Muller Message-Passing Automata and Logics ⋆

Benedikt Bollig

LSV, ENS Cachan, CNRS
61 Av. du Président Wilson, F-94235 Cachan Cedex, France

Dietrich Kuske

Institut für Informatik, Universität Leipzig
Postfach 100920 D-04009 Leipzig, Germany

Abstract

We study nonterminating message-passing automata whose behavior is described by infinite message sequence charts. As a first result, we show that Muller, Büchi, and termination-detecting Muller acceptance are equivalent for these devices. To describe the expressive power of these automata, we give a logical characterization. More precisely, we show that they have the same expressive power as the existential fragment of a monadic second-order logic featuring a first-order quantifier to express that there are infinitely many elements satisfying some property. This result is based on Vinner's extension of the classical Ehrenfeucht-Fraïssé game to cope with the infinity quantifier.

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

1 Introduction

The study of the relation between logical formalisms and operational automata devices has been a fascinating area of computer science and has produced some splendid results. From a logicians point of view, this relation allows us to decide logical theories effectively, from a system developer's point of view, the logical formalism

⋆ Work partially supported by DAAD-PROCOPE, by the ANR project DOTS, and by the project Timed-DISCOVERI.

Email addresses: `bollig@lsv.ens-cachan.fr` (Benedikt Bollig), `kuske@informatik.uni-leipzig.de` (Dietrich Kuske).

might be considered as a specification language formalizing essential properties of a system, whereas the automaton appears as a model of the system itself.

The probably most famous connection between automata theory and classical logic has been established by Büchi and Elgot, who showed that finite automata and monadic second-order (MSO) logic are expressively equivalent [3, 9]. The sequential nature of finite automata, however, limits their use in the modeling of distributed systems which called for more general automata models that employ some communication mechanism between their components. This communication can be ensured by shared variables (e.g., asynchronous automata whose behavior can be described by Mazurkiewicz traces) or by the exchange of messages along channels (e.g., message passing automata whose behavior can be described by message sequence charts). For terminating behaviors, the expressive power of these models has been related to that of some sort of MSO logic [1, 8, 11, 13, 19].

One single execution of a distributed system is often modeled as a directed acyclic graph (V, E) with a set of events V and a binary relation E that describes the causal dependency between events. Any MSO property of words [3, 9], Mazurkiewicz traces [19], or (existentially) bounded message sequence charts [11, 13] can be equivalently expressed by the appropriate automata model (and vice versa). It should be noted that the transitive closure of the causal dependency E, which forms the temporal precedence relation and is often denoted $\leq$, can be described in MSO. It can therefore also be used in the above cases. Since message-passing automata can in general not be complemented, MSO is too powerful in the context of unbounded message sequence charts [1]; but the restriction of MSO to its existential fragment (EMSO) is equivalent to message-passing automata without any channel bounds [1].

When modeling reactive systems, one is rather interested in infinite behaviors. Indeed, Büchi showed that MSO logic over infinite words is still as expressive as finite automata that require at least one final state to be visited infinitely often. Such an acceptance condition comes in many flavors, and variations thereof give rise to Büchi, Muller, Rabin, and Streett automata, which, in the nondeterministic case, are all equivalent [20]. The same applies to the settings of asynchronous (cellular) automata over infinite Mazurkiewicz traces [5, 10]. The paper [15] proposes message-passing automata with a Muller acceptance condition to make it capable of accepting infinite MSCs. As it turns out, the resulting automata model is equivalent to MSO logic over MSCs, provided the channel capacity is bounded [15].

It is the aim of this paper to lift the boundedness condition in this result, i.e., to characterize nonterminating behaviors of Muller message-passing automata with unbounded channels. After introducing the necessary notions, Section 2.3 shows that *Muller-, Büchi-, and even termination-detecting Muller-MPAs all have the same expressive power* (Theorem 8). In a termination-detecting Muller MPA, the acceptance condition can distinguish between the infinite repetition of a local state and the appearance of this state as the final one. This distinction is not directly possible in a Muller MPA. The proofs of these equivalence results use direct automata constructions. Contrary to the setting of terminating behaviors, EMSO is weaker than Muller message-passing automata: the set of infinite MSCs that send infinitely many messages from the first to the second component cannot be described by some EMSO formula. To overcome this deficiency, we introduce the additional first-order quantifier $\exists^{\infty} x \varphi(x)$ requesting infinitely many events x to satisfy some property $\varphi(x)$. As we deal with structures of bounded degree (which would not be the case if we employed the transitive closure of the edge relation), we can exploit the close connection of Ehrenfeucht-Fraïssé games and locally threshold testable languages [16, 20].

Let us come back to the setting of infinite message sequence charts. *Our main result states that* EMSO$^{\infty}$*, i.e., the extension of existential monadic second-order logic with an infinity quantifier, is expressively equivalent to message-passing automata with nonterminating behaviors* (Theorem 16). Our proof follows the route of [1] that dealt with finite message sequence charts and EMSO and could therefore build on a powerful result on this logic and its first-order fragment, namely Hanf's theorem [12]. Recall that Hanf's theorem can be proved using Ehrenfeucht-Fraïssé games. In order to have an analogue of Hanf's theorem for the extension of first-order logic by the infinity quantifier $\exists^{\infty}$, we use Vinner's extension [21] of Ehrenfeucht-Fraïssé games. This is the theme of Section 3 which leads to a Hanf-type theorem (Theorem 11). As a result, any first-order sentence with infinity quantifier can be translated into some conditions on the number of realizations of spheres. Building on [1], Section 4 shows that these conditions can be checked by message-passing automata equipped with a (termination-detecting) Muller condition. It also characterizes the expressive power of existential monadic second-order logic without the infinity quantifier by message-passing automata and the termination-detecting Staiger-Wagner acceptance condition (Theorem 18).

2 Message-Passing Automata with Nonterminating Behavior

We consider communicating systems where several sequential agents exchange messages through channels, executing send and receive actions. A send action is of the form $i!j$ indicating that agent i sends a message to agent j. The complementary receive action is denoted $j?i$. Here, agent j can read a message provided it has been sent through the corresponding channel from i to j. So let us, throughout the paper, fix a finite set Ag of *agents*. For an agent i, we denote by Σ_i the set $\{i!j, i?j \mid j \in Ag \setminus \{i\}\}$ of *actions* that are available to i. The union $\bigcup_{i \in Ag} \Sigma_i$ of all the actions is denoted Σ.

2.1 Message-Passing Automata and Their Behavior

Let us make precise our model of a reactive system with a message-passing mechanism, which goes back to Brand and Zafiropulo [2] and was later extended to deal with infinite scenarios [15]. These automata consist of independent local machines, one for each agent, that exchange messages along fifo channels. *Throughout this paper, we fix a finite set Ag of agents.*

Definition 1 *A* message-passing automaton *(or, for short, MPA) is a structure $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota)$ where*

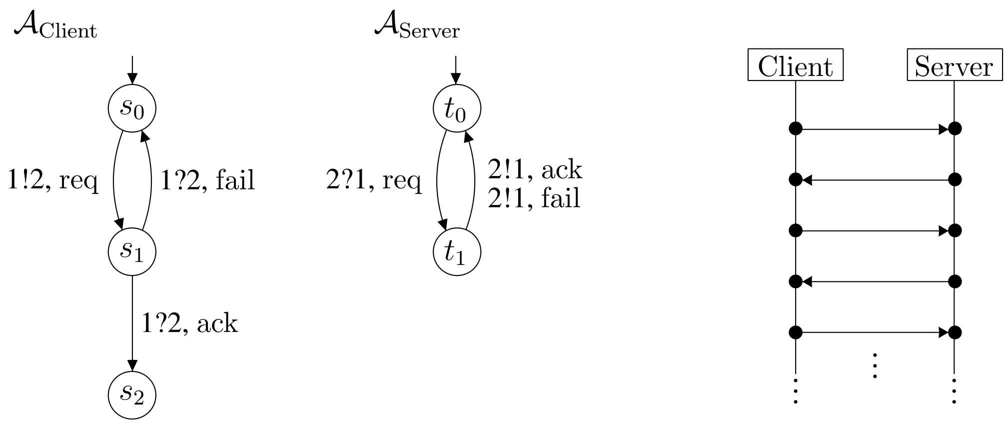
- *$\mathcal{D}$ is a nonempty finite set of* synchronization data*,*
- *for each $i \in Ag$, $\mathcal{A}_i$ is a pair (Q_i, Δ_i) where*
  - *Q_i is a finite set of* local states *and*
  - *$\Delta_i \subseteq Q_i \times \Sigma_i \times \mathcal{D} \times Q_i$ is the set of* local transitions*, and*
- *$\iota \in \prod_{i \in Ag} Q_i$ is the* global initial state*.*

