Monitor Logics for Quantitative Monitor Automata

Erik Paul

January 12, 2020

1 Introduction

In the last years, there has been increasing interest in quantitative features of the specification and analysis of systems. Such quantitative aspects include the consumption of a certain resource or the output of a benefit. Both weighted automata and weighted logics [7] are means to achieve this quantitative description of systems. They can be employed for both finite and infinite input.

Very recently, Chatterjee et al. introduced a new automaton model operating on infinite words [4]. *Quantitative Monitor Automata* are equipped with a finite number of monitor counters. At each transition, a counter can be started, terminated, or the value of the counter can be increased or decreased. The term “monitor” stems from the fact that the values of the counters do not influence the behavior of the automaton. The values of the counters when they are terminated provide an infinite sequence of weights, which is evaluated into a single weight using a *valuation function*.

Quantitative Monitor Automata possess several interesting features. They are expressively equivalent to a subclass of *Nested Weighted Automata* [3], an automaton model which for many valuation functions has decidable emptiness and universality problems. Quantitative Monitor Automata are also very expressive. As an example, imagine a storehouse with a resource which is restocked at regular intervals. Between restocks, demands can remove one unit of this resource at a time. Such a succession of restocks and demands can be modeled as an infinite sequence over the alphabet $\{\mathsf{restock}, \mathsf{demand}\}$. Interesting quantitative properties of such a sequence include the long-term average demand, the minimum demand, and the maximum demand between restocks. These properties can be described using Quantitative Monitor Automata. At every restock a counter is started, counting the number of demands until the next restock. An appropriate valuation function then computes the desired property. For the average demand, this can be achieved with the *Cesàro mean* which was introduced to automata theory by Chatterjee et al. in [2]. Note that behaviors like these cannot be modeled using *weighted Büchi-automata* [12, 13] or their extension with valuation functions [8]. In the latter model, the Cesàro mean of every weight-sequence is bounded by the largest transition weight in the automaton. This is not the case for Quantitative Monitor Automata.

In this paper, we develop a logic which is expressively equivalent to Quantitative Monitor Automata. Our main results are the following.

- We introduce a new logic which we call *Monitor Logic*.
- We show that this Monitor Logic is expressively equivalent to Quantitative Monitor Automata.
- We show various closure properties of Quantitative Monitor Automata and prove that Muller and Büchi acceptance conditions provide the same expressive power.

The relationship between automata and logics plays a large role in specification and verification. Statements are often easier to formulate in the form of a logic formula rather than directly as an automaton. Consequently, the fundamental Büchi-Elgot-Trakhtenbrot Theorem [1, 11, 16], which established the coincidence of regular languages with languages definable in monadic second order logic, has found use in many areas of application. An extension to semiring weighted automata was later given by Droste and Gastin [6].

Our logic is equipped with three quantifiers. A sum quantifier to handle the computations on the counters, a valuation quantifier to handle the valuation, and a third quantifier to combine the weights of all runs on a word. Our biggest challenge was to find appropriate restrictions on the use of the quantifiers. Without any restrictions, the logic would be too powerful, which we also formally prove using counter examples. The most important result of our considerations is that the computations of the sum quantifier should depend on an MSO-definable condition.

We note that our constructions are effective. Given a formula from our logic, we can effectively construct a Quantitative Monitor Automaton describing this formula. Conversely, for every automaton we can effectively construct a formula whose semantics coincides with the behavior of the automaton.

2 Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the natural numbers, $\mathbb{Z}$ the integers, and $\mathbb{R}$ the reals. For a set X, we denote the power set of X by $\mathcal{P}(X)$. For two sets X and Y and a mapping $f\colon X \to Y$, we call X the *domain* of f, denoted by $\mathrm{dom}(f)$. For a subset $X' \subseteq X$, the *restriction of f to X'*, denoted by $f{\restriction}_{X'}$, is the mapping $f{\restriction}_{X'}\colon X' \to Y$ defined by $f{\restriction}_{X'}(x) = f(x)$ for every $x \in X'$. For a second mapping $g\colon X \to Y$, we write $f = g$ if for all $x \in X$ we have $f(x) = g(x)$. An *alphabet* Σ is a finite non-empty set. An infinite word over Σ is a sequence $w = a_0a_1a_2\ldots$ from Σ. The set of infinite words over Σ is denoted by Σ^ω. The set of finite words Σ^* over Σ is defined as the set of finite sequences $a_0a_1\ldots a_n$ from Σ. The empty word is denoted by ε. A mapping $\Sigma^\omega \to \mathbb{R} \cup \{\infty\}$ is called a *series*.

A *(non-deterministic) Muller automaton* (NMA) over Σ is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ where (1) Q is a finite set (of states), (2) Σ is an alphabet, (3) $q_0 \in Q$ is the initial state, (4) $\delta \subseteq Q \times \Sigma \times Q$ is the set of transitions, and (5) $\mathcal{F} \subseteq \mathcal{P}(Q)$ is the set of final sets.

Let $w = a_0a_1\ldots \in \Sigma^\omega$ be an infinite word. A *run of $\mathcal{A}$ on w* is an infinite sequence of transitions $r = (d_i)_{i\geq 0}$ so that $d_i = (q_i, a_i, q_{i+1}) \in \delta$ for all $i \geq 0$. We denote by $\mathrm{In}^Q(r)$ the set of states which appear infinitely many times in r, i.e.,

$$\mathrm{In}^Q(r) = \{q \in Q \mid \forall i \exists j \geq i\colon d_j = (q, a_j, q_{j+1})\}.$$

A run r of $\mathcal{A}$ on $w \in \Sigma^\omega$ is called *accepting* if $\mathrm{In}^Q(r) \in \mathcal{F}$, that is, if the states which appear infinitely many times in r form a set in $\mathcal{F}$. In this case, we say that w is *recognized (accepted) by $\mathcal{A}$*. The set of accepting runs on a word $w \in \Sigma^\omega$ is denoted by $\mathrm{Acc}_{\mathcal{A}}(w)$. The *infinitary language* of $\mathcal{A}$, denoted by $\mathcal{L}_\omega(\mathcal{A})$, is the set of all infinite words that are accepted by $\mathcal{A}$. A language $L \subseteq \Sigma^\omega$ is called ω-recognizable if there exists a Muller automaton $\mathcal{A}$ so that $L = \mathcal{L}_\omega(\mathcal{A})$.

