of some EMSO(Act, P_c)-formula $\varphi(X_1, \ldots, X_n)$, which, as mentioned in Section 2, defines a subset of $\mathbb{MSC}^{\{0,1\}^n}$.

In [15,24,9], a run of an MPA is defined on linearizations of MSCs rather than on MSCs, which reflects an operational behavior at the expense that several execution sequences might stand for one and the same run. Usually, such a view relies on the *global transition relation* of $\mathcal{A}$, which, in turn, defers to the notion of a *configuration*. Let us be more precise and consider an MPA $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_p = (S_p, \Delta_p)$. The set of *configurations* of $\mathcal{A}$, denoted by $Conf_{\mathcal{A}}$, is the cartesian product $S_{\mathcal{A}} \times \mathcal{C}_{\mathcal{A}}$ where $\mathcal{C}_{\mathcal{A}} := \{\chi \mid \chi : Ch \to \mathcal{D}^*\}$ is the set of possible *channel contents* of $\mathcal{A}$. Now, the *global transition relation* of $\mathcal{A}$, $\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathcal{D} \times Conf_{\mathcal{A}}$, is defined as follows:

- $((\overline{s}, \chi), p!q, m, (\overline{s}', \chi')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p!q, m, \overline{s}'[p]) \in \Delta_p$,
  - $\chi' = \chi[(p,q)/m \cdot \chi((p,q))]$ (i.e., χ' maps (p,q) to $m \cdot \chi((p,q))$ and, otherwise, coincides with χ), and
  - for all $r \in P \setminus \{p\}$, $\overline{s}[r] = \overline{s}'[r]$.
- $((\overline{s}, \chi), p?q, m, (\overline{s}', \chi')) \in \Longrightarrow_{\mathcal{A}}$ if there is a word $w \in \mathcal{D}^*$ such that
  - $(\overline{s}[p], p?q, m, \overline{s}'[p]) \in \Delta_p$,
  - $\chi((q,p)) = w \cdot m$,
  - $\chi' = \chi[(q,p)/w]$, and
  - for all $r \in P \setminus \{p\}$, $\overline{s}[r] = \overline{s}'[r]$.

Let $\chi_\varepsilon : Ch \to \mathcal{D}^*$ map each channel onto the empty word. If we set $(\overline{s}^{in}, \chi_\varepsilon)$ to be the *initial configuration* and $F \times \{\chi_\varepsilon\}$ to be the set of *final configurations*, $\mathcal{A}$ defines in the canonical way a word language $L_w(\mathcal{A}) \subseteq Act^*$. As one can easily verify, it holds $L_w(\mathcal{A}) = Lin(L(\mathcal{A}))$.

4.2 The Expressiveness of Message-Passing Automata

We now turn towards our main result, which states that any EMSO-definable MSC language is implementable as an MPA and, vice versa, any MSC language recognized by some MPA has an appropriate EMSO counterpart.

The easier part is to provide an EMSO formula for a given MPA. We can hereby mainly follow similar constructions applied, for example, to finite word and asynchronous automata.

Lemma 4.3 $\quad \mathcal{MPA} \subseteq \mathcal{EMSO}_{\mathbb{MSC}}$

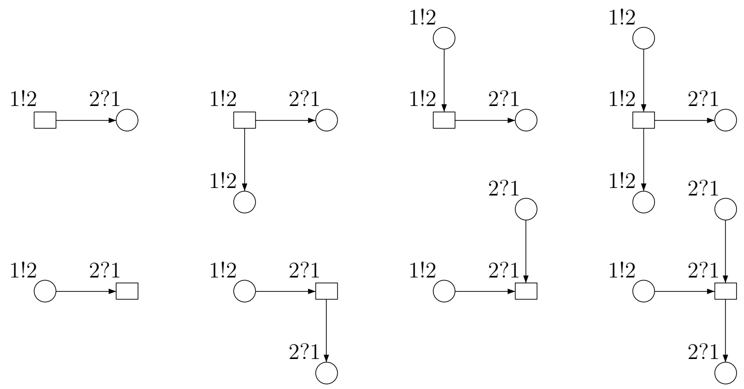


Fig. 4. A graph acceptor over (Act, P_c)

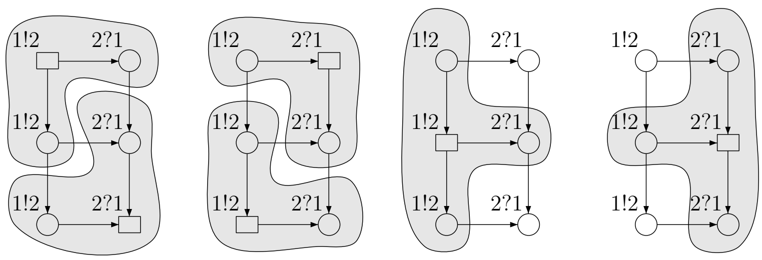


Fig. 5. The run of a graph acceptor

Proof Several instances of this problem have been considered in the literature and can be easily adapted to our setting. See [30,17], for examples. □

Corollary 4.4 The following two problems are undecidable:

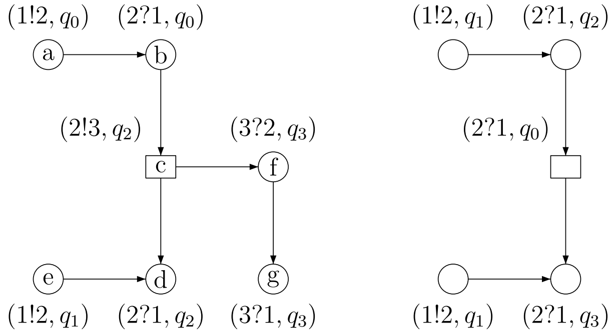
(a) Satisfiability for EMSO sentences over $\mathbb{MSC}$
(b) Universality for Σ_2-sentences over $\mathbb{MSC}$

Proof Using Remark 4.2 and Lemma 4.3, we get Corollary 4.4 (a). Corollary 4.4 (b) follows from an easy reduction from the satisfiability problem: there is an MSC satisfying a given EMSO sentence φ iff *not* any MSC satisfies the dual of φ, which can be written as a Σ_2-sentence. □

We now show that an EMSO(Act, P_c)-sentence that is interpreted over MSCs can be transformed into an equivalent MPA.

Theorem 4.5 $\mathcal{MPA} = \mathcal{EMSO}_{\mathbb{MSC}}$

Proof It remains to show inclusion from right to left. So suppose φ to be an EMSO(Act, P_c)-sentence. As $\mathbb{MSC}$ is a set of bounded degree (cf. Remark 3.2), we can, according to Theorem 2.4, assume the existence of a graph acceptor $\mathcal{B}$ over (Act, P_c) that, running on MSCs, recognizes the MSC language defined by φ. In turn, $\mathcal{B}$ will be translated into an MPA $\mathcal{A}$ that captures the application of $\mathcal{B}$ to MSCs, i.e., $L(\mathcal{A}) = L_{\mathbb{MSC}}(\mathcal{B})$. So let $\mathcal{B} = (Q, R, \mathfrak{S}, Occ)$ be a graph



(a) (b)

Fig. 6. The sphere(s) of a graph acceptor over (Act, P_c)

acceptor over (Act, P_c). (A simple graph acceptor tailored to MSCs—without occurrence constraints and a singleton as set of states—and a corresponding run are depicted in Figures 4 and 5, respectively. However, there is an equivalent graph acceptor even with radius 0.)

For our purpose, it suffices to consider only those R-spheres $H \in \mathfrak{S}$ for which there is a Q-extended MSC $M = (E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda) \in \mathbb{MSC}^Q$, which has an extended labeling function $\lambda : E \to Act \times Q$, and an event $e \in E$ such that H is the R-sphere of M around e. Other spheres cannot contribute to an MSC. Because, to become part of a run on some MSC M, an R-sphere has to admit an embedding into M. Accordingly, the 2-sphere illustrated in Figure 6 (a) may contribute to a run on an MSC (it can be complemented by a 1!3-labeled event arranged in order between the two other events of process 1), while the 2-sphere illustrated aside is irrelevant and will be ignored in the following. This assumption is essential, as it ensures that, for each $H = (E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma) \in \mathfrak{S}$ and $e \in E$, $d_H(e, \gamma) < R$ implies that E also contains a communication partner of e wrt. $\lhd_c$.

In the following, we use notions that we have introduced for MSCs also for spheres $(E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma)$ over $(Act \times Q, P_c)$, such as $P(e)$ and E_p (to indicate the process of $e \in E$ and as abbreviation for $\lambda^{-1}(Act_p \times Q)$, respectively). Note also that, wrt. spheres, $\leq_p$ is not necessarily a total order. For example, considering the 2-sphere from Figure 6 (a), $P(\mathrm{a}) = 1$, $E_1 = \{\mathrm{a}, \mathrm{e}\}$, and $\mathrm{b} \leq_2 \mathrm{d}$, but *not* $\mathrm{a} \leq_1 \mathrm{e}$. Let $maxE := \max\{|E| \mid (E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma) \in \mathfrak{S}\}$ and let $\mathfrak{S}^+$ be the set of *extended R-spheres*, i.e., the set of structures $((E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma, e), i)$ where $(E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma) \in \mathfrak{S}$, $e \in E$ is the *active node*, and $i \in \{1, \ldots, 4 \cdot maxE^2 + 1\}$ is the current *instance*. For $p \in P$, we define $\mathfrak{S}_p := \{(E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma) \in \mathfrak{S} \mid P(\gamma) = p\}$ and, furthermore, $\mathfrak{S}_p^+ := \{((E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma, e), i) \in \mathfrak{S}^+ \mid P(e) = p\}$. Finally, let $\max(Occ)$ denote the least threshold n such that Occ does not distinguish occurrence numbers $\geq n$. For readability, we let in the following $\lhd$ denote the collection $(\{\lhd_p\}_{p\in P}, \lhd_c)$ and just write $(E, \lhd, \lambda, \gamma)$ instead of $(E, \{\lhd_p\}_{p\in P}, \lhd_c, \lambda, \gamma)$.

The idea of the transformation is that, roughly speaking, $\mathcal{A}$ guesses a tiling of the MSC to be read and then verifies that the tiling corresponds to an accepting run of $\mathcal{B}$. Accordingly, a local state of $\mathcal{A}$ holds a set of *active* R-spheres, i.e., a set of spheres that play a role in its immediate environment of distance at most R. Each local state s (apart from the initial states, as we will see) carries exactly one extended R-sphere $((E, \lhd, \lambda, \gamma, e), i) \in \mathfrak{S}^+$ with $\gamma = e$, which means that a run of $\mathcal{B}$ assigns $(E, \lhd, \lambda, \gamma)$ to the event that corresponds to s. To establish isomorphism between $(E, \lhd, \lambda, \gamma)$ and the R-sphere induced by s, s transfers/obtains its obligations in form of an extended R-sphere $((E, \lhd, \lambda, \gamma, e'), i)$ to/from its immediate neighbors, respectively. For example, provided e is labeled with a send action and there is $e' \in E$ with $e \lhd_{\mathrm{c}} e'$, the message to be sent in state s will contain $((E, \lhd, \lambda, \gamma, e'), i)$, which, in turn, the receiving process understands as a requirement to be satisfied. As there may be an overlapping of isomorphic R-spheres, a state can hold several instances of one and the same sphere, which then refer to distinct states/events as corresponding sphere center. Those instances will be distinguished by means of the natural i. The benefit of i will become clear before long.