The operational behavior of an MPA proceeds as one might expect. Any two local machines $\mathcal{A}_i$ and $\mathcal{A}_j$ with $i \neq j$ are connected by two fifo channels, (i, j) and (j, i), the first for sending messages from i to j and the second for the reverse direction. An agent i can execute send and receive actions according to its specification in terms of $\mathcal{A}_i$. Executing $i!j$ has the effect of writing a message into the channel (i, j). Actually, this message is supplemented by some synchronization data from $\mathcal{D}$ to extend the expressive power of MPAs. The benefit of synchronization data will become clear when we define the behavior of MPAs formally. Accordingly, $j?i$, which is executed by agent j, receives the message from i that is located at the top of the channel (i, j).

An example MPA is illustrated in Fig. 1(a). A client process communicates with a server by sending requests and receiving, on each request, either a fail message or an acknowledgment. The set of synchronization data is $\{\mathrm{req}, \mathrm{fail}, \mathrm{ack}\}$, and, e.g., $(s_0, 1!2, \mathrm{req}, s_1) \in \Delta_{\mathrm{Client}}$ is a local transition that may be taken by the client.

To describe the behavior of an MPA formally, we use the standardized formalism of *message sequence charts* (MSCs, [14]). There, the sequential behavior of an agent i is described by a vertical time-line, which will be modeled as a sequence of edges in a graph whose nodes are labeled with actions from Σ_i and referred to as *events*. Moreover, a send node and the corresponding receive node are joint by a (horizontal or diagonal) message arrow. The edge relation of an MSC gives rise to a partial order relation constraining the execution order of the nodes. Moreover, edges are labeled with elements from $C = Ag \uplus \{\mathrm{msg}\}$ to identify message and process arrows.

Definition 2 *A* message sequence chart *(MSC, for short) is an edge- and node-labeled directed graph $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$ where*



(a) An MPA over $\{\text{Client}, \text{Server}\}$ (b) An infinite MSC

Fig. 1.

- V *is the set of* events*, and* $E = \bigcup_{\ell \in C} E_\ell \subseteq V \times V$ *a set of edges,*
- $\lambda : V \to \Sigma$ *is the* event-labeling function*,*
- E^* *is a partial order on* V *(we write* $u \leq v$ *for* $(u, v) \in E^*$*),*
- *for any* $v \in V$*,* $\{u \in V \mid u \leq v\}$ *is a finite set,*
- *for any* $i \in Ag$*,* E_i *is the cover relation*[1] *of some total order on* $V_i = \lambda^{-1}(\Sigma_i)$*,*
- *for any* $(u, v) \in E_{\text{msg}}$*, there exist* $i, j \in Ag$ *distinct such that* $\lambda(u) = i!j$ *and* $\lambda(v) = j?i$*,*
- *for any* $u \in V$*, there is* $v \in V$ *such that* $(u, v) \in E_{\text{msg}}$ *or* $(v, u) \in E_{\text{msg}}$*, and*
- *for any* $(u, v), (u', v') \in E_{\text{msg}}$ *with* $\lambda(u) = \lambda(u')$*, we have* $u \leq u'$ *iff* $v \leq v'$*.*

The last condition in the definition above expresses that messages are received in the same order in which they have been sent. Hence it reflects that we deal with fifo channels only. The last three conditions ensure that E_{msg} provides a bijection between the sending and the receiving events.

Fig. 1(b) gives an example of an infinite MSC as a diagram. The events of each process are arranged along the vertical lines and messages are shown as horizontal or downward-sloping directed edges. The MSC depicts one possible behavior of the MPA from Fig. 1(a). Recall that messages sent in the MPA are considered to be synchronization messages and do not appear in the MSC itself.

To describe the behavior of our automata model formally, let $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota)$ with $\mathcal{A}_i = (Q_i, \Delta_i)$ be an MPA, and let $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$ be an MSC. For a mapping $\rho : V \to \bigcup_{i \in Ag} Q_i$ (which is a candidate for a run of $\mathcal{A}$ on M), we define the mapping $\rho^- : V \to \bigcup_{i \in Ag} Q_i$ as follows. Let $i \in Ag$ and $v \in V_i$. If we can find $u \in V_i$ such that $(u, v) \in E_i$, then we set $\rho^-(v) = \rho(u)$. If there is no such u, we let $\rho^-(v) = \iota[i]$. A *run* of $\mathcal{A}$ on M is a pair (ρ, μ) of mappings $\rho : V \to \bigcup_{i \in Ag} Q_i$ and $\mu : V \to \mathcal{D}$ such that

[1] The cover relation of a total or partial order $\preceq$ on V_i is its direct successor relation $\prec \setminus \prec^2$.

- for any $(u,v) \in E_{\mathrm{msg}}$, $\mu(u) = \mu(v)$ and
- for any $i \in Ag$ and $v \in V_i$, $(\rho^-(v), \lambda(v), \mu(v), \rho(v)) \in \Delta_i$.

2.2 *Muller, Büchi, and Staiger-Wagner Message-Passing Automata*

We will now extend our automata model with some acceptance modes that originate from the work on automata on infinite words. Recall that Büchi's acceptance condition for infinite words reads: "there is an accepting state that is assumed beyond any point in time". This formulation is also useful for finite words provided we assume that the automaton stays in the last state of its run after reading the whole word. Thus, acceptance depends on the set of states assumed cofinally. Since MPAs have local states, Inf_ρ collects, for every agent $i \in Ag$, the set of states assumed cofinally. The function Inf^+_ρ records, in addition, whether agent i performs finitely many (indicated by $\overline{\infty}$) or infinitely many (indicated by ∞) actions. Finally, the function Occ^+_ρ collects all states that agent i encounters during the run.

So let us first give the following definitions. Let $\mathcal{A} = (((Q_i, \Delta_i))_{i \in Ag}, \mathcal{D}, \iota)$ be an MPA (we set $Q = \bigcup_{i \in Ag} Q_i$) and let $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$ be an MSC. For a mapping $\rho : V \to Q$, we define functions $\mathrm{Inf}_\rho : Ag \to 2^Q$ and $\mathrm{Inf}^+_\rho, \mathrm{Occ}^+_\rho : Ag \to 2^Q \times \{\infty, \overline{\infty}\}$ as follows (with $i \in Ag$):

$$\mathrm{Inf}_\rho[i] = \begin{cases} \{q \mid \forall u \in V_i\ \exists v \in V_i : u \leq v \text{ and } q = \rho(v)\} & \text{if } V_i \neq \emptyset \\ \{\iota[i]\} & \text{otherwise} \end{cases}$$

$$\mathrm{Inf}^+_\rho[i] = \begin{cases} (\mathrm{Inf}_\rho[i], \overline{\infty}) & \text{if } V_i \text{ is finite} \\ (\mathrm{Inf}_\rho[i], \infty) & \text{otherwise} \end{cases}$$

$$\mathrm{Occ}^+_\rho[i] = \begin{cases} (\rho^{-1}(V_i), \overline{\infty}) & \text{if } V_i \text{ is finite} \\ (\rho^{-1}(V_i), \infty) & \text{otherwise} \end{cases}$$

If V_i is finite, then $\mathrm{Inf}_\rho[i]$ describes the state assumed at the event that is maximal in V_i (which is the local state $\iota[i]$ if V_i is even empty). If V_i is infinite, then $\mathrm{Inf}_\rho[i]$ is the set of states assumed infinitely often. If $\mathrm{Inf}_\rho[i]$ is a singleton, we do not know whether V_i is finite or not – this additional information is present in $\mathrm{Inf}^+_\rho[i]$. Similarly, $\mathrm{Occ}^+_\rho[i]$ provides all states that have been visited as well as the information whether there are finitely or infinitely many events on process i.