3 Quantitative Monitor Automata

An *ω-valuation function* is a mapping $\mathrm{Val}\colon \mathbb{Z}^{\mathbb{N}} \to \mathbb{R} \cup \{-\infty, \infty\}$ which assigns real values, $-\infty$, or ∞ to infinite sequences of integers. Typical examples of such functions are the Cesàro mean $\textsc{Ces}((z_i)_{i\geq 0}) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=0}^{n-1} z_i$, the supremum $\textsc{Sup}((z_i)_{i\geq 0}) = \sup_{i\geq 0} z_i$, the infimum $\textsc{Inf}((z_i)_{i\geq 0}) = \inf_{i\geq 0} z_i$, the limit superior $\textsc{LimSup}((z_i)_{i\geq 0}) = \limsup_{i\to\infty} z_i$, and the limit inferior $\textsc{LimInf}((z_i)_{i\geq 0}) = \liminf_{i\to\infty} z_i$.

For a new symbol $\mathbb{1}$ and an ω-valuation function Val, we extend the domain of Val to sequences $(z_i)_{i\geq 0}$ from $\mathbb{Z} \cup \{\mathbb{1}\}$ as follows. If at some point $(z_i)_{i\geq 0}$ becomes constantly $\mathbb{1}$, we let $\mathrm{Val}((z_i)_{i\geq 0}) = \infty$. Otherwise, we let $(z_{i_k})_{k\geq 0}$ be the subsequence of $(z_i)_{i\geq 0}$ which contains all elements which are not $\mathbb{1}$ and define $\mathrm{Val}((z_i)_{i\geq 0}) = \mathrm{Val}((z_{i_k})_{k\geq 0})$.

We recall Quantitative Monitor Automata as introduced in [4]. We use a different name, however, in order to distinguish between Büchi and Muller acceptance conditions.

A *Büchi automaton with monitor counters* (BMCA) $\mathcal{A}$ is a tuple $(Q, \Sigma, I, \delta, F, n, \mathrm{Val})$ where (1) Q is a finite set (of states), (2) Σ is an alphabet, (3) $I \subseteq Q$ is the set of initial states, (4) δ is a finite subset of $Q \times \Sigma \times Q \times (\mathbb{Z} \cup \{s, t\})^n$, called the transition relation, such that for every $(p, a, q, \bar{u}) \in \delta$ at most one component of $\bar{u}$ contains s, (5) F is the set of accepting states, (6) $n \geq 1$ is the number of counters, and (7) Val is an ω-valuation function.

Intuitively, the meaning of a transition $(p, a, q, \bar{u})$ is that if the automaton is in state p and reads an a, it can move to state q and either (1) start (or activate) counter j if $u_j = s$, or (2) add u_j to the current value of counter j if this counter is active and $u_j \in \mathbb{Z}$, or (3) stop (or deactivate) counter j if $u_j = t$, for $j = 1, \ldots, n$. Initially, all counters are inactive. We will also call $\mathcal{A}$ an n-BMCA or a Val-BMCA, thereby stressing the number of counters or the ω-valuation function used.

Let $a_0a_1\ldots \in \Sigma^\omega$ be an infinite word. A *run of $\mathcal{A}$ on w* is an infinite sequence of transitions $r = (d_i)_{i\geq 0}$ so that $d_i = (q_i, a_i, q_{i+1}, \bar{u}^i) \in \delta$ for all $i \geq 0$. A run r of $\mathcal{A}$ on $w \in \Sigma^\omega$ is called *accepting* if (1) $q_0 \in I$, (2) $\mathrm{In}^Q(r) \cap F \neq \emptyset$, (3) if $u^i_j = s$ for some $i \geq 0$, then there exists $l > i$ such that $u^l_j = t$ and for all $k \in \{i+1, \ldots, l-1\}$ we have $u^k_j \in \mathbb{Z}$, (4) if $u^i_j = t$ for some $i \geq 0$, then there exists $l < i$ such that $u^l_j = s$ and for all $k \in \{l+1, \dots, i-1\}$ we have $u^k_j \in \mathbb{Z}$, and (5) infinitely often some counter is activated, i.e.,

$$\{i \geq 0 \mid u^i_j = s \text{ for some } j\}$$

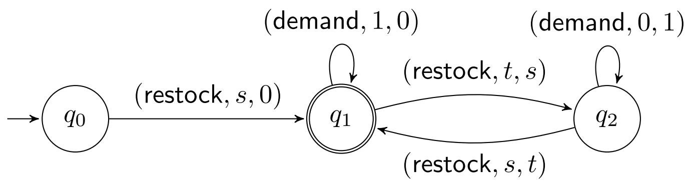
is an infinite set. The set of accepting runs on a word $w \in \Sigma^\omega$ is denoted by $\mathrm{Acc}_{\mathcal{A}}(w)$.