Let us turn to the construction of $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathcal{D}, \overline{s}^{in}, F)$, $\mathcal{A}_p = (S_p, \Delta_p)$, which is given as follows:

For $p \in P$, a local state of $\mathcal{A}_p$ is a pair $(\mathcal{S}, \nu)$ where

- ν is a mapping $\mathfrak{S}_p \to \{0, \ldots, \max(Occ)\}$ (let in the following ν_p^0 denote the function that maps each R-sphere $H \in \mathfrak{S}_p$ to 0) and
- $\mathcal{S}$ is either the empty set or it is a subset of $\mathfrak{S}_p^+$ such that
  - there is exactly one extended R-sphere $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}$ with $\gamma = e$ (whose component $(E, \lhd, \lambda, \gamma)$ we identify by $\varsigma(\mathcal{S})$ from now on) and
  - for any two $((E, \lhd, \lambda, \gamma, e), i), ((E', \lhd', \lambda', \gamma', e'), i') \in \mathcal{S}$,
    (a) $\lambda(e) = \lambda'(e') \in Act_p \times Q$ (so that we can assign a well-defined unique label $\lambda(\mathcal{S}) \in Act_p \times Q$ to $\mathcal{S}$, namely the labeling $\lambda(e)$ for some extended sphere $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}$) and
    (b) if $(E, \lhd, \lambda, \gamma) \cong (E', \lhd', \lambda', \gamma')$ and $i = i'$, then $e = e'$.

The set $\mathcal{D}$ of synchronization messages is the cartesian product $2^{\mathfrak{S}^+} \times 2^{\mathfrak{S}^+}$. Roughly speaking, the first component of a message contains obligations the receiving state/event has to satisfy, while the second component imposes requirements that must *not* be satisfied by the receiving process to ensure isomorphism.

Moreover, $\overline{s}^{in} = ((\emptyset, \nu_p^0))_{p \in P}$ and, for $(\mathcal{S}_p, \nu_p) \in S_p$, $((\mathcal{S}_p, \nu_p))_{p \in P} \in F$ if the union of mappings ν_p satisfies Occ and, for all $p \in P$ and $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}_p$, e is maximal in $(E_p, \leq_p)$.

So let us turn towards the definition of the p-local transition relation Δ_p and define $((\mathcal{S}, \nu), \sigma, (\mathcal{P}, \mathcal{N}), (\mathcal{S}', \nu')) \in \Delta_p$ if the following hold:

1. $\lambda(\mathcal{S}') = (\sigma, q)$ for some $q \in Q$.
2. For any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}$ and $e' \in E_p$, if $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{S}'$, then $e \lhd_p e'$.
3. For any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}'$, if $\mathcal{S} \neq \emptyset$ and e is minimal in $(E_p, \leq_p)$, then $d(e, \gamma) = R$.
4. For any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}$, if e is maximal in $(E_p, \leq_p)$, then $d(e, \gamma) = R$.
5. For any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}'$, if e is not minimal in $(E_p, \leq_p)$, then we have $((E, \lhd, \lambda, \gamma, e^-), i) \in \mathcal{S}$ where $e^- \in E_p$ is the unique event with $e^- \lhd_p e$.
6. For any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}$, if e is not maximal in $(E_p, \leq_p)$, then we have $((E, \lhd, \lambda, \gamma, e^+), i) \in \mathcal{S}'$ where $e^+ \in E_p$ is the unique event such that $e \lhd_p e^+$.
7. (i) In case that $\sigma = p!q$ for some $q \in P$:
    (a) for any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}'$ and any $e' \in E$, if $e \lhd_c e'$, then we have $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{P}$,
    (b) for any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{S}'$ and any $e' \in E$, if $e \not\lhd_c e'$, then we have $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{N}$, and
    (c) for any $((E, \lhd, \lambda, \gamma, e), i) \in \mathcal{P}$, there is $e' \in E$ such that $e' \lhd_c e$ and $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{S}'$.

   (ii) In case that $\sigma = p?q$ for some $q \in P$:
    (a) $\mathcal{P} \subseteq \mathcal{S}'$,
    (b) $\mathcal{N} \cap \mathcal{S}' = \emptyset$, and
    (c) for any $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{S}'$, if there is $e \in E$ with $e \lhd_c e'$, then $((E, \lhd, \lambda, \gamma, e'), i) \in \mathcal{P}$.
8. $\nu' = \nu[\varsigma(\mathcal{S}')/\min\{\nu(\varsigma(\mathcal{S}')) + 1, \max(Occ)\}]$ (ν' maps $\varsigma(\mathcal{S}')$ to the minimum of $\nu(\varsigma(\mathcal{S}')) + 1$ and $\max(Occ)$ and, otherwise, coincides with ν).

Thus, Condition 1. guarantees that any state within a run has the same labeling as the event it is assigned to. Condition 2. makes sure that, whenever there is a $\lhd_p$-edge in the input MSC, then there is a corresponding edge in the extended sphere that is passed from the source to the target state of the corresponding transition. Conversely, if there is no $\lhd_p$-edge between two nodes in the extended sphere, then it must not be passed directly to impose the same behavior on the MSC, i.e., the corresponding events in the MSC must not touch each other. Conditions 3. and, dually, 4. make sure that a sphere that does not make use of the whole radius R is employed in the initial or final phase of a run only. By Conditions 5. and 6., extended spheres must be passed along a process line as far as possible, hereby starting in a minimal and ending in a maximal active node. Condition 7. ensures the corresponding beyond process lines, i.e., for messages. Finally, Condition 8. guarantees that the second component of each state correctly keeps track the number of spheres used so far.

Example 4.6 In the following, let H denote the 2-sphere from Figure 6 (a).

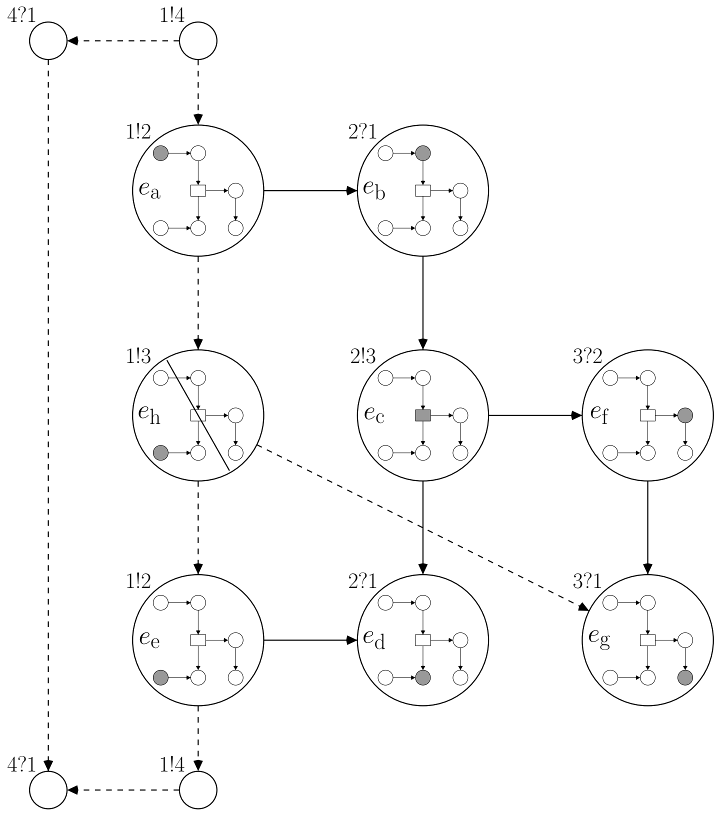


Fig. 7. Simulating a graph acceptor

Figure 7, showing some MSC M with four processes, illustrates the transition behavior of the MPA $\mathcal{A}$. It demonstrates how a run of $\mathcal{A}$ on M transfers extensions of H from one event of M to a neighboring one to make sure that the 2-sphere around event e_c (which is indicated by solid edges) is isomorphic to H. For example, the state that is taken on event e_a may contain the extended sphere (H, a). (For clarity, control states and the instance i are omitted.) As $\mathrm{a} \lhd_\mathrm{c} \mathrm{b}$ (wrt. the edge relation of H), $\mathcal{A}$ passes (H, b) in form of a message to process 2. Receiving (H, b), process 2 becomes aware it should bind e_b to some state that contains (H, b) (Conditions 7. (i) (a) and 7. (ii) (a) from the definition of the transition relation). As, in H, b is followed by c, so e_c has to be associated with a state containing (H, c) (Condition 6.). In contrast, e_h is not allowed to carry the extended sphere (H, e), unless it belongs to a different instance of H (Condition 2.). Now consider e_d, which holds the extended sphere (H, d). Due to Condition 5., the preceding state, which is associated to e_c, must contain (H, c), which means that a run cannot simply enter H beginning with d. Moreover, as e_d is a receive event, $\mathcal{A}$ has to receive a message containing (H, d) (Condition 7. (ii) (c)). In turn, the corresponding send event e_e has to be associated with a state that holds (H, e) (Condition 7. (i) (c)). As $d(a, c) = d(e, c) = 2$, the (illustrated parts of the) states assigned to e_a and e_e satisfy Conditions 3. and 4.

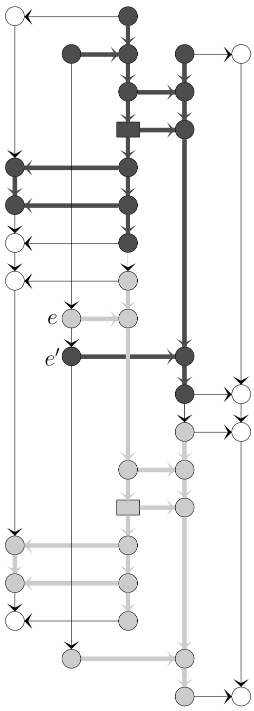


Fig. 8. Why we need different instances of extended spheres

Claim 4.7 $L_{\mathbb{MSC}}(\mathcal{B}) \subseteq L(\mathcal{A})$

Proof of Claim 4.7. Let $\rho : \widetilde{E} \to Q$ be an accepting run of $\mathcal{B}$ on the MSC $M = (\widetilde{E}, \{\widetilde{\lhd}_p\}_{p\in P}, \widetilde{\lhd}_c, \widetilde{\lambda}) \in \mathbb{MSC}$ and let in the following $\widetilde{\lhd}$ denote $\widetilde{\lhd}_c \cup \bigcup_{p\in P} \widetilde{\lhd}_p$ and $\widehat{\rho}$ stand for the mapping $\widetilde{E} \to \mathfrak{S}$ that maps an event $e \in \widetilde{E}$ onto the R-sphere of $(\widetilde{E}, \{\widetilde{\lhd}_p\}_{p\in P}, \widetilde{\lhd}_c, (\widetilde{\lambda}, \rho))$ around e. We show that there is an accepting run of $\mathcal{A}$ on M.