Definition 3 *A* Büchi MPA *or* Muller MPA *is a structure* $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota, \mathcal{F})$ *with* $\mathcal{A}_i = (Q_i, \Delta_i)$ *such that* $((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota)$ *is an MPA and* $\mathcal{F} \subseteq \prod_{i \in Ag} 2^{Q_i}$.

Now let (ρ, μ) *be some run of* $\mathcal{A}$ *on the MSC* $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$.

(1) If $\mathcal{A}$ *is a Büchi MPA, then the run* (ρ, μ) *is accepting if there is* $\overline{q} \in \mathcal{F}$ *such that* $\overline{q}[i] \cap \mathrm{Inf}_\rho[i] \neq \emptyset$ *for all* $i \in Ag$.

(2) If $\mathcal{A}$ is a Muller MPA, then the run (ρ, μ) is accepting if $\mathrm{Inf}_\rho \in \mathcal{F}$.

Definition 4 *A* termination-detecting Staiger-Wagner MPA *or* termination-detecting Muller MPA *is a structure $\mathcal{A} = ((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota, \mathcal{F})$ with $\mathcal{A}_i = (Q_i, \Delta_i)$ such that $((\mathcal{A}_i)_{i \in Ag}, \mathcal{D}, \iota)$ is an MPA and $\mathcal{F} \subseteq \prod_{i \in Ag}(2^{Q_i} \times \{\infty, \overline{\infty}\})$.*

Let (ρ, μ) be some run of $\mathcal{A}$ on the MSC $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$.

(1) If $\mathcal{A}$ is a termination-detecting Muller MPA, then (ρ, μ) is accepting if $\mathrm{Inf}_\rho^+ \in \mathcal{F}$.
(2) If $\mathcal{A}$ is a termination-detecting Staiger-Wagner MPA, then (ρ, μ) is accepting if $\mathrm{Occ}_\rho^+ \in \mathcal{F}$.

If $\mathcal{A}$ is some of these MPAs, then the *language $L(\mathcal{A})$ accepted by $\mathcal{A}$* is the set of those MSCs that admit an accepting run of $\mathcal{A}$.

Example 5 *Consider $\mathcal{A}$ to be the MPA over $\{\mathrm{Client}, \mathrm{Server}\}$ from Fig. 1(a). If $\mathcal{A}$ is supposed to be a Büchi MPA that is equipped with the acceptance condition $\mathcal{F} = \{(\{s_0, s_1\}, \{t_0, t_1\})\ ,\ (\{s_2\}, \{t_0\})\}$, then $L(\mathcal{A})$ contains the infinite MSC from Fig. 1(b) and, furthermore, any of its finite prefixes. In particular, a run might end up with sending a request without being followed by a server message. If, in contrast, $\mathcal{F}$ is seen as a Muller condition, then $L(\mathcal{A})$ contains, beside the infinite MSC, only those finite MSCs that end up with a message from the server to the client. In that case, $\mathcal{F}$ is equivalent to the termination-detecting Muller condition $\mathcal{F}' = \{((\{s_0, s_1\}, \infty), (\{t_0, t_1\}, \infty))\ ,\ ((\{s_2\}, \overline{\infty}), (\{t_0\}, \overline{\infty}))\}$. If, however, $\mathcal{F}'$ is considered as a termination-detecting Staiger-Wagner condition, then this admits only the infinite MSC from Fig. 1(b).*

The generalized model of termination-detecting Muller MPAs will turn out to be helpful when, in Section 4, we study the relationship between logic and MPAs. Let us first prove that termination-detecting Muller MPAs are not more expressive than Muller or Büchi MPAs (whereas termination-detecting Staiger-Wagner MPAs are strictly weaker).

2.3 Muller and Büchi MPAs vs. Termination-Detecting Muller MPAs

We examine the expressive power of our acceptance modes and start with the observation that Büchi MPAs are closed under union and intersection.

Proposition 6 *Let $\mathcal{A}^1$ and $\mathcal{A}^2$ be Büchi MPAs. There are Büchi MPAs $\mathcal{A}$ and $\mathcal{B}$ such that $L(\mathcal{A}) = L(\mathcal{A}^1) \cup L(\mathcal{A}^2)$ and $L(\mathcal{B}) = L(\mathcal{A}^1) \cap L(\mathcal{A}^2)$.*

PROOF. Suppose that $\mathcal{A}^1$ and $\mathcal{A}^2$ are given as $((\mathcal{A}_i^1)_{i\in Ag}, \mathcal{D}^1, \iota^1, \mathcal{F}^1)$ with $\mathcal{A}_i^1 = (Q_i^1, \Delta_i^1)$ and $((\mathcal{A}^2)_{i\in Ag}, \mathcal{D}^2, \iota^2, \mathcal{F}^2)$ with $\mathcal{A}_i^2 = (Q_i^2, \Delta_i^2)$, respectively. We will assume that all the sets of states and the set of synchronization messages are disjoint.

To recognize $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$, $\mathcal{A} = (((Q_i, \Delta_i))_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F})$ simulates either $\mathcal{A}_1$ or $\mathcal{A}_2$. Hence, we set $\mathcal{D} = \mathcal{D}^1 \uplus \mathcal{D}^2$ and $Q_i = Q_i^1 \uplus Q_i^2 \uplus \{\iota_i\}$ for any $i \in Ag$. Hereby, ι_i is a fresh state (i.e., it is not contained in $Q_i^1 \uplus Q_i^2$) and we set $\iota = (\iota_i)_{i\in Ag}$. For $i \in Ag$, let furthermore $\Delta_i = \Delta_i^1 \uplus \Delta_i^2 \uplus \{(\iota_i, \sigma, m, q) \mid (\iota^1[i], \sigma, m, q) \in \Delta_i^1 \text{ or } (\iota^2[i], \sigma, m, q) \in \Delta_i^2\}$. It just remains to specify the acceptance condition as $\mathcal{F} = \{(F_i)_{i\in Ag} \in \prod_{i\in Ag} 2^{Q_i} \mid \text{there is } n \in \{1,2\}$ and $(G_i)_{i\in Ag} \in \mathcal{F}^n$ such that, for any $i \in Ag$, $F_i = G_i$ or both $F_i = \{\iota_i\}$ and $G_i = \{\iota^n[i]\}\}$.

Let us construct a Büchi MPA that recognizes $L(\mathcal{A}^1) \cap L(\mathcal{A}^2)$. Since the class of MSC languages accepted by Büchi MPAs is closed under union, it suffices to consider the case $\mathcal{F}^1 = \{\overline{q}^1\}$ and $\mathcal{F}^2 = \{\overline{q}^2\}$. The Büchi MPA $\mathcal{B}$ will simulate $\mathcal{A}^1$ and $\mathcal{A}^2$ simultaneously. In addition, each process is equipped with a slightly modified flag construction [4]. We set $Q_i = Q_i^1 \times Q_i^2 \times \{0,1,2\}$ for any $i \in Ag$, and we let ι be given, for any $i \in Ag$, by

$$\iota[i] = \begin{cases} (\iota^1[i], \iota^2[i], 2) & \text{if } (\iota^1[i], \iota^2[i]) \in \overline{q}^1[i] \times \overline{q}^2[i] \\ (\iota^1[i], \iota^2[i], 0) & \text{otherwise} \end{cases}$$

The set of synchronization messages $\mathcal{D}$ is $\mathcal{D}^1 \times \mathcal{D}^2$. For $i \in Ag$, let furthermore Δ_i contain the tuple $((q_1, q_2, n), \sigma, (m_1, m_2), (q_1', q_2', n'))$ if $(q_1, \sigma, m_1, q_1') \in \Delta_i^1$, $(q_2, \sigma, m_2, q_2') \in \Delta_i^2$, and

$$n' = \begin{cases} 2 & \text{if } q_1 \in \overline{q}^1[i] \text{ and } q_2 \in \overline{q}^2[i] \\ 0 & \text{if } n = 2 \text{ and } (q_1 \notin \overline{q}^1[i] \text{ or } q_2 \notin \overline{q}^2[i]) \\ n+1 & \text{if } n < 2,\ q_{n+1} \in \overline{q}^{n+1}[i], \text{ and } (q_1 \notin \overline{q}^1[i] \text{ or } q_2 \notin \overline{q}^2[i]) \\ n & \text{otherwise} \end{cases}$$

Finally, we set $\mathcal{F} = \prod_{i\in Ag}(\overline{q}^1[i] \times \overline{q}^2[i] \times \{2\})$. This corresponds to the classical flag construction for ω-word automata, where a counter n indicates that a process is waiting for a (local) final state of $\mathcal{A}^{n+1}$. Thus, when the counter is set to 2, then a final state of each component automaton has been seen. Here, we allow in addition that the counter is set to 2 if all component states of the composite machine are accepting. This takes into consideration that some of the processes might execute only finitely many actions. $\square$

Recall that, in a termination-detecting Muller MPA, the acceptance condition can distinguish between the infinite repetition of a local state and the appearance of this state as the final one, which is not directly possible in a Muller or Büchi MPA. To solve this problem, we first state that a Büchi MPA can determine whether a particular agent performs finitely or infinitely many actions. Note that this is not the case in the word setting when considering both finite and infinite words. In our distributed setting, however, the distinction between the infinite repetition of a local state and the appearance of this state as the final one is possible.