An accepting run r defines a sequence $(z_i)_{i \geq 0}$ from $\mathbb{Z} \cup \{\mathbb{1}\}$ as follows. If $u^i_j = s$ for some $j \in \{1, \dots, n\}$ and $l > i$ is such that $u^l_j = t$ and for all $k \in \{i+1, \dots, l-1\}$ we have $u^k_j \in \mathbb{Z}$, then $z_i = \sum_{k=i+1}^{l-1} u^k_j$. If $u^i_j \neq s$ for all $j \in \{1, \dots, n\}$, then $z_i = \mathbb{1}$. We also call $(z_i)_{i \geq 0}$ the *weight-sequence associated to* r. The weight of the run r is defined as $\mathrm{Val}(r) = \mathrm{Val}((z_i)_{i \geq 0})$. The *behavior* of the automaton $\mathcal{A}$ is the series $[\![\mathcal{A}]\!] \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ defined by $[\![\mathcal{A}]\!](w) = \inf_{r \in \mathrm{Acc}_{\mathcal{A}}(w)} \mathrm{Val}(r)$, where the infimum over the empty set is defined as ∞. A series $S \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ is called *MC-recognizable* if there exists a BMCA $\mathcal{A}$ such that $[\![\mathcal{A}]\!] = S$. The notions of *n-MC-recognizable* and Val-*MC-recognizable* are defined likewise.

A *Muller automaton with monitor counters* (MMCA) is defined like a BMCA, but instead of a set of accepting states we have a set of accepting sets $\mathcal{F} \subseteq \mathcal{P}(Q)$. The condition (2) for a run r on a word $w \in \Sigma^\omega$ to be accepting is then replaced by $\mathrm{In}^Q(r) \in \mathcal{F}$, i.e., a Muller acceptance condition.

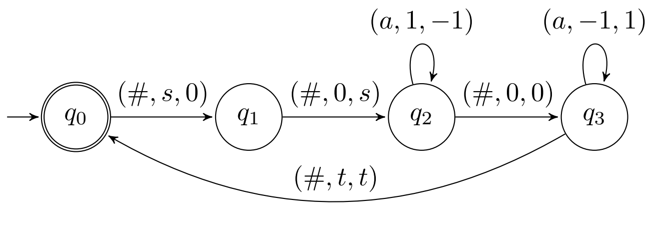
Büchi automata with monitor counters use a Büchi acceptance condition, i.e., at least one accepting state has to appear infinitely often. In Lemma 3 we show that using a Muller acceptance condition does not influence the expressive power.

Example 1. Consider the alphabet $\Sigma = \{\mathsf{demand}, \mathsf{restock}\}$ with the ω-valuation function $\mathrm{Val} = \textsc{Ces}$. We model a storehouse with some sort of supply which is restocked whenever restock is encountered, and one unit of the supply is removed at every demand. Given an infinite sequence of restocks and demands, we are interested in the long-time average number of demands between restocks. Under the assumption that every such sequence starts with a restock, this behavior is modeled by the following automaton with two monitor counters.



When for the valuation function we take Inf or Sup, the automaton above describes the lowest or highest demand ever encountered, for the latter assuming that the numbers of demands are bounded.

Example 2 ([4])**.** Consider the alphabet $\Sigma = \{a, \#\}$ and the language L consisting of words $(\#^2 a^* \# a^* \#)^\omega$. On these words, we consider the quantitative property "the maximal block-length difference between even and odd positions", i.e., the value of the word $\#\# a^{m_1} \# a^{m_2} \#\#\# \dots$ shall be $\sup_{i \geq 1} |m_{2i-1} - m_{2i}|$. With the choice $\mathrm{Val} = \textsc{Sup}$, a BMCA realizing this behavior is the following.



Each $(\#^2 a^{m_1} \# a^{m_2} \#)$-block is processed by starting both counters on the first two $\#$'s, accumulating m_1 into the first counter and accumulating $-m_1$ into the second, reading $\#$, then accumulating $m_1 - m_2$ into the first counter and $-m_1 + m_2$ into the second, and finally terminating both counters on the last $\#$. Thus, the associated weight-sequence for only this block is $(m_1 - m_2, -m_1 + m_2, \mathbb{1}, \dots, \mathbb{1})$. Clearly, the final value of counter 1 is always the negative of the final value in counter 2. Since our ω-valuation function is Sup, only the positive counter value actually plays a role in the value assigned to the whole word, and this positive value is $|m_1 - m_2|$.

In the rest of this section, we prove various closure properties for automata with monitor counters and that BMCA and MMCA have the same expressive power.

Lemma 3. *Büchi automata with monitor counters are expressively equivalent to Muller automata with monitor counters.*

Proof. The proof is similar to the standard construction to show that Büchi automata are expressively equivalent to Muller automata, see for example [9].

Let $\mathcal{A} = (Q, \Sigma, I, \delta, F, n, \mathrm{Val})$ be a BMCA, we define the MMCA $\mathcal{A}' = (Q, \Sigma, I, \delta, \mathcal{F}, n, \mathrm{Val})$ by $\mathcal{F} = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$. Then on every word w, the accepting runs of $\mathcal{A}$ on w coincide with the accepting runs of $\mathcal{A}'$ on w, i.e., $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$.

Conversely, let $\mathcal{A} = (Q, \Sigma, I, \delta, \mathcal{F}, n, \mathrm{Val})$ be an MMCA. We construct a BMCA $\mathcal{A}' = (Q', \Sigma, I', \delta', F', n, \mathrm{Val})$ as follows.

$$\begin{aligned}
Q' &= Q \cup (Q \times \mathcal{F} \times \mathcal{P}(Q))\\
I' &= I \cup \{(q, F, \{q\}) \mid F \in \mathcal{F}, q \in I \cap F\}\\
F' &= \{(q, F, F) \mid F \in \mathcal{F}, q \in F\}\\
\delta' &= \quad \delta\\
&\quad \cup \{(p, a, (q, F, \{q\}), \bar{u}) \mid F \in \mathcal{F}, p \in Q \setminus F, q \in F, a \in \Sigma, (p, a, q, \bar{u}) \in \delta\}\\
&\quad \cup \{((p, F, R), a, (q, F, R \cup \{q\}), \bar{u}) \mid F \in \mathcal{F}, p, q \in F, R \subsetneq F, a \in \Sigma, (p, a, q, \bar{u}) \in \delta\}\\
&\quad \cup \{((p, F, F), a, (q, F, \{q\}), \bar{u}) \mid F \in \mathcal{F}, p, q \in F, a \in \Sigma, (p, a, q, \bar{u}) \in \delta\}
\end{aligned}$$