Consider Figure 8, which depicts an MSC inducing two isomorphic spheres, say of type H. Obviously, e' is actually not allowed to carry H forward. As the example shows, however, both e and e' must be able to carry distinct copies of H as long as they defer to distinct events of the MSC at hand as sphere centers. This is accomplished by enabling a state to carry even controversial spheres, which are then equipped with distinct instances deferring to distinct events as sphere centers. The following claim states that an assignment of instances, which resolves such a conflict and where the number of required instances only depends on $\mathcal{B}$, is always possible.

Claim 4.8 There is a mapping $i_{M,\rho} : \widetilde{E} \to \{1, \ldots, 4\cdot maxE^2+1\}$ such that, for any $e, e', e_0, e_0' \in \widetilde{E}$ with $\widehat{\rho}(e) \cong \widehat{\rho}(e')$, $e \neq e'$, $d(e_0, e) \leq R$, and $d(e_0', e') \leq R$, if $e_0 \mathrel{\widetilde{\lhd}} e_0'$ or $e_0' \mathrel{\widetilde{\lhd}} e_0$ or $e_0 = e_0'$, then $i_{M,\rho}(e) \neq i_{M,\rho}(e')$.

Proof of Claim 4.8. We can reduce the existence of $i_{M,\rho}$ to the existence of a graph coloring. Recall some basic definitions: A *graph* G is a structure $(V, Arcs)$

where V is a finite set of *vertices* and $Arcs \subseteq V \times V$ is a set of *arcs*. For a natural $n \geq 1$, a graph $G = (V, Arcs)$ is called *n-colorable* if there is a mapping $\chi : V \to \{1, \ldots, n\}$ such that $(u, v) \in Arcs$ implies $\chi(u) \neq \chi(v)$ for any two nodes $u, v \in V$ (we then say that G is *n-colored* by χ). Furthermore, for $d \in \mathbb{N}$, G is said to be of *degree d* if $d = \max\{|Arcs(u)| \mid u \in V\}$ where, for $u \in V$, $Arcs(u) = \{v \in V \mid (u, v) \in Arcs \text{ or } (v, u) \in Arcs\}$. It is easy to show that, for any $d \in \mathbb{N}$ and any graph G of degree d without self-loops, G is $(d+1)$-colorable. The mapping $i_{M,\rho}$ can now be obtained as follows: Let G be the graph $(\widetilde{E}, Arcs)$ where, for any $e, e' \in \widetilde{E}$, $(e, e') \in Arcs$ iff $e \neq e'$, $\widehat{\rho}(e) \cong \widehat{\rho}(e')$, and there is $e_0, e_0' \in \widetilde{E}$ with $d(e_0, e) \leq R$, $d(e_0', e') \leq R$, and ($e_0 \mathrel{\widetilde{\lhd}} e_0'$ or $e_0' \mathrel{\widetilde{\lhd}} e_0$ or $e_0 = e_0'$). As G cannot be of degree greater than $4 \cdot maxE^2$ (for each $e \in \widetilde{E}$, there are at most four distinct events $e' \in \widetilde{E}$ such that $e \mathrel{\widetilde{\lhd}} e'$, $e' \mathrel{\widetilde{\lhd}} e$, or $e = e'$), it can be $4 \cdot maxE^2 + 1$-colored by some mapping $\chi : \widetilde{E} \to \{1, \ldots, 4 \cdot maxE^2 + 1\}$. Now set $i_{M,\rho}$ to be χ. This concludes the proof of Claim 4.8. ■

Now let $i_{M,\rho}$ be the mapping from the above construction. For $H \in \mathfrak{S}$ and $e \in \widetilde{E}$, let furthermore $le_M(H, e) = |\{e' \in \widetilde{E}_{P(e)} \mid e' \mathrel{\widetilde{\leq}}_{P(e)} e,\ H \cong \widehat{\rho}(e')\}|$ and the mapping $r : \widetilde{E} \to \bigcup_{p \in P} S_p$ be given as follows: for $e \in \widetilde{E}$, we define $r(e) = (\mathcal{S}, \nu)$ where

(1) $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ iff there is an event $e' \in \widetilde{E}$ such that $d(e', e) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e)$, and $i = i_{M,\rho}(e')$, and
(2) for $H \in \mathfrak{S}_{P(e)}$, $\nu(H) = \min\{le_M(H, e), \max(Occ)\}$.

For $e \in \widetilde{E}$, we first verify that, in fact, $r(e) = (\mathcal{S}, \nu)$ is a valid state of $\mathcal{A}$. So suppose there are extended R-spheres $((E, \lhd, \lambda, \gamma, e_0), i), ((E', \lhd', \lambda', \gamma', e_0'), i') \in \mathcal{S}$. Of course, it holds $\lambda(e_0) = \lambda'(e_0')$. Assume now that both $\gamma = e_0$ and $\gamma' = e_0'$. But then $(E, \lhd, \lambda, \gamma, \gamma) \cong (\widehat{\rho}(e), e)$ and $(E', \lhd', \lambda', \gamma', \gamma') \cong (\widehat{\rho}(e), e)$ imply $(E, \lhd, \lambda, \gamma, \gamma) \cong (E', \lhd', \lambda', \gamma', \gamma')$. In particular, it holds $\varsigma(\mathcal{S}) = (E, \lhd, \lambda, \gamma) \cong \widehat{\rho}(e)$. Furthermore, $i = i' = i_{M,\rho}(e)$. Now assume $(E, \lhd, \lambda, \gamma) \cong (E', \lhd', \lambda', \gamma')$ and $i = i'$. There are events $e_1, e_2 \in \widetilde{E}$ such that $d(e_1, e) \leq R$, $d(e_2, e) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e_1), e)$, $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e_2), e)$, and $i = i_{M,\rho}(e_1) = i_{M,\rho}(e_2)$. Clearly, we have $\widehat{\rho}(e_1) \cong \widehat{\rho}(e_2)$. Furthermore, $e_1 = e_2$ and, consequently, $e_0 = e_0'$. Because $e_1 \neq e_2$, according to Claim 4.8, implies $i_{M,\rho}(e_1) \neq i_{M,\rho}(e_2)$, which contradicts the premise.

Let $m : \widetilde{\lhd}_{\mathrm{c}} \to \mathcal{D}$ map a pair $(e_s, e_r) \in \widetilde{\lhd}_{\mathrm{c}}$ onto $(\mathcal{P}, \mathcal{N})$ where (set $(\mathcal{S}, \nu)$ to be $r(e_s)$) $\mathcal{P} = \{((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathfrak{S}^+ \mid \text{there is } e_0 \in E \text{ with } ((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S} \text{ and } e_0 \lhd_{\mathrm{c}} e_0'\}$ and $\mathcal{N} = \{((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathfrak{S}^+ \mid \text{there is } e_0 \in E \text{ such that } ((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S} \text{ and } e_0 \not\lhd_{\mathrm{c}} e_0'\}$. In the following, we verify that (r, m) is a run of $\mathcal{A}$ on M.

For any distinct processes $p, q \in P$, $e \in \widetilde{E}_p$, and $e_r \in \widetilde{E}_q$ with $e \mathrel{\widetilde{\lhd}}_{\mathrm{c}} e_r$, we check that $(r^-(e), \widetilde{\lambda}(e), m((e, e_r)), r(e)) \in \Delta_p$. So set $(\mathcal{S}, \nu)$ to be $r^-(e)$ and $(\mathcal{S}', \nu')$ to be $r(e)$.

1. Of course, $\lambda(\mathcal{S}') = (\widetilde{\lambda}(e), q)$ for some $q \in Q$.
2. Let $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ and $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{S}'$ for some event $e_0' \in E_p$ and let $e^- \in \widetilde{E}_p$ such that $e^- \mathrel{\widetilde{\lhd}_p} e$ (as $\mathcal{S} \neq \emptyset$, such an e^- must exist). There is $e^{-\prime}, e' \in \widetilde{E}$ such that $d(e^{-\prime}, e^-) \leq R$, $d(e', e) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e^{-\prime}), e^-)$, $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e'), e)$, and $i = i_{M,\rho}(e^{-\prime}) = i_{M,\rho}(e')$. We show $e^{-\prime} = e'$, as this implies $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e^-)$, $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e'), e)$, and $e^- \mathrel{\widetilde{\lhd}_p} e$ imply $e_0 \lhd_p e_0'$. But $e^{-\prime} \neq e'$, according to Claim 4.8, implies $i_{M,\rho}(e') \neq i_{M,\rho}(e)$, which leads to a contradiction.
3. Suppose $\mathcal{S} \neq \emptyset$ and suppose there is $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}'$ with e_0 minimal in $(E_p, <_p)$. There is $e' \in \widetilde{E}$ such that $d(e', e) \leq R$ and, moreover, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e)$. As $\mathcal{S} \neq \emptyset$, e is not minimal in $(\widetilde{E}_p, \widetilde{<}_p)$ and, consequently, $d(\gamma, e_0) = d(e', e) = R$ (if $d(e', e) < R$, e would have to be minimal in $(\widetilde{E}_p, \widetilde{<}_p)$).
4. Let $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ with e_0 maximal in $(E_p, <_p)$ and let $e^- \in \widetilde{E}_p$ such that $e^- \mathrel{\widetilde{\lhd}_p} e$. Furthermore, as $r^-(e) = r(e^-)$, there is $e^{-\prime} \in \widetilde{E}$ such that both $d(e^{-\prime}, e^-) \leq R$ and $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e^{-\prime}), e^-)$. As e^- is not maximal in $(\widetilde{E}_p, \widetilde{<}_p)$, $d(e_0, \gamma) = d(e^{-\prime}, e^-) = R$ (analogously to 3., if $d(e^{-\prime}, e^-) < R$, e^- would have to be maximal in $(\widetilde{E}_p, \widetilde{<}_p)$).
5. Suppose there is an extended R-sphere $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}'$ with e_0 not minimal in $(E_p, <_p)$. Let $e_0^- \in E$ such that $e_0^- \lhd_p e_0$. As $r(e) = (\mathcal{S}', \nu')$, there is $e' \in \widetilde{E}$ with $d(e', e) \leq R$ such that $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e)$ and $i = i_{M,\rho}(e')$. As a consequence, e is not minimal in $(\widetilde{E}_p, \widetilde{<}_p)$ so that there is $e^- \in \widetilde{E}$ with $e^- \mathrel{\widetilde{\lhd}_p} e$. As furthermore $d(e', e^-) = d(\gamma, e_0^-) \leq R$ and $(E, \lhd, \lambda, \gamma, e_0^-) \cong (\widehat{\rho}(e'), e^-)$, it holds $((E, \lhd, \lambda, \gamma, e_0^-), i) \in \mathcal{S}$.
6. Suppose there is an extended R-sphere $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ (then e is not minimal in $(\widetilde{E}_p, \widetilde{<}_p)$, so let $e^- \in \widetilde{E}_p$ such that $e^- \mathrel{\widetilde{\lhd}_p} e$) with e_0 not maximal in $(E_p, <_p)$. Let $e_0^+ \in E$ such that $e_0 \lhd_p e_0^+$. As we have $r^-(e) = r(e^-) = (\mathcal{S}, \nu)$, there exists $e^{-\prime} \in \widetilde{E}$ with $d(e^{-\prime}, e^-) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e^{-\prime}), e^-)$, and $i = i_{M,\rho}(e^{-\prime})$. Since then $d(e^{-\prime}, e) = d(\gamma, e_0^+) \leq R$ and also $(E, \lhd, \lambda, \gamma, e_0^+) \cong (\widehat{\rho}(e^{-\prime}), e)$, we have $((E, \lhd, \lambda, \gamma, e_0^+), i) \in \mathcal{S}'$.
7. Let $\mathcal{P}, \mathcal{N} \subseteq \mathfrak{S}^+$ such that $m((e, e_r)) = (\mathcal{P}, \mathcal{N})$.
   (a) Let $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}'$ and $e_0' \in E$. According to the definition of m, $e_0 \lhd_{\mathrm{c}} e_0'$ implies $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{P}$.
   (b) Let $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}'$ and $e_0' \in E$. According to the definition of m, $e_0 \not\lhd_{\mathrm{c}} e_0'$ implies $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{N}$.
   (c) Let $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{P}$. Then, due to the definition of $\mathcal{P}$, there is $e_0' \in E$ with $e_0' \lhd_{\mathrm{c}} e_0$ and $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{S}'$.
8. As $\varsigma(\mathcal{S}') \cong \widehat{\rho}(e)$ and $|\{e' \mathrel{\widetilde{\leq}_p} e \mid \varsigma(\mathcal{S}') \cong \widehat{\rho}(e')\}| = |\{e' \mathrel{\widetilde{<}_p} e \mid \varsigma(\mathcal{S}') \cong \widehat{\rho}(e')\}| + 1$, we have $\nu'(\varsigma(\mathcal{S}')) = \min\{|\{e' \mathrel{\widetilde{<}_p} e \mid \varsigma(\mathcal{S}') \cong \widehat{\rho}(e')\}| + 1, \max(Occ)\}$. Furthermore, $\nu'(H) = \nu(H)$ if $H \neq \varsigma(\mathcal{S}')$.