Lemma 7 *Let $k \in Ag$. There exist Büchi MPAs $\mathcal{A}$ and $\mathcal{B}$ such that, for any MSC $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$, we have $M \in L(\mathcal{A})$ iff V_k is infinite, and $M \in L(\mathcal{B})$ iff V_k is finite.*

PROOF. The construction of $\mathcal{B}$ is straightforward. Process k has three states: the initial one, an intermediate, and a sink state; the initial and the sink state are accepting. It is forced to leave the initial state with the first event and go into the intermediate or the sink state; it can stay in the intermediate state for as long as it wishes, and it can move into the sink state nondeterministically. This final move makes sense only at the last event of process k since the run would get stuck otherwise.

We next build, for $\sigma \in \Sigma_k$, a Büchi MPA $\mathcal{A}_\sigma$ that accepts those MSCs in which σ is executed infinitely often. Then the union of all the languages $L(\mathcal{A}_\sigma)$ for $\sigma \in \Sigma_k$ can be accepted by a Büchi MPA $\mathcal{A}$ by Prop. 6. Let σ' be the communication action complementing σ, which is executed by some k' (e.g., if σ is of the form $k!k'$, then $\sigma' = k'?k$). The idea is that k and k' work together to detect that, in fact, σ and σ' occur infinitely often. Both agents toggle between states 0 and 1 when executing σ and σ', respectively. However, in the acceptance condition, k requires 0 to be taken infinitely often, whereas k' claims to visit 1 infinitely often. Formally, we set $\mathcal{A}_\sigma = (((Q_i, \Delta_i))_{i \in Ag}, \mathcal{D}, \iota, \mathcal{F})$ with $\mathcal{D} \neq \emptyset$ arbitrary, $Q_i = \{0, 1\}$ for any $i \in Ag$, and Δ_i contains any tuple $(q, \tau, m, q') \in Q_i \times \Sigma_i \times \mathcal{D} \times Q_i$ such that $\tau \in \{\sigma, \sigma'\}$ iff $q' = 1 - q$. Moreover, $\iota = (0)_{i \in Ag}$, and $\mathcal{F} = \{\overline{q}\}$ where $\overline{q}[i] = \{0\}$ for any $i \neq k'$, and $\overline{q}[k'] = \{1\}$. $\square$

We now show that Büchi and Muller MPAs are as expressive as termination-detecting Muller MPAs.

Theorem 8 *Let L be a set of MSCs. Then the following are equivalent:*

(1) there exists a Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$.
(2) there exists a Büchi MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$.
(3) there exists a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$.

PROOF. We show $(1) \rightarrow (3) \rightarrow (2) \rightarrow (1)$.

$(1) \to (3)$. Suppose $\mathcal{A} = ((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F})$ is a Muller MPA with $\mathcal{A}_i = (Q_i, \Delta_i)$. Let π_1 denote the projection of $\prod_{i\in Ag}(2^{Q_i} \times \{\infty, \overline{\infty}\})$ onto the first components. Then, let $\mathcal{F}'$ comprise all tuples $\overline{q} \in \prod_{i\in Ag}(2^{Q_i} \times \{\infty, \overline{\infty}\})$ with $\pi_1(\overline{q}) \in \mathcal{F}$. This defines a termination-detecting Muller MPA $\mathcal{A}' = ((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F}')$ that certainly accepts the same language as $\mathcal{A}$ does.

$(3) \to (2)$. Let $\mathcal{A} = ((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F})$ with $\mathcal{A}_i = (Q_i, \Delta_i)$ be some termination-detecting Muller MPA. Then, the language of the termination-detecting Muller MPA $((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \{\overline{q}\})$ is an intersection of $L(((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \{\pi_1(\overline{q})\}))$ with some sets of the form $\{M \mid V_k \text{ is infinite}\}$ and $\{M \mid V_k \text{ is finite}\}$. Since any of these sets can be accepted by a Büchi MPA (Lemma 7) and since Büchi MPAs are closed under union and intersection (Prop. 6), the implication $(3) \to (2)$ follows.

$(2) \to (1)$. Let $\mathcal{A} = ((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F})$ with $\mathcal{A}_i = (Q_i, \Delta_i)$ be a Büchi MPA. To obtain an equivalent Muller MPA $\mathcal{A}'$, we need to adapt the acceptance condition accordingly. We let $\mathcal{A}' = ((\mathcal{A}_i)_{i\in Ag}, \mathcal{D}, \iota, \mathcal{F}')$ where a tuple $(F_i')_{i\in Ag} \in \prod_{i\in Ag} 2^{Q_i}$ is contained in $\mathcal{F}'$ iff there is $(F_i)_{i\in Ag} \in \mathcal{F}$ such that, for any $i \in Ag$, $F_i \cap F_i' \neq \emptyset$. $\square$

3 Structures, Logic, and the Ehrenfeucht-Fraïssé Game

Note that MSCs can be seen as relational structures whose signature contains binary relations E_ℓ for $\ell \in C$ and unary relations R_a for $a \in \Sigma$. Since it does not cause additional difficulty and since the results of this section can be of interest also beyond MSCs, we formulate them in more generality. Throughout this section, we fix some purely relational signature σ, i.e., σ is a finite set of relation symbols (and each relation symbol has its associated arity). For $h \in \mathbb{N}$, let σ_h denote the extension of the signature σ by h constant symbols $c_1, c_2, \ldots, c_h$ (in particular, $\sigma_0 = \sigma$). A *σ_h-structure* is a tuple $\mathfrak{A} = (A, (R^{\mathfrak{A}})_{R\in\sigma}, (c_i^{\mathfrak{A}})_{1\leq i\leq h})$ where A is some set, $R^{\mathfrak{A}}$ is a relation on A whose arity is dictated by the arity of the relation symbol R, and $c_i^{\mathfrak{A}}$ is an element of A. If $\mathfrak{A}$ is a σ_h-structure and $\overline{a} = (a_1, \ldots, a_m)$ is a tuple of elements of A, then $(\mathfrak{A}, \overline{a})$ denotes the σ_{h+m}-structure that has, in addition to $\mathfrak{A}$, constants $c_{h+i}^{\mathfrak{A}} = a_i$ for $1 \leq i \leq m$.

3.1 Monadic Second-Order Logic

We fix supplies $\text{Var} = \{x, y, x_1, x_2, \ldots\}$ of *individual* and $\text{VAR} = \{X, Y, \ldots\}$ of *set variables*. The set $\text{MSO}^\infty(\sigma_h)$ of *extended monadic second-order* (or MSO^∞) formulas over σ_h is given by the following grammar:

$$\varphi ::= R(x_1, \ldots, x_n) \mid x_1 = x_2 \mid x_1 \in X \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x\, \varphi \mid \exists X \varphi \mid \exists^\infty x\, \varphi$$

where $n \in \mathbb{N}$, $R \in \sigma$ is an n-ary predicate symbol, x_i is a variable from Var or a constant symbol from σ_h, $x \in$ Var, and $X \in$ VAR.