We let $\pi\colon Q' \to Q$ be the projection defined by $q \mapsto q$ and $(q, F, R) \mapsto q$ for $(q, F, R) \in Q \times \mathcal{F} \times \mathcal{P}(Q)$. We extend π to transitions by $(p', a, q', \bar{u}) \mapsto (\pi(p'), a, \pi(q'), \bar{u})$ and to sequences of transitions from $(\delta')^\omega$ in the obvious way. We claim that for every $w \in \Sigma^\omega$, we have a surjection $\pi\colon \mathrm{Acc}_{\mathcal{A}'}(w) \to \mathrm{Acc}_{\mathcal{A}}(w)$, and that for every $r' \in \mathrm{Acc}_{\mathcal{A}'}(w)$ the weight-sequences associated to r' and $\pi(r')$ are the same.

First, let $r' = (d'_i)_{i \geq 0}$ with $d'_i = (q'_i, a, q'_{i+1}, \bar{u}^i)$ be an accepting run of $\mathcal{A}'$ on w. Then for some $F \in \mathcal{F}$ and $q \in F$, the state (q, F, F) occurs infinitely often. By construction of δ', this means that there exists $i \geq 0$ such that for all $j \geq i$, we have $\pi(q'_j) \in F$, and for every $p \in F$, there are infinitely many j such that $\pi(q'_j) = p$. Thus, $\mathrm{In}^Q(\pi(r')) = F$ and we have $\pi(r') \in \mathrm{Acc}_{\mathcal{A}}(w)$. It is also easy to see that the weight-sequences of r' and $\pi(r')$ coincide.

Now let $r = (d_i)_{i \geq 0}$ with $d_i = (q_i, a_i, q_{i+1}, \bar{u}^i)$ be an accepting run of $\mathcal{A}$ on w and $F = \mathrm{In}^Q(r)$. Then either all q_i are in F, or there is an $i \geq 0$ with $q_i \notin F$ and for all $j > i$, $q_j \in F$.

In the first case, we let $q'_0 = (q_0, F, \{q_0\})$, otherwise we let $q'_j = q_j$ for $j \leq i$ and $q'_{i+1} = (q_{i+1}, F, \{q_{i+1}\})$. Then assuming that $q'_j = (q_j, F, R)$ for $j > i$ is already defined, we let $q'_{j+1} = (q_{j+1}, F, R \cup \{q_{j+1}\})$ if $R \subsetneq F$, and otherwise if $R = F$ we let $q'_{j+1} = (q_{j+1}, F, \{q_{j+1}\})$. Then with $d'_i = (q'_i, a_j, q'_{i+1}, \bar{u}^i)$, the sequence $r' = (d'_i)_{i \geq 0}$ is an accepting run of $\mathcal{A}'$ on w.

In conclusion, we have $\inf_{r' \in \mathrm{Acc}_{\mathcal{A}'}(w)} \mathrm{Val}(r') = \inf_{r \in \mathrm{Acc}_{\mathcal{A}}(w)} \mathrm{Val}(r)$ for all $w \in \Sigma^\omega$, which means $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. □

In the next lemma, we show that MC-recognizable series are closed under projections and their preimage. Given two alphabets Σ and Γ and a mapping $h\colon \Sigma \to \Gamma$, and thus a homomorphism $h\colon \Sigma^\omega \to \Gamma^\omega$, we define for every $S\colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ the *projection* $h(S)\colon \Gamma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ by

$$h(S)(w) = \inf\{S(v) \mid h(v) = w\}$$

for every $w \in \Gamma^\omega$. Moreover, if $S'\colon \Gamma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$, then we define $h^{-1}(S') = S' \circ h$, i.e., $h^{-1}(S')\colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$, $w \mapsto S'(h(w))$.

Lemma 4. *Let* Σ *and* Γ *be two alphabets,* $h\colon \Sigma \to \Gamma$ *be a mapping, and* Val *be an* ω*-valuation function.*

(i) If $S\colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ *is* Val*-MC-recognizable, then the projection* $h(S)\colon \Gamma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ *is also* Val*-MC-recognizable.*

(ii) If $S'\colon \Gamma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ *is* Val*-MC-recognizable, then* $h^{-1}(S')\colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ *is also* Val*-MC-recognizable.*

Proof. We apply an idea also used in [10].

(i) Let $\mathcal{A}_S = (Q_S, \Sigma, I_S, \delta_S, F_S, n_S, \mathrm{Val})$ be a Val-BMCA over Σ with $[\![\mathcal{A}_S]\!] = S$. We construct a new Val-BMCA $\mathcal{A} = (Q, \Gamma, I, \delta, F, n_S, \mathrm{Val})$ over Γ with $[\![\mathcal{A}]\!] = h(S)$ as follows.

- $Q = Q_S \times \Sigma$, $I = I_S \times \{a_0\}$ for some fixed $a_0 \in \Sigma$, $F = F_S \times \Sigma$, and
- $((p, a), b, (p', a'), \bar{u}) \in \delta$ if and only if $h(a') = b$ and $(p, a', p', \bar{u}) \in \delta_S$.