Verifying $(r^-(e), \widetilde{\lambda}(e), m((e_s, e)), r(e)) \in \Delta_p$ for any $e \in \widetilde{E}_p$ and $e_s \in \widetilde{E}$ with $e_s \mathrel{\widetilde{\lhd}_{\mathrm{c}}} e$ differs from the above scheme only in point 7. (set $(\mathcal{S}, \nu)$ to be $r(e_s)$ and $(\mathcal{S}', \nu')$ to be $r(e)$ and let $\mathcal{P}, \mathcal{N} \subseteq \mathfrak{S}^+$ such that $m((e_s, e)) = (\mathcal{P}, \mathcal{N})$):

7. (a) Suppose there is $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{P}$. Then there exists $e_0 \in E$ with $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ and $e_0 \lhd_c e_0'$. Due to $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$, there is $e_s' \in \widetilde{E}$ with $d(e_s', e_s) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e_s'), e_s)$, and $i = i_{M,\rho}(e_s')$. As then $d(e_s', e) = d(\gamma, e_0') \leq R$ and $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e_s'), e)$, $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{S}'$.

   (b) Suppose there is $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{N} \cap \mathcal{S}'$. Then there is $e_0 \in E$ with $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ and $e_0 \not\lhd_c e_0'$. Due to $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$, there is $e_s' \in \widetilde{E}$ satisfying $d(e_s', e_s) \leq R$, $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e_s'), e_s)$, and $i = i_{M,\rho}(e_s')$. Due to $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{S}'$, there is also $e' \in \widetilde{E}$ with $d(e', e) \leq R$, $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e'), e)$, and $i = i_{M,\rho}(e')$. Suppose $e_s' \neq e'$. But then, as $\widehat{\rho}(e_s') \cong \widehat{\rho}(e')$, $i_{M,\rho}(e_s') \neq i_{M,\rho}(e')$, which leads to a contradiction. Now suppose $e_s' = e'$. But then $e_s \mathrel{\widetilde{\lhd}_c} e$ implies $e_0 \lhd_c e_0'$, also contradicting the premise.

   (c) Suppose now there exist $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{S}'$ and $e_0 \in E$ with $e_0 \lhd_c e_0'$. Then there is $e' \in \widetilde{E}$ with $d(e', e) \leq R$, $(E, \lhd, \lambda, \gamma, e_0') \cong (\widehat{\rho}(e'), e)$, and $i = i_{M,\rho}(e')$. As we have $d(e', e_s) = d(\gamma, e_0) \leq R$ and $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e_s)$, it holds $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$ and, thus, $((E, \lhd, \lambda, \gamma, e_0'), i) \in \mathcal{P}$.

In the following, we verify that (r, m) is accepting. So set, given $p \in P$, $(\mathcal{S}_p, \nu_p)$ to be $(\emptyset, \nu_p^0)$ if $\widetilde{E}_p$ is empty and, otherwise, $(\mathcal{S}_p, \nu_p)$ to be $r(e_p)$ where $e_p \in \widetilde{E}_p$ is the maximal event wrt. $\widetilde{\leq}_p$. Clearly, the union of mappings ν_p carries, for each $H \in \mathfrak{S}$, the number of occurrences of H in $\widehat{\rho}$. Furthermore, for all $p \in P$ and $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}_p$, e_0 is maximal in $(E_p, <_p)$. Because suppose there is $e_0' \in E$ with $e_0 \lhd_p e_0'$. But then, as there exists no $e^+ \in \widetilde{E}$ satisfying $e_p \mathrel{\widetilde{\lhd}_p} e^+$, there is no $e' \in \widetilde{E}$ either with $d(e', e_p) \leq R$ such that $(E, \lhd, \lambda, \gamma, e_0) \cong (\widehat{\rho}(e'), e_p)$, which contradicts the definition of r. This concludes the proof of Claim 4.7. □

Claim 4.9 $L(\mathcal{A}) \subseteq L_{\mathbb{MSC}}(\mathcal{B})$

Proof of Claim 4.9. Let (r, m) be an accepting run of $\mathcal{A}$ on the MSC $M = (\widetilde{E}, \{\widetilde{\lhd}_p\}_{p \in P}, \widetilde{\lhd}_c, \widetilde{\lambda}) \in \mathbb{MSC}$ (again, let $\widetilde{\lhd}$ denote $\widetilde{\lhd}_c \cup \bigcup_{p \in P} \widetilde{\lhd}_p$). We define $\rho : \widetilde{E} \to Q$ to map an event $e \in \widetilde{E}$ to the control state that is associated with the sphere center of $\varsigma(\mathcal{S})$ where $r(e) = (\mathcal{S}, \nu)$ for some ν. In other words, let ρ be given by $\rho(e) = q$ if there are $\mathcal{S}$, ν, and σ such that $r(e) = (\mathcal{S}, \nu)$ and $\lambda(\mathcal{S}) = (\sigma, q)$. Then ρ turns out to be an accepting run of $\mathcal{B}$ on M. First, let $\widehat{\rho}$ be the mapping $\widetilde{E} \to \mathfrak{S}$ with $\widehat{\rho}(e) = H$ if there are $\mathcal{S}$ and ν such that $r(e) = (\mathcal{S}, \nu)$ and $H = \varsigma(\mathcal{S})$. For an extended R-sphere $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathfrak{S}^+$ and $e \in \widetilde{E}$, we write $((E, \lhd, \lambda, \gamma, e_0), i) \in r(e)$ if there are $\mathcal{S}$ and ν such that $r(e) = (\mathcal{S}, \nu)$ and $((E, \lhd, \lambda, \gamma, e_0), i) \in \mathcal{S}$.

Claim 4.10 For each $e \in \widetilde{E}$, $((E, \lhd, \lambda, \gamma, \bar{e}), i) \in r(e)$, and $d \in \mathbb{N}$, if there is a sequence of events $e_0, \ldots, e_d \in E$ such that $e_0 = \bar{e}$ and, for each $k \in$

$\{0, \dots, d-1\}$, $e_k \lhd e_{k+1}$ or $e_{k+1} \lhd e_k$, then there is a unique sequence of events $\widehat{e}_0, \dots, \widehat{e}_d \in \widetilde{E}$ with

- $\widehat{e}_0 = e$,
- for each $k \in \{0, \dots, d\}$, $((E, \lhd, \lambda, \gamma, e_k), i) \in r(\widehat{e}_k)$, and
- for each $k \in \{0, \dots, d-1\}$, $\widehat{e}_k \mathrel{\widetilde{\lhd}} \widehat{e}_{k+1}$ iff $e_k \lhd e_{k+1}$ and $\widehat{e}_{k+1} \mathrel{\widetilde{\lhd}} \widehat{e}_k$ iff $e_{k+1} \lhd e_k$.

Proof of Claim 4.10. We proceed by induction. Obviously, the statement holds for $d = 0$. Now assume there is a sequence of events $e_0, \dots, e_d, e_{d+1} \in E$ such that $e_0 = \bar{e}$ and, for each $k \in \{0, \dots, d\}$, $e_k \lhd e_{k+1}$ or $e_{k+1} \lhd e_k$. By induction hypothesis, there is a unique sequence of events $\widehat{e}_0, \dots, \widehat{e}_d \in \widetilde{E}$ with

- $\widehat{e}_0 = e$,
- for each $k \in \{0, \dots, d\}$, $((E, \lhd, \lambda, \gamma, e_k), i) \in r(\widehat{e}_k)$ (in particular, $\lambda(e_k) = (\widetilde{\lambda}(\widehat{e}_k), q)$ for some $q \in Q$), and
- for each $k \in \{0, \dots, d-1\}$, $\widehat{e}_k \mathrel{\widetilde{\lhd}} \widehat{e}_{k+1}$ iff $e_k \lhd e_{k+1}$ (which implies, for one thing, $\widehat{e}_k \mathrel{\widetilde{\lhd}_c} \widehat{e}_{k+1}$ iff $e_k \lhd_c e_{k+1}$) and $\widehat{e}_{k+1} \mathrel{\widetilde{\lhd}} \widehat{e}_k$ iff $e_{k+1} \lhd e_k$.