Let $\mathfrak{A}$ be a σ_h-structure, $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n) \in \mathrm{MSO}^\infty(\sigma_h)$ be a formula, and $\overline{a} = (a_1, \ldots, a_m) \in A^m$ and $\overline{A} = (A_1, \ldots, A_n) \in (2^A)^n$ be tuples of elements and subsets of A. Then the *satisfaction* relation $\mathfrak{A} \models \varphi(\overline{a}, \overline{A})$ is defined as usual such that, for $\psi(y, x_1, \ldots, x_m, X_1, \ldots, X_n) \in \mathrm{MSO}^\infty(\sigma_h)$, $\mathfrak{A} \models (\exists^\infty y\psi)(\overline{a}, \overline{A})$ iff $\mathfrak{A} \models \psi(a, \overline{a}, \overline{A})$ for infinitely many $a \in A$.

We define the following fragments of $\mathrm{MSO}^\infty(\sigma_h)$:

(1) the *first-order fragment* $\mathrm{FO}^\infty(\sigma_h)$ comprises those formulas from $\mathrm{MSO}^\infty(\sigma_h)$ that do not contain any set quantifier
(2) the *existential fragment* $\mathrm{EMSO}^\infty(\sigma_h)$ comprises the formulas from $\mathrm{MSO}^\infty(\sigma_h)$ of the form $\exists X_1 \ldots \exists X_n\varphi$ with $\varphi \in \mathrm{FO}^\infty(\sigma_h)$
(3) *the monadic second-order fragment* $\mathrm{MSO}(\sigma_h)$ comprises those formulas from $\mathrm{MSO}^\infty(\sigma_h)$ that do not contain the quantifier $\exists^\infty$
(4) *first-order logic* $\mathrm{FO}(\sigma_h)$ equals $\mathrm{MSO}(\sigma_h) \cap \mathrm{FO}^\infty(\sigma_h)$
(5) *existential monadic second-order logic* $\mathrm{EMSO}(\sigma_h)$ comprises the formulas from $\mathrm{MSO}(\sigma_h) \cap \mathrm{EMSO}^\infty(\sigma_h)$

The *quantifier-rank* $\mathrm{qr}(\varphi)$ of a formula φ in $\mathrm{FO}^\infty(\sigma_h)$ is the nesting depth of quantifiers in φ. More precisely, $\mathrm{qr}(\varphi) = 0$ if φ is atomic, $\mathrm{qr}(\neg\varphi) = \mathrm{qr}(\varphi)$, $\mathrm{qr}(\varphi \vee \psi) = \max\{\mathrm{qr}(\varphi), \mathrm{qr}(\psi)\}$, and $\mathrm{qr}(\exists x\varphi) = \mathrm{qr}(\exists^\infty x\varphi) = \mathrm{qr}(\varphi) + 1$. For $n \in \mathbb{N}$, we denote by $\mathrm{FO}^\infty(\sigma_h)[n]$ the set of first-order formulas of quantifier rank at most n without free variables. For two σ_h-structures $\mathfrak{A}$ and $\mathfrak{B}$, we say $\mathfrak{A}$ *and* $\mathfrak{B}$ *agree on* $\mathrm{FO}^\infty(\sigma_h)[n]$ if, for all formulas $\varphi \in \mathrm{FO}^\infty(\sigma_h)[n]$, we have $\mathfrak{A} \models \varphi$ if and only if $\mathfrak{B} \models \varphi$. In other words, the structures $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by formulas of quantifier-depth at most n.

3.2 *The* FO^∞*-Game*

The FO^∞-game is an extension of the classical Ehrenfeucht-Fraïssé game, which captures the expressive power of $\mathrm{FO}(\sigma)$. It goes back to Lipner [17] and Vinner [21] (cf. also [18]). It is played between two players named *spoiler* and *duplicator*. A *game position* is a triple $(\mathfrak{A}, \mathfrak{B}, k)$ where $\mathfrak{A}$ and $\mathfrak{B}$ are structures over the same signature σ_h and $k \in \mathbb{N}$. This position is *winning (for duplicator)* if $k = 0$ and the binary relation

$$\{(c_i^{\mathfrak{A}}, c_i^{\mathfrak{B}}) \mid 1 \leq i \leq h\}$$

is a partial isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$. If $k > 0$, the game proceeds as follows (where A and B are the universes of $\mathfrak{A}$ and $\mathfrak{B}$, respectively):

(1) Spoiler chooses to proceed with (2) or (2').
(2) Spoiler chooses $a \in A$ or $b \in B$.

(3) Duplicator chooses an element in the other structure (i.e., $b \in B$ or $a \in A$).
(4) The game proceeds with $((\mathfrak{A}, a), (\mathfrak{B}, b), k-1)$.
(2') Spoiler chooses an infinite subset X of A or of B.
(3') Duplicator chooses an infinite subset Y of the other structure.
(4') Spoiler chooses $y \in Y$.
(5') Duplicator chooses $x \in X$.
(6') The game proceeds with $((\mathfrak{A}, x), (\mathfrak{B}, y), k-1)$ if $x \in A$; otherwise, it proceeds with $((\mathfrak{A}, y), (\mathfrak{B}, x), k-1)$.

For σ_h-structures $\mathfrak{A}$ and $\mathfrak{B}$ and $k \in \mathbb{N}$, we write $\mathfrak{A} \equiv_k^\infty \mathfrak{B}$ if duplicator can force the FO$^\infty$-play started in $(\mathfrak{A}, \mathfrak{B}, k)$ into a winning position.

The classical Ehrenfeucht-Fraïssé game is obtained from this game by forcing spoiler in (1) always to proceed with (2). If, in this Ehrenfeucht-Fraïssé game, duplicator can force the play started in $(\mathfrak{A}, \mathfrak{B}, k)$ into a winning position, we write $\mathfrak{A} \equiv_k \mathfrak{B}$.

The existence of a winning strategy describes precisely those properties that can be expressed using formulas of FO$^\infty(\sigma_h)[k]$ and FO$(\sigma_h)[k]$, see, e.g., [16, 21].

Theorem 9 (Ehrenfeucht-Fraïssé, Vinner) *Let $\mathfrak{A}$ and $\mathfrak{B}$ be σ_h-structures and $k \in \mathbb{N}$.*

(1) $\mathfrak{A}$ and $\mathfrak{B}$ agree on FO$(\sigma_h)[k]$ *iff $\mathfrak{A} \equiv_k \mathfrak{B}$.*
(2) $\mathfrak{A}$ and $\mathfrak{B}$ agree on FO$^\infty(\sigma_h)[k]$ *iff $\mathfrak{A} \equiv_k^\infty \mathfrak{B}$.*

3.3 Threshold Equivalence

In the context of structures of bounded degree, *threshold equivalence* provides a refinement of $\equiv_k$ and, finally, a normal form of FO formulas that restricts to counting of spheres up to a certain threshold [16, 20]. Here, we develop a similar result for the logic FO$^\infty(\sigma)$.

The *Gaifman graph* $G(\mathfrak{A})$ of a σ_h-structure $\mathfrak{A}$ is an undirected graph (A, E) with universe A (i.e., the universe of the structure $\mathfrak{A}$). Two elements $a, b \in A$ are connected by an edge (i.e., $(a, b) \in E$) if they belong to some tuple in some relation, i.e., if there is a relation symbol $R \in \sigma$ and a tuple $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ such that $a, b \in \{a_1, a_2, \ldots, a_n\}$. We will speak of the degree of a in $\mathfrak{A}$ whenever we actually mean the degree of a in the Gaifman graph of $\mathfrak{A}$. If all elements of $\mathfrak{A}$ have degree at most l, then we say that $\mathfrak{A}$ has degree at most l. Now let $a, b \in A$. Then the *distance* $d_{\mathfrak{A}}(a, b)$ (or $d(a, b)$ if $\mathfrak{A}$ is understood) denotes the minimal length of a path connecting a and b in the Gaifman graph $G(\mathfrak{A})$. For $\overline{a} = (a_1, \ldots, a_n) \in A^n$ and $b \in A$, we write $d(\overline{a}, b) = \min\{d(a_1, b), \ldots, d(a_n, b)\}$. Let $r \in \mathbb{N}$ and $\overline{c}$ denote the h-tuple of constants in the σ_h-structure $\mathfrak{A}$. The *r-sphere r-*Sph$(\mathfrak{A})$ *of* $\mathfrak{A}$ is the substructure of $\mathfrak{A}$ generated by the universe $\{b \in A \mid d_{\mathfrak{A}}(\overline{c}, b) \leq r\}$. Then also r-Sph$(\mathfrak{A})$ is a σ_h-structure whose constants are precisely $\overline{c}$. If, in the extreme, $h = 0$, then the set $\{b \in A \mid d_{\mathfrak{A}}(\overline{c}, b) \leq r\}$ is empty and the sphere is the empty structure. For an n-tuple $\overline{a}$ of elements in $\mathfrak{A}$, the *r-sphere of $\mathfrak{A}$ around $\overline{a}$* is the r-sphere of the extension $(\mathfrak{A}, \overline{a})$ of $\mathfrak{A}$ with constants $\overline{a}$.