Then $r = ((q_0, a_0), b_1, (q_1, a_1), \bar{u}^1)((q_1, a_1), b_2, (q_2, a_2), \bar{u}^2)\ldots$ is a run of $\mathcal{A}$ on $b_1 b_2 \ldots$ if and only if $h(a_1 a_2 \ldots) = b_1 b_2 \ldots$ and $r_S = (q_0, a_1, q_1, \bar{u}^1)(q_1, a_2, q_2, \bar{u}^2)\ldots$ is a run of $\mathcal{A}_S$ on $a_1 a_2 \ldots$. Moreover, r is accepting if and only if $q_0 \in I_S$, at least one $q \in F_S$ appears infinitely often in the first component of the states, and the conditions (3), (4), and (5) concerning the counters are satisfied, i.e., if and only if r_S is accepting. By construction, we have $\mathrm{Val}(r) = \mathrm{Val}(r_S)$, and therefore $[\![\mathcal{A}]\!] = h([\![\mathcal{A}_S]\!])$.

(ii) Let $\mathcal{A}_{S'} = (Q_{S'}, \Gamma, I_{S'}, \delta_{S'}, F_{S'}, n_{S'}, \mathrm{Val})$ be a Val-BMCA over Γ with $[\![\mathcal{A}_{S'}]\!] = S'$. Then we let $\mathcal{A} = (Q_{S'}, \Sigma, I_{S'}, \delta, F_{S'}, n_{S'}, \mathrm{Val})$ be a Val-BMCA over Σ with $(p, a, q, \bar{u}) \in \delta$ if and only if $(p, h(a), q, \bar{u}) \in \delta_{S'}$. It is easy to see that $\mathcal{A}$ recognizes $h^{-1}(S') = S' \circ h$. □

For two series $S_1, S_2 \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$, the minimum $\min(S_1, S_2)$ of S_1 and S_2 is defined pointwise, i.e.,

$$\min(S_1, S_2)(w) = \min\{S_1(w), S_2(w)\}.$$

As the next lemma shows, taking the minimum of MC-recognizable series preserves recognizability.

Lemma 5. *For a given ω-valuation function* Val*, the* Val*-MC-recognizable series are closed under minimum.*

Proof. We show this using the usual union construction for automata: for two BMCA $\mathcal{A}_1 = (Q_1, \Sigma, I_1, \delta_1, F_1, n_1, \mathrm{Val})$ and $\mathcal{A}_2 = (Q_2, \Sigma, I_2, \delta_2, F_2, n_2, \mathrm{Val})$ with disjoint state spaces, the BMCA $(Q_1 \cup Q_2, \Sigma, I_1 \cup I_2, \delta_1 \cup \delta_2, F_1 \cup F_2, \max\{n_1, n_2\}, \mathrm{Val})$ recognizes $\min([\![\mathcal{A}_1]\!], [\![\mathcal{A}_2]\!])$. Here, we implicitly fill every tuple of weights $\bar{u}$ of a transition from $\delta_1 \cup \delta_2$ with 0's if it does not have $\max\{n_1, n_2\}$ entries. □

Let $L \subseteq \Sigma^\omega$ and $S \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$. The *intersection of L and S* is the series $L \cap S \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ defined for $w \in \Sigma^\omega$ by

$$L \cap S(w) = \begin{cases} S(w) & \text{if } w \in L \\ \infty & \text{otherwise.} \end{cases}$$

As the next lemma shows, intersecting an ω-recognizable language with an MC-recognizable series preserves MC-recognizability.

Lemma 6. *Let* Val *be an ω-valuation function, let $L \subseteq \Sigma^\omega$ be ω-recognizable, and let $S \colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ be* Val*-MC-recognizable. Then $L \cap S$ is also* Val*-MC-recognizable.*

Proof. The proof is similar to the standard product construction to show that recognizable languages are closed under intersection. Let $\mathcal{A} = (Q, \Sigma, q_0, \delta, \mathcal{F})$ be an NMA with $\mathcal{L}_\omega(\mathcal{A}) = L$ and $\mathcal{A}' = (Q', \Sigma, I', \delta', \mathcal{F}', n, \mathrm{Val})$ an MMCA with $[\![\mathcal{A}']\!] = S$. We construct a new MMCA $\mathcal{A}'' = (Q'', \Sigma, I'', \delta'', \mathcal{F}'', n, \mathrm{Val})$ with $[\![\mathcal{A}'']\!] = L \cap S$ as follows. We let $Q'' = Q \times Q'$, $I'' = \{q_0\} \times I'$, and we let $\pi_1 \colon Q'' \to Q$ and $\pi_2 \colon Q'' \to Q'$ be the projections. Then we let $F'' \in \mathcal{F}''$ iff both $\pi_1(F'') \in \mathcal{F}$ and $\pi_2(Q'') \in \mathcal{F}'$. The set of transitions $\delta'' \subset Q'' \times \Sigma \times Q'' \times (\mathbb{Z} \cup \{s, t\})^n$ is defined by $((p, p'), a, (q, q'), \bar{u}) \in \delta''$ iff $(p, a, q) \in \delta$ and $(p', a, q', \bar{u}) \in \delta'$. Then for every infinite word $w \in \Sigma^\omega$, there is an obvious bijection between the pairs of accepting runs $(r, r') \in \mathrm{Acc}_{\mathcal{A}}(w) \times \mathrm{Acc}_{\mathcal{A}'}(w)$ and the runs $r'' \in \mathrm{Acc}_{\mathcal{A}''}(w)$, and under this bijection we have $\mathrm{Val}(r'') = \mathrm{Val}(r')$. Thus for every word $w \in L$ we have $[\![\mathcal{A}'']\!](w) = S(w)$, as $\mathrm{Acc}_{\mathcal{A}}(w) \neq \emptyset$. For $w \notin L$ we have $\mathrm{Acc}_{\mathcal{A}}(w) = \emptyset$ and therefore $[\![\mathcal{A}'']\!](w) = \infty$. □

4 Monitor MSO logic

In this section, we develop a logic which captures exactly the MC-recognizable series. We first want to give a motivation for the quantifiers and restrictions we use in our logic. We are looking for a logic which is expressively equivalent to automata with monitor counters. It is clear that we need a valuation quantifier in order to model the valuation done by the automata. The question is which types of formulas should be allowed in the scope of the valuation quantifier. From [8], it follows that allowing only *almost Boolean* formulas (see below) is too weak. We would only describe Muller automata over valuation monoids, and these are strictly weaker than automata with monitor counters [4].