Suppose that

- $e_d \lhd_p e_{d+1}$ for some $p \in P$. As e_d is not maximal in $(E_p, <_p)$, $r(\widehat{e}_d)$ cannot be part of a final state so that there is a (unique) event $\widehat{e}_{d+1} \in \widetilde{E}$ with $\widehat{e}_d \mathrel{\widetilde{\lhd}_p} \widehat{e}_{d+1}$. Furthermore, due to item 6. from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e_{d+1}), i) \in r(\widehat{e}_{d+1})$.
- $e_{d+1} \lhd_p e_d$ for some $p \in P$. As e_d is not minimal in $(E_p, <_p)$, there is, according to item 5. from the definition of Δ_p, a (unique) event $\widehat{e}_{d+1} \in \widetilde{E}$ with $\widehat{e}_{d+1} \mathrel{\widetilde{\lhd}_p} \widehat{e}_d$ and $((E, \lhd, \lambda, \gamma, e_{d+1}), i) \in r(\widehat{e}_{d+1})$.
- $e_d \lhd_c e_{d+1}$. There is a (unique) event $\widehat{e}_{d+1} \in \widetilde{E}$ with $\widehat{e}_d \mathrel{\widetilde{\lhd}_c} \widehat{e}_{d+1}$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}_d, \widehat{e}_{d+1}))$. According to item 7. (i) (a) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e_{d+1}), i) \in \mathcal{P}$. With 7. (ii) (a), it follows $((E, \lhd, \lambda, \gamma, e_{d+1}), i) \in r(\widehat{e}_{d+1})$.
- $e_{d+1} \lhd_c e_d$. There is a (unique) event $\widehat{e}_{d+1} \in \widetilde{E}$ with $\widehat{e}_{d+1} \mathrel{\widetilde{\lhd}_c} \widehat{e}_d$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}_{d+1}, \widehat{e}_d))$. According to item 7. (ii) (c) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e_d), i) \in \mathcal{P}$. With 7. (i) (c), it follows $((E, \lhd, \lambda, \gamma, e_{d+1}), i) \in r(\widehat{e}_{d+1})$.

This concludes the proof of Claim 4.10. ∎

We have to show that, for each $e \in \widetilde{E}$, the R-sphere of $(\widetilde{E}, \{\widetilde{\lhd}_p\}_{p \in P}, \widetilde{\lhd}_c, (\widetilde{\lambda}, \rho))$ around e is isomorphic to $\widehat{\rho}(e)$. So let $e \in \widetilde{E}$ and set $(E, \lhd, \lambda, \gamma)$ to be $\widehat{\rho}(e)$ and $i \in \{1, \dots, 4 \cdot maxE^2 + 1\}$ to be the unique element with $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(e)$.

Claim 4.11 For each $d \in \{0, \dots, R\}$, there is an isomorphism

$$h : d\text{-Sph}((\widetilde{E}, \{\widetilde{\lhd}_p\}_{p \in P}, \widetilde{\lhd}_{\mathrm{c}}, (\widetilde{\lambda}, \rho)), e) \to d\text{-Sph}((E, \lhd, \lambda), \gamma)$$

such that, for each $\widehat{e} \in \widetilde{E}$ with $d(\widehat{e}, e) \leq d$, $((E, \lhd, \lambda, \gamma, h(\widehat{e})), i) \in r(\widehat{e})$.

Proof of Claim 4.11. Let us proceed by induction. We easily see that the statement holds for $d = 0$. Now assume $d < R$ and there is an isomorphism $h : d\text{-Sph}((\widetilde{E}, \{\widetilde{\lhd}_p\}_{p \in P}, \widetilde{\lhd}_{\mathrm{c}}, (\widetilde{\lambda}, \rho)), e) \to d\text{-Sph}((E, \lhd, \lambda), \gamma)$ such that, for each $\widehat{e} \in \widetilde{E}$ with $d(\widehat{e}, e) \leq d$, $((E, \lhd, \lambda, \gamma, h(\widehat{e})), i) \in r(\widehat{e})$.

Extended sphere simulates MSC Suppose there is $\widehat{e}_1, \widehat{e}'_1, \widehat{e}_2, \widehat{e}'_2 \in \widetilde{E}$ such that $d(\widehat{e}_1, e) = d(\widehat{e}_2, e) = d$, $d(\widehat{e}'_1, e) = d(\widehat{e}'_2, e) = d + 1$, ($\widehat{e}_1 \mathrel{\widetilde{\lhd}} \widehat{e}'_1$ or $\widehat{e}'_1 \mathrel{\widetilde{\lhd}} \widehat{e}_1$), and ($\widehat{e}_2 \mathrel{\widetilde{\lhd}} \widehat{e}'_2$ or $\widehat{e}'_2 \mathrel{\widetilde{\lhd}} \widehat{e}_2$). Furthermore, suppose (let e_1 and e_2 denote $h(\widehat{e}_1)$ and $h(\widehat{e}_2)$, respectively)

- $\widehat{e}_1 \mathrel{\widetilde{\lhd}_p} \widehat{e}'_1$ for some $p \in P$. As $d(\widehat{e}_1, e) < R$, we have $d(e_1, \gamma) < R$. Due to item 4. from the definition of Δ_p, e_1 is not maximal in $(E_p, <_p)$ so that there is $e'_1 \in E$ with $e_1 \lhd_p e'_1$ and, due to item 6. and $((E, \lhd, \lambda, \gamma, e_1), i) \in r(\widehat{e}_1)$, $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
- $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_p} \widehat{e}_1$ for some $p \in P$. As $d(\widehat{e}_1, e)$ is less than R, so is $d(e_1, \gamma)$. Due to item 3. from the definition of Δ_p, e_1 is not minimal in $(E_p, <_p)$ so that there is $e'_1 \in E$ with $e'_1 \lhd_p e_1$ and, due to item 5. and $((E, \lhd, \lambda, \gamma, e_1), i) \in r(\widehat{e}_1)$, $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
- $\widehat{e}_1 \mathrel{\widetilde{\lhd}_{\mathrm{c}}} \widehat{e}'_1$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}_1, \widehat{e}'_1))$. As $d(\widehat{e}_1, e) < R$ and, thus, $d(e_1, \gamma) < R$, there is $e'_1 \in E$ such that $e_1 \lhd_{\mathrm{c}} e'_1$. (This is because $(E, \lhd, \lambda, \gamma)$ can be embedded into some MSC.) According to item 7. (i) (a) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e'_1), i) \in \mathcal{P}$. Due to item 7. (ii) (a), it then follows from $((E, \lhd, \lambda, \gamma, e_1), i) \in r(\widehat{e}_1)$ that $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
- $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_{\mathrm{c}}} \widehat{e}_1$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}'_1, \widehat{e}_1))$. As $d(\widehat{e}_1, e) < R$ and, consequently, $d(e_1, \gamma) < R$, there is also $e'_1 \in E$ such that $e'_1 \lhd_{\mathrm{c}} e_1$. (Recall that $(E, \lhd, \lambda, \gamma)$ can be embedded into some MSC.) According to item 7. (ii) (c) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e_1), i) \in \mathcal{P}$. Due to item 7. (i) (c), it then follows that $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.

Thus, depending on $\widehat{e}'_1$, we obtain from e_1 a unique event $e'_1 \in E$, which we denote by $h'(\widehat{e}'_1)$. According to the above scheme, we obtain from e_2 a unique event $e'_2 \in E$, denoted by $h'(\widehat{e}'_2)$. It holds $d(e'_1, \gamma) = d(e'_2, \gamma) = d + 1$. Now suppose

- $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_p} \widehat{e}'_2$ for some $p \in P$. As we already have $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$ and $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(\widehat{e}'_2)$, it follows from item 2. of the definition of Δ_p that $e'_1 \lhd_p e'_2$.
- $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_{\mathrm{c}}} \widehat{e}'_2$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}'_1, \widehat{e}'_2))$ and suppose $e'_1 \lhd_{\mathrm{c}} e'_2$ does not hold. But then, according to items 7. (i) (b) and 7. (ii) (b) from the definition

of Δ_p, $((E, \lhd, \lambda, \gamma, e'_2), i) \in \mathcal{N}$ and $((E, \lhd, \lambda, \gamma, e'_2), i) \notin r(\widehat{e}'_2)$, resulting in a contradiction.
– $\widehat{e}'_1 = \widehat{e}'_2$. Then $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$ and $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(\widehat{e}'_1)$ implies $e'_1 = e'_2$ (otherwise, $r(\widehat{e}'_1)$ would not be a valid state of $\mathcal{A}$).

The cases $\widehat{e}'_2 \mathrel{\widetilde{\lhd}_p} \widehat{e}'_1$ and $\widehat{e}'_2 \mathrel{\widetilde{\lhd}_c} \widehat{e}'_1$ are handled analogously.

MSC simulates extended sphere Suppose there is $e_1, e'_1, e_2, e'_2 \in E$ such that $d(e_1, \gamma) = d(e_2, \gamma) = d$, $d(e'_1, \gamma) = d(e'_2, \gamma) = d + 1$, ($e_1 \lhd e'_1$ or $e'_1 \lhd e_1$) and ($e_2 \lhd e'_2$ or $e'_2 \lhd e_2$). We now proceed as in the proof of Claim 4.10. So suppose (let $\widehat{e}_1$ and $\widehat{e}_2$ denote $h^{-1}(e_1)$ and $h^{-1}(e_2)$, respectively)

– $e_1 \lhd_p e'_1$ for some $p \in P$. As e_1 is not maximal in $(E_p, <_p)$, $r(\widehat{e}_1)$ cannot be part of a final state so that there is $\widehat{e}'_1 \in \widetilde{E}$ with $\widehat{e}_1 \mathrel{\widetilde{\lhd}_p} \widehat{e}'_1$. Furthermore, due to item 6. from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
– $e'_1 \lhd_p e_1$ for some $p \in P$. As e_1 is not minimal in $(E_p, <_p)$ there is, according to item 5. from the definition of Δ_p, $\widehat{e}'_1 \in \widetilde{E}$ with $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_p} \widehat{e}_1$ and $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
– $e_1 \lhd_c e'_1$. There is $\widehat{e}'_1 \in \widetilde{E}$ with $\widehat{e}_1 \mathrel{\widetilde{\lhd}_c} \widehat{e}'_1$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}_1, \widehat{e}'_1))$. According to item 7. (i) (a) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e'_1), i) \in \mathcal{P}$. With 7. (ii) (a), it follows $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.
– $e'_1 \lhd_c e_1$. There is $\widehat{e}'_1 \in \widetilde{E}$ with $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_c} \widehat{e}_1$. Set $(\mathcal{P}, \mathcal{N})$ to be $m((\widehat{e}'_1, \widehat{e}_1))$. According to item 7. (ii) (c) from the definition of Δ_p, $((E, \lhd, \lambda, \gamma, e_1), i) \in \mathcal{P}$. With 7. (i) (c), it follows $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$.