For $t \in \mathbb{N}$, let $\sim_t$ and $\sim_t^\infty$ denote the equivalence relations on $\mathbb{N} \cup \{\infty\}$ defined by

- $m \sim_t n$ iff $m = n$ or $t < m$ and $t < n$
- $m \sim_t^\infty n$ iff $m = n$ or $t < m < \infty$ and $t < n < \infty$

Definition 10 *Let $r, t, h \in \mathbb{N}$ and let $\mathfrak{A}$ and $\mathfrak{B}$ be σ_h-structures. Then we write $\mathfrak{A} \leftrightarrows_{r,t} \mathfrak{B}$ if $|\{a' \in A \mid r\text{-Sph}(\mathfrak{A}, a') \cong \tau\}| \sim_t |\{b' \in B \mid r\text{-Sph}(\mathfrak{B}, b') \cong \tau\}|$ whenever there exists $a \in A$ with $\tau = r\text{-Sph}(\mathfrak{A}, a)$, or there exists $b \in B$ with $\tau = r\text{-Sph}(\mathfrak{B}, b)$.*

Similarly, $\leftrightarrows_{r,t}^\infty$ is defined based on $\sim_t^\infty$ instead of $\sim_t$.

In other words, $\leftrightarrows_{r,t}$ and $\leftrightarrows_{r,t}^\infty$ distinguish structures on the basis of the number of realizations of r-spheres up to some threshold t. But the former does not distinguish between "many" and "infinitely many" realizations of a sphere. The latter identifies all natural numbers $t + 1, t + 2, \ldots$, but makes a difference between any of them and infinity.

Theorem 11 *Let $h, l, n \geq 0$, $r_0 = t_0 = 0$, and for $k \geq 0$, $r_{k+1} = 3r_k + 1$ and $t_{k+1} = t_k + (h + n - k) \cdot l^{2r_k+1}$.*

Then, for any σ_h-structures $\mathfrak{A}$ and $\mathfrak{B}$ of degree at most l with $\mathfrak{A} \leftrightarrows_{r_n,t_n}^\infty \mathfrak{B}$, we have $\mathfrak{A} \equiv_n^\infty \mathfrak{B}$.

PROOF. One first shows that duplicator can force the FO$^\infty$-play from $(\mathfrak{A}, \mathfrak{B}, k + 1)$, where $\mathfrak{A}$ and $\mathfrak{B}$ are two $\sigma_{h+n-(k+1)}$-structures with $\mathfrak{A} \leftrightarrows_{r_{k+1},t_{k+1}}^\infty \mathfrak{B}$, into some position $((\mathfrak{A}, a), (\mathfrak{B}, b), k)$ with $(\mathfrak{A}, a) \leftrightarrows_{r_k,t_k}^\infty (\mathfrak{B}, b)$.

First suppose spoiler chooses in (1) to proceed with (2). More precisely, suppose he chooses an element $a \in A$. Then, since $\mathfrak{A} \leftrightarrows_{r_{k+1},t_{k+1}}^\infty \mathfrak{B}$, there exists $b \in B$ with r_{k+1}-Sph$(\mathfrak{A}, a) \cong r_{k+1}$-Sph$(\mathfrak{B}, b)$. We verify $(\mathfrak{A}, a) \leftrightarrows_{r_k,t_k}^\infty (\mathfrak{B}, b)$. First note that for all $a', a'' \in A$ and $b' \in B$, with r_k-Sph$(\mathfrak{A}, a, a') \cong r_k$-Sph$(\mathfrak{A}, a, a'') \cong r_k$-Sph$(\mathfrak{B}, b, b')$, we have $a' \in (2r_k + 1)$-Sph$(\mathfrak{A}, a)$ iff $a'' \in (2r_k + 1)$-Sph$(\mathfrak{A}, a)$ iff $b' \in (2r_k + 1)$-Sph$(\mathfrak{B}, b)$. Now let $a' \in A$ be arbitrary and set $\tau = r_k$-Sph$(\mathfrak{A}, a, a')$. We distinguish two cases.

(1) First suppose $a' \in (2r_k + 1)$-Sph$(\mathfrak{A}, a)$. Then

$$|\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a, a'') \cong \tau\}|$$

$$
\begin{aligned}
&= |\{a'' \in (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a, a'') \cong \tau\}| \\
&= |\{b'' \in (2r_k+1)\text{-Sph}(\mathfrak{B}, b) \mid r_k\text{-Sph}(\mathfrak{B}, b, b'') \cong \tau\}| \\
&= |\{b'' \in B \mid r_k\text{-Sph}(\mathfrak{B}, b, b'') \cong \tau\}|
\end{aligned}
$$

since the $(2r_k+1)$-spheres of $(\mathfrak{A}, a)$ and $(\mathfrak{B}, b)$ are isomorphic.

(2) Alternatively, let $a' \notin (2r_k+1)\text{-Sph}(\mathfrak{A}, a)$ and, for notational simplicity, $\tau' = r_k\text{-Sph}(\mathfrak{A}, a')$. Then we obtain

$$
\begin{aligned}
&|\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a, a'') \cong \tau\}| \\
&= |\{a'' \in A \setminus (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a, a'') \cong \tau\}| \\
&= |\{a'' \in A \setminus (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \\
&= |\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \\
&\quad - |\{a'' \in (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}|
\end{aligned}
$$

since $r_k\text{-Sph}(\mathfrak{A}, a, a'')$ for $a'' \notin (2r_k+1)\text{-Sph}(\mathfrak{A}, a)$ is completely determined by the r_k-spheres of $(\mathfrak{A}, a)$ and $(\mathfrak{A}, a'')$.

From $\mathfrak{A} \leftrightarrows^{\infty}_{r_{k+1},t_{k+1}} \mathfrak{B}$, we obtain

$$
|\{a'' \in A \mid r_{k+1}\text{-Sph}(\mathfrak{A}, a'') \cong \tau''\}| \sim^{\infty}_{t_{k+1}} |\{b'' \in B \mid r_{k+1}\text{-Sph}(\mathfrak{B}, b'') \cong \tau''\}|
$$

with $\tau'' = r_{k+1}\text{-Sph}(\mathfrak{A}, a')$. Since $\mathfrak{A}$ and $\mathfrak{B}$ are structures of finite degree over a finite signature, and since $r_k \leq r_{k+1}$, this implies

$$
|\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \sim^{\infty}_{t_{k+1}} |\{b'' \in B \mid r_k\text{-Sph}(\mathfrak{B}, b'') \cong \tau'\}| \,.
$$

Furthermore

$$
\begin{aligned}
&|\{a'' \in (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \\
&= |\{b'' \in (2r_k+1)\text{-Sph}(\mathfrak{B}, b) \mid r_k\text{-Sph}(\mathfrak{B}, b'') \cong \tau'\}| \\
&\leq (h+n-k) \cdot l^{2r_k+1} \,.
\end{aligned}
$$

Hence we obtain

$$
\begin{aligned}
&|\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \\
&\quad - |\{a'' \in (2r_k+1)\text{-Sph}(\mathfrak{A}, a) \mid r_k\text{-Sph}(\mathfrak{A}, a'') \cong \tau'\}| \\
&\sim^{\infty}_{t_k} |\{b'' \in B \mid r_k\text{-Sph}(\mathfrak{B}, b'') \cong \tau'\}| \\
&\quad - |\{b'' \in (2r_k+1)\text{-Sph}(\mathfrak{B}, b) \mid r_k\text{-Sph}(\mathfrak{B}, b'') \cong \tau'\}|
\end{aligned}
$$

which, as above, equals the number of elements $b'' \in B$ with $r_k\text{-Sph}(\mathfrak{B}, b, b'') \cong \tau$.