We therefore have to allow at least some other quantifier in the scope of the valuation quantifier. Taking into account the automaton model we want to describe, this should be a sum quantifier. Most weighted logics [6, 9, 8, 14, 15, 5] use quantifiers that act unconditionally on the whole input, i.e., on the whole word, tree, or picture. However, in Lemma 11 we will see that in our case, an unrestricted sum quantifier is too powerful.

The intention of the sum quantifier as we define it here is to have a sum quantifier which acts on infinite words, but still computes only finite sums on a given word. The computation of the sum quantifier depends on a first order variable x and a second order variable X provided to it. The variable X serves as a "list" of start and stop positions, and the variable x indicates where the summation on the infinite word should take place. Simply put, the sum is evaluated to $\mathbb{1}$ if x does not point to a position in X or there is no successor of x in X. Otherwise, if y is x's successor in X, the sum is taken from $x+1$ to $y-1$.

Intuitively, each sum quantifier corresponds to a counter. In a run of an automaton with monitor counters, not more than one counter can be started at each letter of the given word. Therefore, we use Boolean formulas to choose which counter to use. We combine these choices between counters into so-called *x-summing* formulas, where x is the first order variable provided to each sum quantifier in the formula.

We provide a countable set $\mathcal{V}$ of first and second order variables, where lower case letters like x and y denote first order variables and capital letters like X and Y denote second order variables. We define a three step logic over an alphabet Σ and an ω-valuation function Val according to the following grammars.

$$\begin{aligned}
&\beta ::= P_a(x) \mid x \leq y \mid x \in X \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\beta \mid \exists X.\beta \\
&\psi ::= k \mid \beta \,?\, \psi : \psi \\
&\zeta_x ::= \mathbb{1} \mid \beta \,?\, \zeta_x : \zeta_x \mid \bigoplus^{x,X} y.\psi \\
&\varphi ::= \beta \,?\, \varphi : \varphi \mid \min(\varphi, \varphi) \mid \inf x.\varphi \mid \inf X.\varphi \mid \operatorname{Val} x.\zeta_x
\end{aligned}$$

where $x, y, X \in \mathcal{V}$, $a \in \Sigma$, and $k \in \mathbb{Z}$. The formulas β are called *Boolean* or *MSO* formulas, the formulas ψ *almost Boolean* formulas, the formulas ζ_x *x-summing* formulas, and the formulas φ *monitor MSO* (mMSO) formulas. We denote the sets of Boolean, almost Boolean, and x-summing formulas over Σ by $\mathrm{MSO}(\Sigma)$, $\mathrm{mMSO}^{\text{a-bool}}(\Sigma)$, and $\mathrm{mMSO}^{x}(\Sigma)$, respectively, and the set of monitor MSO formulas over Σ and Val by $\mathrm{mMSO}(\Sigma, \mathrm{Val})$. We remark that within an x-summing formula, the first order variable provided to each sum quantifier is always x. This restriction is not imposed on the second order quantifiers, i.e., $\beta \,?\, \bigoplus^{x,X} y.\psi_1 : \bigoplus^{x,Z} y.\psi_2$ is an x-summing formula, but $\beta \,?\, \bigoplus^{x,X} y.\psi_1 : \bigoplus^{z,Z} y.\psi_2$ is neither an x-summing nor a z-summing formula. Also note that the x-summing formulas are only auxiliary formulas, see Remark 7 later on.

The set of free variables $\mathrm{Free}(\varphi)$ is defined as usual, i.e., $\exists, \inf$, and Val bind variables, and in $\bigoplus^{x,X} y.\psi$ the variable y is bound. A formula without free variables is called a *sentence.*

Let $w = a_0 a_1 \ldots \in \Sigma^\omega$. Let $\mathcal{V}$ be a finite set of first and second order variables with $\mathrm{Free}(\varphi) \subseteq \mathcal{V}$. A $(\mathcal{V}, w)$-assignment is a mapping $\rho\colon \mathcal{V} \to \mathbb{N} \cup \mathcal{P}(\mathbb{N})$ where every first order variable is mapped to an element of $\mathbb{N}$ and every second order variable is mapped to a subset of $\mathbb{N}$. The *update* $\rho[x \to i]$ for $i \in \mathbb{N}$ is defined as $\rho[x \to i](x) = i$ and $\rho[x \to i](\mathcal{X}) = \rho(\mathcal{X})$ for all $\mathcal{X} \in \mathcal{V} \setminus \{x\}$. The update $\rho[X \to I]$ for $I \subseteq \mathbb{N}$ is defined similarly. We encode $(\mathcal{V}, w)$-assignments as usual with an extended alphabet $\Sigma_{\mathcal{V}} = \Sigma \times \{0,1\}^{\mathcal{V}}$: to a pair (w, ρ), we associate the word $(a_0, \rho_0)(a_1, \rho_1) \ldots \in \Sigma_{\mathcal{V}}^\omega$ where

$$\rho_i(\mathcal{X}) = \begin{cases} 1 & \text{if } \mathcal{X} \text{ is a first order variable and } i = \rho(\mathcal{X}) \\ 1 & \text{if } \mathcal{X} \text{ is a second order variable and } i \in \rho(\mathcal{X}) \\ 0 & \text{otherwise} \end{cases}$$

for $i \in \mathbb{N}$. An infinite word $(a_0, \rho_0)(a_1, \rho_1) \ldots$ over $\Sigma_{\mathcal{V}}$ is called *valid* if and only if for every first order variable the respective row in the $\{0,1\}^{\mathcal{V}}$-coordinate contains exactly one 1. In this case, we denote this word by (w, ρ), where w is the projection to Σ and ρ is the $(\mathcal{V}, w)$-assignment we obtain from the ρ_i by reversing the above association. It is not difficult to see that the set

$$N_{\mathcal{V}} = \{(w, \rho) \in \Sigma_{\mathcal{V}}^\omega \mid (w, \rho) \text{ is valid}\}$$

is ω-recognizable. Let $(w, \rho) \in \Sigma_{\mathcal{V}}^\omega$ with $w = a_0 a_1 \ldots \in \Sigma^\omega$. For a Boolean formula β we define the satisfaction relation $(w, \rho) \models \beta$ as usual: if (w, ρ) is not valid, then $(w, \rho) \models \beta$ does not hold; otherwise we define it as follows.