According to the above scheme, we obtain from $\widehat{e}_2$ a unique event $\widehat{e}'_2$. Suppose

– $e'_1 \lhd_p e'_2$ for some $p \in P$. Assume $\widehat{e}'_1 \mathrel{\widetilde{\not\lhd}_p} \widehat{e}'_2$. According to the definition of the states of $\mathcal{A}$, $e'_1 \neq e'_2$, $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$, and $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(\widehat{e}'_2)$ implies $\widehat{e}'_1 \neq \widehat{e}'_2$. But then, following the scheme depicted in Figure 9, we can construct an infinite sequence $x_1, x_2, \ldots \in \widetilde{E}$ inducing an infinite set of (pairwise distinct) events: Suppose $\widehat{e}'_1 \mathrel{\widetilde{<}_p} \widehat{e}'_2$. (The other case is handled analogously.) Set $x_1 \in \widetilde{E}$ to be the unique event satisfying $\widehat{e}'_1 \mathrel{\widetilde{\lhd}_p} x_1$. We have $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(x_1)$ and $x_1 \mathrel{\widetilde{<}_p} \widehat{e}'_2$. According to Claim 4.10, there is $x_2 \in \widetilde{E}$ such that $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(x_2)$ and $x_2 \mathrel{\widetilde{<}_{P(e)}} e$. (There is a path in $(E, \lhd, \lambda)$ from e'_2 to γ that, according to Claim 4.10, takes M from $\widehat{e}'_2$ to e. Apply this path to x_1 yielding a path to a unique event $x_2 \in \widetilde{E}$ with $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(x_2)$. From $x_1 \mathrel{\widetilde{<}_p} \widehat{e}'_2$, it easily follows that $x_2 \mathrel{\widetilde{<}_{P(e)}} e$.) Similarly, there is $x_3 \in \widetilde{E}$ with $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(x_3)$ and $x_3 \mathrel{\widetilde{<}_p} \widehat{e}'_1$. Now let $x_4 \in \widetilde{E}$ be the unique event such that $x_3 \mathrel{\widetilde{\lhd}_p} x_4$ and $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(x_4)$ (as $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(\widehat{e}'_1)$, it holds $x_4 \mathrel{\widetilde{<}_p} \widehat{e}'_1$) and let, again following Claim 4.10, $x_5 \in \widetilde{E}$ be an event with $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(x_5)$ and $x_5 \mathrel{\widetilde{<}_{P(x_2)}} x_2$ and $x_6 \in \widetilde{E}$ be an event with $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(x_6)$ and $x_6 \mathrel{\widetilde{<}_p} x_3$. Continuing this scheme yields an infinite set of events, contradicting the premise that we deal with finite MSCs.

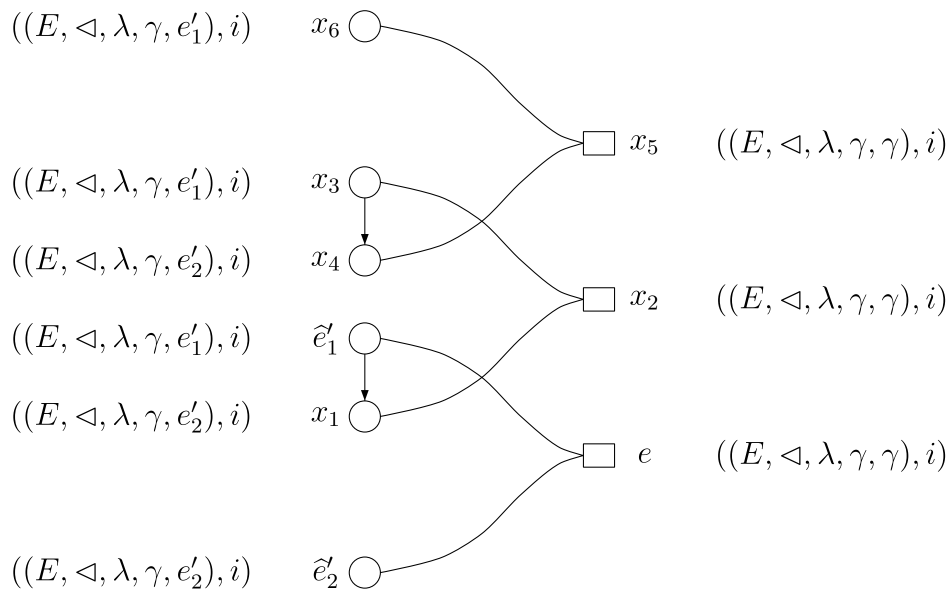


Fig. 9. An infinite sequence of events

- $e'_1 \lhd_c e'_2$. Assuming $\widehat{e}'_1 \tilde{\not\lhd}_c \widehat{e}'_2$, we proceed according to the very same scheme as in case $e'_1 \lhd_p e'_2$ to generate an infinite sequence $x_1, x_2, \ldots \in \widetilde{E}$ inducing an infinite set of events, i.e., set $x_1 \in \widetilde{E}$ to be the unique event such that $\widehat{e}'_1 \tilde{\lhd}_c x_1$ and $((E, \lhd, \lambda, \gamma, e'_2), i) \in r(x_1)$. Assuming $x_1 \tilde{<}_{P(\widehat{e}'_2)} \widehat{e}'_2$, we can find $x_2 \in \widetilde{E}$ with $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(x_2)$ and $x_2 \tilde{<}_{P(e)} e$ and so on.
- $e'_1 = e'_2$. Again, assuming $\widehat{e}'_1 \neq \widehat{e}'_2$, we generate a sequence $x_1, x_2, \ldots \in \widetilde{E}$ inducing an infinite set of events as follows: Suppose $\widehat{e}'_1 \tilde{<}_{P(\widehat{e}'_2)} \widehat{e}'_2$. According to Claim 4.10, we can find $x_1 \in \widetilde{E}$ such that $((E, \lhd, \lambda, \gamma, \gamma), i) \in r(x_1)$ and $x_1 \tilde{<}_{P(e)} e$. Furthermore, there is $x_2 \in \widetilde{E}$ satisfying $((E, \lhd, \lambda, \gamma, e'_1), i) \in r(x_2)$ and $x_2 \tilde{<}_{P(\widehat{e}'_1)} \widehat{e}'_1$ and so on.

The cases $e'_2 \lhd_p e'_1$ and $e'_2 \lhd_c e'_1$ are handled analogously. From the above results, we conclude that the map $\widehat{h} : (d{+}1)\text{-Sph}((\widetilde{E}, \{\tilde{\lhd}_p\}_{p \in P}, \tilde{\lhd}_c, (\tilde{\lambda}, \rho)), e) \to (d+1)\text{-Sph}((E, \lhd, \lambda), \gamma)$ given by

$$\widehat{h}(\widehat{e}) = \begin{cases} h(\widehat{e}) & \text{if } d(\widehat{e}, e) \leq d \\ h'(\widehat{e}) & \text{if } d(\widehat{e}, e) = d + 1 \end{cases}$$

(for $\widehat{e} \in \widetilde{E}$ with $d(\widehat{e}, e) \leq d + 1$) is an isomorphism satisfying, for any $\widehat{e} \in \widetilde{E}$ with $d(\widehat{e}, e) \leq d + 1$, $((E, \lhd, \lambda, \gamma, \widehat{h}(\widehat{e})), i) \in r(\widehat{e})$. This proves Claim 4.11. ■

As $((\mathcal{S}_p, \nu_p))_{p \in P} \in F$ only if the union of mappings ν_p is a model of Occ, an accepting run of $\mathcal{A}$ makes sure that the number of occurrences of an R-sphere meets the obligations imposed by $\mathcal{B}$. This concludes the proof of Claim 4.9 and the proof of Theorem 4.5. □

It is an easy task to transform an MPA into an equivalent graph acceptor with radius 1. In fact, two subsequent local transitions with target and, respectively, source s—where the first transition is accompanied by, say, sending a message m from p to q—can be seen as a pattern of radius 1 around a $(p!q,(s,m))$-labeled sphere center. Thus, we can conclude the following, extending known results in the settings of words, grids, and Mazurkiewicz traces [28]:

Corollary 4.12 $1\text{-}\mathcal{GA}_{\mathbb{MSC}} = \mathcal{GA}_{\mathbb{MSC}}$

5 Beyond Implementability

In this section, we turn our attention to the relation between MSO logic over MSCs and its existential fragment. We show that MSO logic is strictly more expressive than EMSO. Together with the results of the previous section, this will be used to prove that MPAs cannot be complemented in general solving an open problem raised by Kuske [17]. We then study the expressiveness of deterministic MPAs relative to the general case. Altogether, we highlight the application limitations of MPAs.

5.1 EMSO vs. MSO

Let us first recall the corresponding problem in the bounded setting where we restrict the interpretation of formulas to *bounded* MSCs, which get along with systems whose channel capacity is restricted. Those systems turned out to have simpler, more liberal logical characterizations than their unrestricted counterparts and, furthermore, enjoy some nice algorithmic properties (see [9] for an overview). In general, we distinguish two kinds of boundedness. If we require any execution of an MSCs (by which we mean a linear extension of an MSC) to correspond to a fixed channel capacity, we will speak of a *universally*-bounded MSC [14]. If, in contrast, we require at least one linearization to fit into the channel restriction, we call an MSC *existentially*-bounded [19]. While *regularity* [13] gives rise to universally-bounded MSC languages, an existential bound suffices to ensure decidability of some model-checking problems such as the problem whether an MSO formula is satisfied by all MSCs from a given *high-level MSC* [20,21]. Let $B \geq 1$. As we define boundedness in terms of linear extensions of MSCs, we first call a word $w \in Act^*$ B-bounded if, for any prefix v of w and any $(p,q) \in Ch$, $|v|_{p!q} - |v|_{q?p} \leq B$ (where $|v|_\sigma$ denotes the number of occurences of σ in v). An MSC $M \in \mathbb{MSC}$ is called *universally-B-bounded* ($\forall B$-bounded) if, for any $w \in Lin(M)$, w is B-bounded, and it is called *existentially-B-bounded* ($\exists B$-bounded) if there is at least one $w \in Lin(M)$ such that w is B-bounded. In other words, universal boundedness is safe in the sense that any possible execution sequence does not claim more memory than some given upper bound, whereas existential boundedness allows an MSC to be executed even if this does not apply to each of its linear extensions. We call an MSC language $L \subseteq \mathbb{MSC}$ $\forall B$-/$\exists B$-bounded if, for any MSC $M \in L$, M is $\forall B$-/$\exists B$-bounded. We call L $\forall$-/$\exists$-bounded if it is $\forall B$-/$\exists B$-bounded, respectively, for some B.

The following is an easy consequence of our results from the previous section and known results [15]:

Theorem 5.1 For any $\forall$-bounded MSC language L, the following statements are equivalent:

(1) $L \in \mathcal{EMSO}_{\mathbb{MSC}}$
(2) $L \in \mathcal{MSO}_{\mathbb{MSC}}$
(3) $L \in \mathcal{EMSO}[\leq]_{\mathbb{MSC}}$
(4) $L \in \mathcal{MSO}[\leq]_{\mathbb{MSC}}$
(5) $L \in \mathcal{MPA}$

Thus, our work subsumes the results by Henriksen et al. [15]. Recently, it was even shown that, if we restrict to $\exists$-bounded MSC languages, any $\mathrm{MSO}_{\mathbb{MSC}}$-definable set is implementable, generalizing Theorem 5.1:

Theorem 5.2 ([9,10]) Theorem 5.1 holds for $\exists$-bounded MSC languages verbatim.

In the following, we show that, in contrast to the bounded case (no matter if globally or existentially, as we have seen), quantifier alternation forms a hierarchy, i.e., MSO over MSCs is strictly more expressive than MPAs.

Matz and Thomas proved infinity of the monadic quantifier-alternation hierarchy over grids [23,29] (cf. Theorem 2.6). We show how grids can be encoded into MSCs and then rewrite their result in terms of MSCs adapting their proof to our setting.

Theorem 5.3 The monadic quantifier-alternation hierarchy over $\mathbb{MSC}$ is infinite.