Thus, in all cases, we showed

$$
|\{a'' \in A \mid r_k\text{-Sph}(\mathfrak{A}, a, a'') \cong \tau\}| \sim^{\infty}_{t_k} |\{b'' \in B \mid r_k\text{-Sph}(\mathfrak{B}, b, b'') \cong \tau\}|
$$

which implies $(\mathfrak{A}, a) \leftrightarrows^{\infty}_{r_k,t_k} (\mathfrak{B}, b)$ as required.

Now suppose spoiler chooses in (1) to proceed with (2'), more precisely, he chooses an infinite set $X \subseteq A$. Then duplicator chooses

$$Y = \{b \in B \mid \exists a \in X : r_{k+1}\text{-Sph}(\mathfrak{A}, a) \cong r_{k+1}\text{-Sph}(\mathfrak{B}, b)\} \ .$$

In step (4'), spoiler chooses some $b \in Y$. Then, by the choice of Y, duplicator can answer with some $a \in X$ satisfying r_{k+1}-Sph$(\mathfrak{A}, a) \cong r_{k+1}$-Sph$(\mathfrak{B}, b)$. Then $(\mathfrak{A}, a) \leftrightarrows^{\infty}_{r_k,t_k} (\mathfrak{B}, b)$ follows as above.

By induction, duplicator can force any play from $(\mathfrak{A}, \mathfrak{B}, n)$ with $\mathfrak{A} \leftrightarrows^{\infty}_{r_n,t_n} \mathfrak{B}$ into a position $(\mathfrak{A}', \mathfrak{B}', 0)$ with $\mathfrak{A}' \leftrightarrows^{\infty}_{0,0} \mathfrak{B}'$. Let $a \in A$ be arbitrary. Then, since $\mathfrak{A}' \leftrightarrows^{\infty}_{0,0} \mathfrak{B}'$, there exists $b \in B$ with 0-Sph$(\mathfrak{A}', a) \cong 0$-Sph$(\mathfrak{B}', b)$ implying 0-Sph$(\mathfrak{A}') \cong 0$-Sph$(\mathfrak{B}')$. Note that 0-Sph$(\mathfrak{A}')$ and 0-Sph$(\mathfrak{B}')$ are the restriction of $\mathfrak{A}'$ and $\mathfrak{B}'$ to their constants. Hence the game position $(\mathfrak{A}', \mathfrak{B}', 0)$ is winning for duplicator. $\Box$

A formula ψ with one free variable x is *local* if there exists $r \geq 0$ such that any subformula of the form $\exists y \alpha$ is of the form $\exists y (d(x, y) \leq r \wedge \beta)$. As a consequence of Theorem 11, we obtain a normal form for FO$^{\infty}$ formulas:

Corollary 12 *Let $l \geq 0$ and φ be a formula from* FO$^{\infty}(\sigma)$ *without free variables. Then there exists a positive Boolean combination α of formulas of the form*

$$\exists^{=t} x\, \psi(x) \ \textit{ and } \ \exists^{>t} x\, \psi(x) \ \textit{ and } \ \exists^{<\infty} x\, \psi(x) \ \textit{ and } \ \exists^{\infty} x\, \psi(x)$$

with $\psi \in$ FO(σ) *local such that for all σ-structures $\mathfrak{A}$ of degree at most l, we have*

$$\mathfrak{A} \models \varphi \iff \mathfrak{A} \models \alpha \ .$$

If φ is an FO *formula, then the Boolean combination α contains only formulas of the form $\exists^{=t} x\, \psi(x)$ and $\exists^{>t} x\, \psi(x)$ (Hanf's theorem [12]).*

PROOF. Let n be the quantifier rank of the formula φ. There are only finitely many isomorphism types of structures of the form r_n-Sph$(\mathfrak{B}, b)$ for $\mathfrak{B}$ a σ-structure of degree at most l and $b \in \mathfrak{B}$. For every such structure, there is a local first-order formula ψ with $\mathfrak{A} \models \psi(a)$ iff r_n-Sph$(\mathfrak{B}, b) \cong r_n$-Sph$(\mathfrak{A}, a)$. Now every $\leftrightarrows^{\infty}_{r_n,t_n}$-equivalence class can be described by a Boolean combination as required. Since there are only finitely many such equivalence classes, the result follows. $\Box$

Note that our proof is not constructive, i.e., we give no effective construction of the Boolean combination α. The same applies to the proofs of Hanf's theorem that can be found, e.g., in [7, 16]. Differently, the original proof by Hanf was effective. We leave it as an open question whether also the above corollary can be given a constructive proof.

4 Message-Passing Automata and Logics

This section relates the expressive power of all types of MPAs and the extended logic. Let σ denote the purely relational signature consisting of binary relation symbols E_ℓ for $\ell \in C$ and the unary relation symbols R_a for $a \in \Sigma$. Then every MSC is a σ-structure. As expected, we will write the formula $R_a(x)$ as $\lambda(x) = a$. Moreover, we write EMSO$^\infty$ for EMSO$^\infty(\sigma)$, and FO$^\infty$ etc. are to be understood similarly.

Example 13 *The* FO$^\infty$*-formula* $\exists^\infty x\, (\lambda(x) = \text{Client!Server})$ *expresses that* Client *sends infinitely many messages to* Server. *Observe that we cannot do without the infinity quantifier to express this property, which can be easily shown using Hanf's Theorem. Moreover, the* FO*-formula* $\forall x((\bigvee_{\sigma \in \Sigma_{\text{Client}}} \lambda(x) = \sigma) \rightarrow \exists y E_{\text{Client}}(x, y))$ *is satisfied by all those MSCs in which* Client *executes infinitely many actions.*

MPAs can be used to compute the sphere around any node of an MSC. This feature, described formally in the following proposition, is the key connection between these automata and the logical characterization of first-order expressible properties.

Proposition 14 (cf. [1]) *Let* $r \in \mathbb{N}$. *There are a termination-detecting Muller/ termination-detecting Staiger-Wagner MPA* $\mathcal{A}_r = (((Q_i, \Delta_i))_{i \in Ag}, \mathcal{D}, \iota, \mathcal{F})$ *and a mapping* η *from* $\bigcup_{i \in Ag} Q_i$ *into the set of* σ_1*-structures such that* $L(\mathcal{A}_r)$ *is the set of all MSCs and, for any MSC* $M = (V, \{E_\ell\}_{\ell \in C}, \lambda)$*, for any accepting run* (ρ, μ) *of* $\mathcal{A}_r$ *on* M*, and for any* $u \in V$*, we have* $\eta(\rho(u)) = r$-Sph(M, u).

Note that, at some point, the construction from [1] makes use of the argument that an MSC is finite. To be applicable to our setting, however, this argument can be replaced by the fact that the *past* of any event is finite.

4.1 Termination-Detecting Muller MPAs and Logic

A σ_1-structure S is an *r-sphere in some MSC* if there exists an MSC M and a vertex v of M with $S \cong r$-Sph(M, v). For an MSC M' and an r-sphere in some MSC S, let $|M'|_S$ denote the number of vertices v of M' with $S \cong r$-Sph(M', v).

Lemma 15 *Let* $r \in \mathbb{N}$, $t \in \mathbb{N} \cup \{\infty\}$*, and* S *be some* r*-sphere in some MSC. There exist termination-detecting Muller MPAs recognizing the sets of MSCs* M *with* $|M|_S = t$ *and* $t < |M|_S < \infty$*, respectively.*

PROOF. In all cases, one starts from the termination-detecting Muller MPA $\mathcal{A}_r = (((Q_i, \Delta_i))_{i \in Ag}, \mathcal{D}, \iota, \mathcal{F})$ and the function η from Prop. 14. Let the only constant from S be labeled by some letter from Σ_i.