$$\begin{aligned}
(w,\rho) &\models P_a(x) &&\iff a_{\rho(x)} = a\\
(w,\rho) &\models x \leq y &&\iff \rho(x) \leq \rho(y)\\
(w,\rho) &\models x \in X &&\iff \rho(x) \in \rho(X)\\
(w,\rho) &\models \neg\beta &&\iff (w,\rho) \models \beta \text{ does not hold}\\
(w,\rho) &\models \beta_1 \vee \beta_2 &&\iff (w,\rho) \models \beta_1 \text{ or } (w,\rho) \models \beta_2\\
(w,\rho) &\models \exists x.\beta &&\iff (w,\rho[x \to i]) \models \beta \text{ for some } i \in \mathbb{N}\\
(w,\rho) &\models \exists X.\beta &&\iff (w,\rho[X \to I]) \models \beta \text{ for some } I \subseteq \mathbb{N}.
\end{aligned}$$

Let β be an MSO formula. We will write Σ_β for $\Sigma_{\mathrm{Free}(\beta)}$ and N_β for $N_{\mathrm{Free}(\beta)}$. We recall the fundamental Büchi theorem [1], namely that for $\mathrm{Free}(\beta) \subseteq \mathcal{V}$ the language

$$\mathcal{L}_\mathcal{V}(\beta) = \{(w,\rho) \in N_\mathcal{V} \mid (w,\rho) \models \beta\}$$

defined by β over $\Sigma_\mathcal{V}$ is ω-recognizable. We abbreviate $\mathcal{L}(\beta) = \mathcal{L}_{\mathrm{Free}(\beta)}(\beta)$. Conversely, every ω-recognizable language $L \subseteq \Sigma^\omega$ is definable by an MSO sentence β, i.e., $L = \mathcal{L}(\beta)$.

We now address the semantics of the remaining formulas. Let Val be an ω-valuation function. For an almost Boolean, x-summing, or monitor MSO formula η, we define the *semantics* $[\![\eta]\!]_\mathcal{V}(w,\rho)$ *of* η *under the* $(\mathcal{V},w)$*-assignment* ρ as follows: if (w,ρ) is not valid, then $[\![\eta]\!]_\mathcal{V}(w,\rho) = \infty$; otherwise the semantics are defined as follows.

$$\begin{aligned}
[\![k]\!]_\mathcal{V}(w,\rho) &= k\\
[\![\beta \mathbin{?} \psi_1 : \psi_2]\!]_\mathcal{V}(w,\rho) &= \begin{cases} [\![\psi_1]\!]_\mathcal{V}(w,\rho) & \text{if } (w,\rho) \models \beta\\ [\![\psi_2]\!]_\mathcal{V}(w,\rho) & \text{otherwise}\end{cases}\\
[\![\textstyle\bigoplus^{x,X} y.\psi]\!]_\mathcal{V}(w,\rho) &= \begin{cases} \sum_{i=\rho(x)+1}^{\min\{j \in \rho(X) \mid j > \rho(x)\}-1} [\![\psi]\!]_{\mathcal{V}\cup\{y\}}(w,\rho[y \to i]) & \text{if } \rho(x) \in \rho(X) \text{ and}\\ & \{j \in \rho(X) \mid j > \rho(x)\} \neq \emptyset\\ \mathbb{1} & \text{otherwise.}\end{cases}\\
[\![\min(\varphi_1,\varphi_2)]\!]_\mathcal{V}(w,\rho) &= \min\{[\![\varphi_1]\!]_\mathcal{V}(w,\rho), [\![\varphi_2]\!]_\mathcal{V}(w,\rho)\}\\
[\![\inf x.\varphi]\!]_\mathcal{V}(w,\rho) &= \inf_{i\in\mathbb{N}} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x \to i])\\
[\![\inf X.\varphi]\!]_\mathcal{V}(w,\rho) &= \inf_{I\subseteq\mathbb{N}} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(w,\rho[X \to I])\\
[\![\mathrm{Val}\, x.\zeta_x]\!]_\mathcal{V}(w,\rho) &= \mathrm{Val}(([\![\zeta_x]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x \to i]))_{i\in\mathbb{N}})
\end{aligned}$$

We write $[\![\eta]\!]$ for $[\![\eta]\!]_{\mathrm{Free}(\eta)}$.

Remark 7. From the semantics defined here it is clear that every x-summing sentence ζ_x is semantically equivalent to $\mathbb{1}$. In this sense, the x-summing formulas constitute no meaningful fragment of our logic, and are only auxiliary formulas for the construction of monitor MSO formulas.

In Lemma 12 we will see that the first order variable x is necessarily also the variable which is quantified by Val, i.e., allowing formulas like $\mathrm{Val}\, x.\zeta_z$ leads to formulas which are not MC-recognizable.

Note also that for every valid (w,ρ), we have $[\![\mathrm{Val}\, x.\mathbb{1}]\!]_\mathcal{V}(w,\rho) = \infty$. By abuse of notation, we can thus define the abbreviation $\infty = \mathrm{Val}\, x.\mathbb{1}$.