Proof A grid $G(n, m)$ can be folded to an MSC $M(n, m)$ as exemplarily shown for $G(3, 5)$ in Figure 10. A similar encoding was used in [28] to transfer results on grids to the setting of acyclic graphs with bounded antichains. By the type of an event, we recognize which events really correspond to a node of the grid, namely those that are labeled with a send action performed by process 1 or 2. Formally, $M(n,m)$ is given by its projections as follows:

$$M(n,m) \upharpoonright (Act_1, \{1\}) = \begin{cases} (1!2)^n \, [(1?2)(1!2)]^{n((m-1)/2)} & \text{if } m \text{ is odd} \\ (1!2)^n \, [(1?2)(1!2)]^{n((m/2)-1)} \, (1?2)^n & \text{if } m \text{ is even} \end{cases}$$

$$M(n,m) \upharpoonright (Act_2, \{2\}) = \begin{cases} [(2?1)(2!1)]^{n((m-1)/2)} \, (2?1)^n & \text{if } m \text{ is odd} \\ [(2?1)(2!1)]^{n(m/2)} & \text{if } m \text{ is even} \end{cases}$$

A grid language $\mathcal{G}$ defines the MSC language $L(\mathcal{G}) := \{M(n,m) \mid G(n,m) \in \mathcal{G}\}$. For a function $f : \mathbb{N}_{\geq 1} \to \mathbb{N}_{\geq 1}$, we furthermore write $L(f)$ as a shorthand for the MSC language $L(\mathcal{G}(f))$. We now closely follow [29], which resumes the result of [23]. So let, for $k \in \mathbb{N}$, the functions $s_k, f_k : \mathbb{N}_{\geq 1} \to \mathbb{N}_{\geq 1}$ be inductively defined via $s_0(n) = n$, $s_{k+1}(n) = 2^{s_k(n)}$, $f_0(n) = n$, and $f_{k+1}(n) = f_k(n) \cdot 2^{f_k(n)}$.

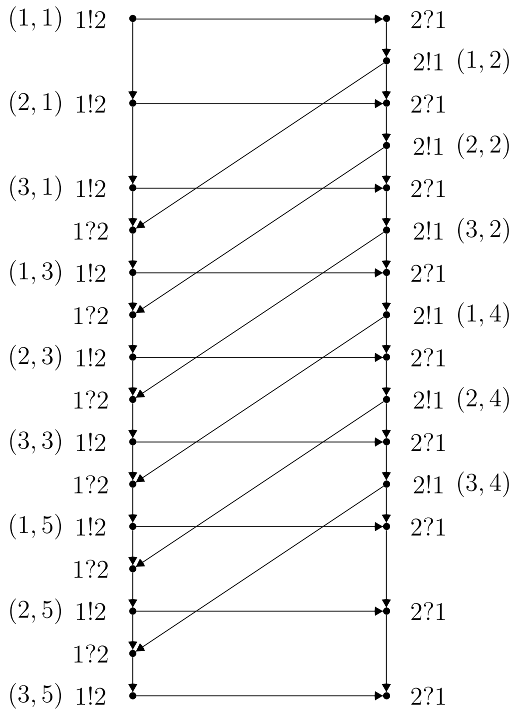


Fig. 10. Folding the $(3, 5)$-grid

Claim 5.4 For each $k \in \mathbb{N}$, the MSC language $L(f_k)$ is $(\Sigma_{2k+3})_{\mathbb{MSC}}$-definable.

Proof of Claim 5.4. We will show that, for any $k \geq 1$, if a grid language $\mathcal{G}$ is $(\Sigma_k)_{\mathbb{GR}}$-definable (over grids), then $L(\mathcal{G})$ is $(\Sigma_k)_{\mathbb{MSC}}$-definable (over MSCs). The claim then follows from the fact that any grid language $\mathcal{G}(f_k)$ is $(\Sigma_{2k+3})_{\mathbb{GR}}$-definable [29]. So let $k \in \mathbb{N}_{\geq 1}$. We can easily determine an EMSO-sentence

$\varphi_{\mathcal{GF}} = \exists\overline{X}\psi_{\mathcal{GF}}(\overline{X})$ (over MSCs) with first-order kernel $\psi_{\mathcal{GF}}(\overline{X})$ that defines the set of all grid foldings. It requires the existence of a chain iterating between processes 1 and 2. Moreover, let $\varphi = \exists\overline{Y_1}\forall\overline{Y_2}\ldots\exists/\forall\overline{Y_k}\varphi'(\overline{Y_1},\ldots,\overline{Y_k})$ be a Σ_k-sentence (over grids) where $\varphi'(\overline{Y_1},\ldots,\overline{Y_k})$ contains no set quantifiers. Without loss of generality, $\varphi_{\mathcal{GF}}$ and φ employ distinct sets of variables, which, moreover, are supposed to be different from a variable Z. We now determine the Σ_k-sentence Ψ_φ with $L_{\mathbb{MSC}}(\Psi_\varphi) = L(L_{\mathbb{GR}}(\varphi))$, i.e., the foldings of $L_{\mathbb{GR}}(\varphi)$ form exactly the MSC language defined by Ψ_φ. Namely, Ψ_φ is given by

$$\exists Z\exists\overline{X}\exists\overline{Y_1}\forall\overline{Y_2}\ldots\exists/\forall\overline{Y_k}(\psi_{bottom}(Z) \wedge \psi_{\mathcal{GF}}(\overline{X}) \wedge \|\varphi'(\overline{Y_1},\ldots,\overline{Y_k})\|_Z).$$

Hereby, the first-order formula $\psi_{bottom}(Z)$ with free variable Z makes sure that Z is reserved to those send events that correspond to the end of a column (for simplicity, Z may contain some receive events, too). This can be easily formalized starting with the requirement that Z contains the maximal send event on the first process line that is not preceded by some receive event. Furthermore, $\|\varphi'(\overline{Y_1},\ldots,\overline{Y_k})\|_Z$ is inductively derived from $\varphi'(\overline{Y_1},\ldots,\overline{Y_k})$ as follows:

– $\|S_1(x,y)\|_Z =$

$$\begin{aligned}
&\neg(x \in Z)\\
\wedge\ &\textstyle\bigvee_{\sigma\in\{1!2,2!1\}}(\lambda(x)=\sigma \wedge \lambda(y)=\sigma)\\
\wedge\ &\quad x \lessdot_1 y\\
&\vee\ \exists z(\lambda(z)=1?2 \wedge x \lessdot_1 z \wedge z \lessdot_1 y)\\
&\vee\ \exists z(\lambda(z)=2?1 \wedge x \lessdot_2 z \wedge z \lessdot_2 y)
\end{aligned}$$

– $\|S_2(x,y)\|_Z =$

$$\begin{aligned}
&\lambda(x)=1!2 \wedge \lambda(y)=2!1 \wedge \exists z(x \lessdot_{\mathrm{c}} z \wedge z \lessdot_2 y)\\
\vee\ &\lambda(x)=2!1 \wedge \lambda(y)=1!2 \wedge \exists z(x \lessdot_{\mathrm{c}} z \wedge z \lessdot_1 y)
\end{aligned}$$

– $\|\exists x\varphi\|_Z = \exists x((\bigvee_{\sigma\in\{1!2,2!1\}}\lambda(x)=\sigma) \wedge \|\varphi\|_Z)$
– $\|\forall x\varphi\|_Z = \forall x((\bigvee_{\sigma\in\{1!2,2!1\}}\lambda(x)=\sigma) \rightarrow \|\varphi\|_Z)$

The remaining constructors are derived canonically. Note that the above inductive derivation makes sure that only elements that correspond to grid nodes are assigned to $\overline{Y_1},\ldots,\overline{Y_k}$. ■

Claim 5.5 Let $f : \mathbb{N}_{\geq 1} \rightarrow \mathbb{N}_{\geq 1}$ be a function. If $L(f)$ is $(\Sigma_k)_{\mathbb{MSC}}$-definable for some $k \geq 1$, then $f(n)$ is in $s_k(\mathcal{O}(n))$.

Proof of Claim 5.5. Let $k \geq 1$ and let in the following the events of an MSC $(E, \{\lessdot_p\}_{p\in P}, \lessdot_{\mathrm{c}}, \lambda)$ be labeled with elements from $Act \times \{0,1\}^i$ for some $i \in \mathbb{N}_{\geq 1}$, i.e., $\lambda : E \rightarrow Act \times \{0,1\}^i$. But note that the type of an event still depends on the type of its communication action only. Let furthermore $\varphi(Y_1,\ldots,Y_i)$

be a Σ_k-formula defining a set of MSCs over the new label alphabet that are foldings of grids. For a fixed column length $n \geq 1$, we will build a finite (word) automaton $\mathcal{A}_n$ over $(Act \times \{0,1\}^i)^n$ with $s_{k-1}(c^n)$ states (for some constant c) that reads grid-folding MSCs column by column and is equivalent to $\varphi(Y_1, \ldots, Y_i)$ wrt. grid foldings with column length n. Column here means a sequence of communication actions, each provided with an additional label, that represents a column in the corresponding grid. For example, running on the MSC $M(3,5)$ as shown in Figure 10, $\mathcal{A}_3$ first reads the *letter* $(1!2)^3$ (recall that each action is still provided with an extra labeling, which we omit here for the sake of clarity), then continues reading $((2?1)(2!1))^3$ and so on. Then, the shortest word accepted by $\mathcal{A}_n$ has length $\leq s_{k-1}(c^n)$ so that, if $\varphi(Y_1, \ldots, Y_i)$ defines an MSC language $L(f)$ for some f, we have $f(n) \in s_k(\mathcal{O}(n))$. Let us now turn to the construction of $\mathcal{A}_n$. The formula $\varphi(Y_1, \ldots, Y_i)$ is of the form

$$\exists \overline{X_k} \forall \overline{X_{k-1}} \ldots \exists/\forall \overline{X_1} \psi(Y_1, \ldots, Y_i, \overline{X_k}, \ldots, \overline{X_1})$$

or, equivalently,

$$\exists \overline{X_k} \neg \exists \overline{X_{k-1}} \ldots \neg \exists \overline{X_1} \psi'(Y_1, \ldots, Y_i, \overline{X_k}, \ldots, \overline{X_1}).$$

We proceed by induction on k. For $k = 1$, $\varphi(Y_1, \ldots, Y_i)$ is an EMSO-formula. According to Theorem 2.4, its MSC language (consisting of MSCs with extended labelings) coincides with the MSC language of some graph acceptor. The transformation from graph acceptors to MPAs from the proof of Theorem 4.5 can be easily adapted to handle the extended labeling. Thus, $\varphi(Y_1, \ldots, Y_i)$ defines a language that is recognized by some MPA $\mathcal{A} = ((\mathcal{A}_p)_{p \in P}, \mathcal{D}, \overline{s}^{in}, F)$. The automaton $\mathcal{A}_n$ can now be obtained from $\mathcal{A}$ using a part of its global transition relation $\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times (Act \times \{0,1\}^i) \times \mathcal{D} \times Conf_{\mathcal{A}}$. Note that we have to consider only a bounded number of channel contents, as the set of grid foldings with column length n forms a $\forall n$-bounded MSC language. For some constant c, we have $(|S_{\mathcal{A}}| \cdot (|\mathcal{D}| + 1))^{|Ch| \cdot n} \leq c^n$. Thus, $c^n = s_0(c^n)$ is an upper bound for the number of states of $\mathcal{A}_n$, which only depends on the automaton $\mathcal{A}$ and, thus, on $\varphi(Y_1, \ldots, Y_i)$. The induction steps respectively involve both a complementation step (for negation) and a projection step (concerning existential quantification). While the former increases the number of states exponentially, the latter leaves it constant so that, altogether, the required number of states is obtained. This concludes the proof of Claim 5.5. ∎

As $f_{k+1}(n)$ is not in $s_k(\mathcal{O}(n))$, it follows from Claims 5.4 and 5.5 that the hierarchy of classes of $(\Sigma_k)_{\mathbb{MSC}}$-definable MSC languages is infinite. □

Corollary 5.6 $\mathcal{MPA} = \mathcal{EMSO}_{\mathbb{MSC}} \subsetneq \mathcal{MSO}_{\mathbb{MSC}}$

As, for any $f : \mathbb{N}_{\geq 1} \to \mathbb{N}_{\geq 1}$ and $(E, \{\lhd_p\}_{p \in P}, \lhd_c, \lambda) \in L(f)$, $\lhd = \lessdot$, which is first-order definable in terms of $\leq$, we obtain the following:

Corollary 5.7 $\mathcal{MSO}[\leq]_{\mathbb{MSC}}$ and $\mathcal{EMSO}_{\mathbb{MSC}}$ are incomparable wrt. inclusion.