To detect $|M|_S = \infty$, we just keep those accepting tuples $(F_j, \theta_j)_{j \in Ag}$ from $\mathcal{F}$ that satisfy $\theta_i = \infty$ and $S \in \eta(F_i)$.

To detect $t < |M|_S < \infty$ for $t \in \mathbb{N}$, we extend the states of $\mathcal{A}_i$ with a counter that counts the number of realizations of S up to $t + 1$, i.e., the new local state space of agent i is $Q_i \times \{0, \dots, t + 1\}$ with initial state $(\iota[i], 0)$. To distinguish "at least $t + 1$" from "infinitely many" realizations of S, the acceptance condition is the set of tuples $(F_j, \theta_j)_{j \in Ag}$ such that $\theta_i = \overline{\infty}$ implies $F_i \subseteq Q_i \times \{t+1\}$, $\theta_i = \infty$ implies $F_i \subseteq (Q_i \setminus \eta^{-1}(S)) \times \{t + 1\}$, and $(\pi_1(F_j), \theta_j)_{j \in Ag} \in \mathcal{F}$.

To detect $|M|_S = t < \infty$, we use the same states and transitions, but the acceptance condition now requires $F_i \subseteq Q_i \times \{t\}$. $\square$

Theorem 16 *Let L be a set of MSCs. Then, the following are equivalent:*

(1) there exists a termination-detecting Muller MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$.
(2) there exists an EMSO$^\infty$ *sentence φ such that $L = \{M \mid M \models \varphi\}$.*

PROOF. By Theorem 8, it is sufficient to translate a Büchi MPA into an equivalent EMSO$^\infty$ sentence. This construction follows similar instances of that problem, e.g., [6]. Second order variables X_q for $q \in \mathcal{D} \cup \bigcup_{i \in Ag} Q_i$ encode an assignment of messages and states to vertices. The first-order part then expresses that this assignment is a run. In addition, we have to take care of the acceptance condition. Any such condition $\overline{q} \in \prod_{i \in Ag} 2^{Q_i}$ is translated into the conjunction of the following formulas for any $i \in Ag$

$$\bigvee_{q \in \overline{q}[i]} \left(\begin{array}{l} \quad \exists^\infty x (x \in X_q \wedge \lambda(x) \in \Sigma_i) \\ \vee \ \exists x (x \in X_q \wedge \lambda(x) \in \Sigma_i \wedge \neg \exists y (E_i(x, y))) \end{array} \right)$$

(supplemented by $\cdots \vee \forall x \neg \lambda(x) \in \Sigma_i$ if $\iota[i] \in \overline{q}[i]$). The kernel of this formula expresses that the state q is assumed infinitely often by process i or, alternatively, it is assumed by the last event of this process.

Consider the other implication. Since termination-detecting Muller MPAs are closed under projection, it suffices to consider the case $\varphi \in$ FO$^\infty$. By Cor. 12, we can assume φ to be a positive Boolean combination of formulas of the form $\exists^{=t} x\, \psi$, $\exists^{>t} x\, \psi$, and $\exists^\infty x\, \psi$ with ψ local. Note that validity of any of these basic formulas can be checked by a termination-detecting Muller MPA due to Lemma 15. Now the result follows since the class of languages accepted by termination-detecting Muller MPAs is closed under finite union and intersection.

The number of states of the termination-detecting Muller MPA $\mathcal{A}$ constructed from a given EMSO$^\infty$-formula φ is elementary in the size of the formula φ: In Cor. 12, both the radius r and the threshold t are bounded elementarily in the length of the formula φ. We only remark that the number of states of the MPA from Prop. 14 is also elementary in r and t. But it is not clear whether $\mathcal{A}$ can be constructed effectively and, if so, in elementary time. The reason is that the above proof is based on Cor. 12. If φ is from EMSO, then we can rely on Hanf's effective proof. Hence, in that case, the automaton $\mathcal{A}$ can be constructed effectively and, as an inspection of Hanf's proof reveals, in elementary time. In particular, this applies in the setting of [1] where only finite MSCs and the logic EMSO are considered. The following section shows that for the logic EMSO, we do not need the expressive power of Muller MPAs.

4.2 Staiger-Wagner MPAs and Logic

The following lemma describes the counting power of Staiger-Wagner MPAs: As far as finite counting is concerned, termination-detecting Staiger-Wagner MPAs can do as much as termination-detecting Muller MPAs. Similarly to Lemma 15, we can show the following:

Lemma 17 *Let $r, t \in \mathbb{N}$ and let S be some r-sphere in some MSC. There exist termination-detecting Staiger-Wagner MPAs that recognize the sets of MSCs M with $|M|_S = t$ and $t < |M|_S$, respectively.*

PROOF. The proof differs only slightly from that of Lemma 15: one again starts from the MPA $\mathcal{A}_r$ and the mapping η, extends the states with a counter, and defines the transition relation and the initial states as there. But the acceptance condition $\mathcal{F}$ now contains all tuples $\overline{q}$ such that $(q, t) \in \overline{q}[i]$ for some $q \in Q_i$. $\square$

Theorem 18 *Let L be a set of MSCs. Then the following are equivalent:*

(1) there exists a termination-detecting Staiger-Wagner MPA $\mathcal{A}$ such that $L = L(\mathcal{A})$.
(2) there exists an EMSO *sentence φ such that $L = \{M \mid M \models \varphi\}$.*

PROOF. The proof is similar to the proof of Theorem 16. The only difference in the transformation of an automaton into a formula concerns the acceptance condition, which, this time, is given as (a set of) function(s) $\overline{q} : Ag \to 2^Q$. It is expressed as a conjunction of the following conjunct for any $i \in Ag$:

$$\bigwedge_{q \in \overline{q}[i]} \exists x (x \in X_q \wedge \lambda(x) \in \Sigma_i) \;\wedge \bigwedge_{q \in Q \setminus \overline{q}[i]} \forall x (\lambda(x) \in \Sigma_i \to x \notin X_q)$$

For the other transformation, we use Hanf's theorem [12] instead of Cor. 12 and Lemma 17 instead of Lemma 15. □

References

[1] B. Bollig and M. Leucker. Message-passing automata are expressively equivalent to EMSO logic. *Theoretical Computer Science*, 358(2-3):150–172, 2006.

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

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

[4] Y. Choueka. Theories of automata on ω-tapes: a simplified approach. *Journal of Computer and System Sciences*, 8:117–141, 1974.

[5] V. Diekert and A. Muscholl. Deterministic asynchronous automata for infinite traces. *Acta Informatica*, 31:379–397, 1994.

[6] M. Droste, P. Gastin, and D. Kuske. Asynchronous cellular automata for pomsets. *Theoretical Computer Science*, 247(1-2):1–38, 2000.

[7] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1991.

[8] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.

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

[10] P. Gastin and A. Petit. Asynchronous cellular automata for infinite traces. In *Proceedings of ICALP 1992*, volume 623 of *LNCS*, pages 583–594. Springer, 1992.

[11] B. Genest, D. Kuske, and A. Muscholl. A Kleene theorem and model checking algorithms for existentially bounded communicating automata. *Information and Computation*, 204(6):920–956, 2006.

[12] W. 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*, 202(1):1–38, 2005.

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

[15] D. Kuske. Regular sets of infinite message sequence charts. *Information and Computation*, 187:80–109, 2003.

[16] L. Libkin. *Elements of Finite Model Theory*. Springer, Berlin, 2004.

[17] L. D. Lipner. *Some aspects of generalized quantifiers*. PhD thesis, Berkeley, 1970.

[18] A. B. Slomson. Generalized quantifiers and well orderings. *Arch. math. Logik*, 15:57–73, 1972.

[19] W. Thomas. On logical definability of trace languages. In *Proc. of Workshop of the ESPRIT Basic Research Action No 3166: Algebraic and Syntactic Methods in Computer Science (ASMICS), Kochel am See, Germany (1989)*, Report TUM-I9002, TU Munich, 1990.

[20] W. Thomas. Languages, automata and logic. In *Handbook of Formal Languages*, volume 3, Beyond Words, pages 389–455. Springer, Berlin, 1997.

[21] S. Vinner. A generalization of Ehrenfeucht's game and some applications. *Israel Journal of Mathematics*, 12:279–286, 1972.