As $\mathcal{MPA} = \mathcal{EMSO}_{\mathbb{MSC}}$, it follows from Corollary 5.6 that the complement $\mathbb{MSC} \setminus L$ of an MSC language $L \in \mathcal{MPA}$, is not necessarily contained in $\mathcal{MPA}$, too. Thus, we get the answer to an open question, which has been raised by Kuske [17].

Theorem 5.8 $\mathcal{MPA}$ is not closed under complementation.

5.2 Determinism vs. Nondeterminism

Real-life distributed systems are usually deterministic. Determinism is therefore one of the crucial properties an implementation of a distributed protocol should have. Previous results immediately affect the question of whether deterministic MPAs suffice to achieve the full expressive power of general MPAs. It is well-known that, in the framework of words and traces, any finite automaton and, respectively, any asynchronous automaton admits an equivalent deterministic counterpart. However, things are more complicated regarding MSCs. Let us first have a look at the bounded setting.

Theorem 5.9 ([25,17]) For any MPA that recognizes a $\forall$-bounded MSC language, there is an equivalent deterministic one.

The algorithm by Mukund et al. to construct from a nondeterministic MPA a deterministic counterpart is based on a technique called *time stamping*, while Kuske's construction relies on *asynchronous mappings* for traces. Unfortunately, the preceding result cannot be transferred to the unbounded setting.

Theorem 5.10 Deterministic MPAs are strictly weaker than MPAs.

Proof Recall that, without loss of generality, we can assume a deterministic MPA $\mathcal{A} = ((\mathcal{A}_p)_{p\in P}, \mathcal{D}, \overline{s}^{in}, F)$ to be complete in the sense that, for any MSC M, it allows exactly one run on M. If we set $\overline{\mathcal{A}}$ to be the deterministic MPA $((\mathcal{A}_p)_{p\in P}, \mathcal{D}, \overline{s}^{in}, S_{\mathcal{A}} \setminus F)$, it holds $L(\overline{\mathcal{A}}) = \mathbb{MSC} \setminus L(\mathcal{A})$. Thus, the class of languages recognized by some deterministic MPA is closed under complementation. However, as Theorem 5.8 states, $\mathcal{MPA}$ is not closed under complementation, which implies the theorem. $\square$

Unfortunately, Theorems 5.8 and 5.10 show that both EMSO logic and MPAs in their unrestricted form are unlikely to have some nice algorithmic properties that would attract practical interest.

6 Conclusion

Recall that we consider an MSC to be a graph, which corresponds to the view taken in [20] but is different from the one in [15,17], who model an MSC as a labeled partial order $(E, \leq, \lambda)$. However, while the way to define an MSC immediately affects the syntax and expressivity of (fragments of) the corresponding MSO logic, Theorem 5.8 holds independently of that modeling. However, our logic can only be considered to be the canonical (existential) MSO logic if MSCs are given as graphs.

Let us recall the results of the previous sections: we have studied the class of MSC languages that corresponds to EMSO logic and MPAs. By means of graph acceptors, we have shown that MPAs are expressively equivalent to EMSO logic. In particular, for every EMSO sentence, there exists an equivalent MPA. Our proof is based on results by Thomas, which, in turn, refer to Hanf's Theorem. For practical applications, it would be desirable to have a simple effective transformation from (fragments of) EMSO to MPAs of reasonable complexity. Furthermore, we proved that the class of MSC languages definable in MSO logic is strictly larger. Consequently, MPAs cannot be complemented in general. This question was raised in [17]. Finally, we showed the deterministic model of an MPA to be strictly weaker than the general one.

References

[1] R. Alur, K. Etessami, and M. Yannakakis. Inference of Message Sequence Charts. In *Proceedings of the 22nd International Conference on Software Engineering*. ACM, 2000.

[2] R. Alur, K. Etessami, and M. Yannakakis. Realizability and verification of MSC graphs. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP 2001)*, volume 2076 of *Lecture Notes in Computer Science*. Springer, 2001.

[3] R. Alur and M. Yannakakis. Model checking of message sequence charts. In *Proceedings of the 10th International Conference on Concurrency Theory (CONCUR 1999)*, volume 1664 of *Lecture Notes in Computer Science*. Springer, 1999.

[4] H. Ben-Abdallah and S. Leue. Syntactic detection of process divergence and non-local choice in message sequence charts. In *Proceedings of the 3rd International Workshop on Tools and Algorithms for Construction and Analysis of Systems (TACAS 1997)*, volume Lecture Notes in Computer Science 1217. Springer, 1997.

[5] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2), 1983.

[6] J. Büchi. Weak second order logic and finite automata. *Z. Math. Logik, Grundlag. Math.*, 5:66–62, 1960.

[7] B. Courcelle. The monadic second order logic of graphs I: recognizable sets of finite graphs. *Information and Computation*, 85:12–75, 1990.

[8] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–52, 1961.

[9] B. Genest. *L'Odyssée des Graphes de Diagrammes de Séquences (MSC-Graphes)*. PhD thesis, Laboratoire d'Informatique Algorithmique: Fondements et Applications (LIAFA), 2004.

[10] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem for a class of communicating automata with effective algorithms. In *Proceedings of the 8th International Conference on Developments in Language Theory (DLT 2004)*, volume 3340 of *Lecture Notes in Computer Science*. Springer, 2004.

[11] B. Genest, A. Muscholl, H. Seidl, and M. Zeitoun. Infinite-state high-level MSCs: Model-checking and realizability. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP 2002)*, volume 2380 of *Lecture Notes in Computer Science*. Springer, 2002.

[12] W. P. Hanf. Model-theoretic methods in the study of elementary logic. In J. W. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*. North-Holland, Amsterdam, 1965.

[13] J. G. Henriksen, M. Mukund, K. Narayan Kumar, M. Sohoni, and P. S. Thiagarajan. A theory of regular MSC languages. *Information and Computation*, 2004. to appear.

[14] J. G. Henriksen, M. Mukund, K. Narayan Kumar, and P. S. Thiagarajan. On message sequence graphs and finitely generated regular MSC languages. In *Proceedings of the 27th International Colloquium on Automata, Languages and Programming (ICALP 2000), Geneva, Switzerland*, volume 1853 of *Lecture Notes in Computer Science*. Springer, 2000.

[15] J. G. Henriksen, M. Mukund, K. Narayan Kumar, and P. S. Thiagarajan. Regular collections of message sequence charts. In *Proceedings of the 25th International Symposium Mathematical Foundations of Computer Science (MFCS 2000)*, volume 1893 of *Lecture Notes in Computer Science*. Springer, 2000.

[16] ITU-TS Recommendation Z.120: Message Sequence Chart 1999 (MSC99), 1999.

[17] D. Kuske. Regular Sets of Infinite Message Sequence Charts. *Information and Computation*, 187:80–109, 2003.

[18] M. Lohrey. Realizability of high-level message sequence charts: closing the gaps. *Theoretical Computer Science*, 309(1-3):529–554, 2003.

[19] M. Lohrey and A. Muscholl. Bounded MSC Communication. *Information and Computation*, 189(2):160–181, 2004.

[20] P. Madhusudan. Reasoning about Sequential and Branching Behaviours of Message Sequence Graphs. In *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP 2001)*, volume 2076 of *Lecture Notes in Computer Science*. Springer, 2001.

[21] P. Madhusudan and B. Meenakshi. Beyond message sequence graphs. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2001)*, volume 2245 of *Lecture Notes in Computer Science*. Springer, 2001.

[22] O. Matz, N. Schweikardt, and W. Thomas. The monadic quantifier alternation hierarchy over grids and graphs. *Information and Computation*, 179(2), 2002.

[23] O. Matz and W. Thomas. The monadic quantifier alternation hierarchy over graphs is infinite. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS 1997)*. IEEE Computer Society Press, 1997.

[24] R. Morin. Recognizable sets of message sequence charts. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS 2002), Antibes - Juan les Pins, France, March 14-16*, volume 2285 of *Lecture Notes in Computer Science*. Springer, 2002.

[25] M. Mukund, K. Narayan Kumar, and M. Sohoni. Synthesizing distributed finite-state systems from MSCs. In *Proceedings of the 11th International Conference on Concurrency Theory (CONCUR 2000)*, volume 1877 of *Lecture Notes in Computer Science*. Springer, 2000.

[26] A. Muscholl and D. Peled. Message sequence graphs and decision problems on Mazurkiewicz traces. In *Proceedings of the 24th International Symposium on Mathematical Foundations of Computer Science (MFCS 1999)*, volume 1672 of *Lecture Notes in Computer Science*. Springer, 1999.

[27] W. Thomas. On Logics, Tilings, and Automata. In *Proceedings of the 18th International Colloquium on Automata, Languages and Programming (ICALP 1991)*, volume 510 of *Lecture Notes in Computer Science*. Springer, 1991.

[28] W. Thomas. Elements of an automata theory over partial orders. In *Proceedings of Workshop on Partial Order Methods in Verification (POMIV 1996)*, volume 29 of *DIMACS*. AMS, 1996.

[29] W. Thomas. Automata theory on trees and partial orders. In *Proceedings of TAPSOFT 1997: Theory and Practice of Software Development, 7th International Joint Conference CAAP/FASE*, volume 1214 of *Lecture Notes in Computer Science*. Springer, 1997.

[30] W. Thomas. Languages, automata and logic. In A. Salomaa and G. Rozenberg, editors, *Handbook of Formal Languages*, volume 3, Beyond Words. Springer, Berlin, 1997.