Remark 8. The condition used in the definition of the sum quantifier is definable by the MSO formula

$$\mathrm{notLast}(x,X) = x \in X \wedge \exists y.(y \in X \wedge x < y),$$

where $x < y$ is an abbreviation for $x \leq y \wedge \neg(y \leq x)$. We can therefore also write

$$[\![\textstyle\bigoplus^{x,X} y.\psi]\!]_\mathcal{V}(w,\rho) = \begin{cases} \sum_{i=\rho(x)+1}^{\min\{j \in \rho(X) \mid j > \rho(x)\}-1} [\![\psi]\!]_{\mathcal{V}\cup\{y\}}(w,\rho[y \to i]) & \text{if } (w,\rho) \models \mathrm{notLast}(x,X)\\ \mathbb{1} & \text{otherwise.}\end{cases}$$

If we define an unrestricted sum quantifier $\bigoplus y.\psi$ by

$$[\![\textstyle\bigoplus y.\psi]\!]_\mathcal{V}(w,\rho) = \begin{cases} \sum_{i\in\mathbb{N}} [\![\psi]\!]_{\mathcal{V}\cup\{y\}}(w,\rho[y \to i]) & \text{if this sum converges}\\ \infty & \text{otherwise,}\end{cases}$$

we can write our restricted sum quantifier as

$$[\![\textstyle\bigoplus^{x,X} y.\psi]\!]_\mathcal{V}(w,\rho) = [\![\mathrm{notLast}(x,X) \mathbin{?} \bigoplus y.(x < y \wedge \forall z.((x < z \wedge z \leq y) \to \neg z \in X) \mathbin{?} \psi : 0) : \mathbb{1}]\!]_\mathcal{V}(w,\rho).$$

Example 9. Consider Example 1 again, i.e., the alphabet $\Sigma = \{\mathsf{demand}, \mathsf{restock}\}$ with the ω-valuation function $\mathrm{Val} = \textsc{Ces}$. Then the formula

$$\varphi = \inf X. \Big(\forall z.(z \in X \leftrightarrow P_{\mathsf{restock}}(z)) \mathbin{?} \mathrm{Val}\, x. \bigoplus\nolimits^{x,X} y.1 : \infty\Big)$$

describes the average number of $\mathsf{demands}$ between two $\mathsf{restocks}$. We recall that ∞ is simply an abbreviation for the formula $\mathrm{Val}\, x.\mathbb{1}$. As in Example 1, if we take $\textsc{Inf}$ or $\textsc{Sup}$ for the valuation function, the formula above describes the lowest or highest demand ever encountered.

We have the following fundamental lemma which intuitively states that the semantics and recognizability of a formula depend only on the variables occurring in it.

Lemma 10 (Consistency Lemma). *Let $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ and let $\mathcal{V}$ be a finite set of variables with $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$.*

(i) For every valid $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ we have $[\![\varphi]\!]_\mathcal{V}(w, \rho) = [\![\varphi]\!](w, \rho\restriction_{\mathrm{Free}(\varphi)})$.

(ii) $[\![\varphi]\!]$ is Val-MC-recognizable if and only if $[\![\varphi]\!]_\mathcal{V}$ is Val-MC-recognizable.

Proof. (i) This can be shown by induction on φ using the same ideas as in [6]. We first show that the statement also holds for Boolean, almost Boolean, and x-summing formulas.

Let β be of the form $P_a(x)$, $x \le y$, or $x \in X$ where x and y are first order variables and X is a second order variable. Then for every valid $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ with $\mathcal{V} \supseteq \mathrm{Free}(\beta)$, it is immediate from the definition of satisfaction that $(w, \rho) \models \beta$ if and only if $(w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta$.

Next, assume that $\beta = \neg\beta'$ with $\beta' \in \mathrm{MSO}(\Sigma)$ and let $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ be valid. Then since $\mathcal{V} \supseteq \mathrm{Free}(\beta) = \mathrm{Free}(\beta')$, we have by induction that

$$\begin{aligned}(w, \rho) \models \beta &\iff \text{not } (w, \rho) \models \beta' \\ &\iff \text{not } (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta' \\ &\iff (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta.\end{aligned}$$

Now assume that $\beta = \beta_1 \vee \beta_2$ with $\beta_1, \beta_2 \in \mathrm{MSO}(\Sigma)$ and let $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ be valid. Then since $\mathcal{V} \supseteq \mathrm{Free}(\beta) \supseteq \mathrm{Free}(\beta_1)$ and $\mathcal{V} \supseteq \mathrm{Free}(\beta) \supseteq \mathrm{Free}(\beta_2)$, we have by induction that

$$\begin{aligned}(w, \rho) \models \beta &\iff (w, \rho) \models \beta_1 \text{ or } (w, \rho) \models \beta_2 \\ &\iff (w, \rho\restriction_{\mathrm{Free}(\beta_1)}) \models \beta_1 \text{ or } (w, \rho\restriction_{\mathrm{Free}(\beta_2)}) \models \beta_2 \\ &\iff (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta_1 \text{ or } (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta_2 \\ &\iff (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta.\end{aligned}$$

For the first order existential quantifier, assume that $\beta = \exists x.\beta'$ with $\beta' \in \mathrm{MSO}(\Sigma)$ and let $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ be valid. By definition, we have

$$(w, \rho) \models \beta \iff (w, \rho[x \to i]) \models \beta' \text{ for some } i \in \mathbb{N}.$$

Due to $\mathcal{V} \supseteq \mathrm{Free}(\beta)$, we have

$$\mathcal{V} \cup \{x\} \supseteq \mathrm{Free}(\beta) \cup \{x\} \supseteq \mathrm{Free}(\beta') \cup \{x\} \supseteq \mathrm{Free}(\beta').$$

By applying the induction hypothesis twice, we thus have for every $i \in \mathbb{N}$ that

$$\begin{aligned}(w, \rho[x \to i]) \models \beta' &\iff (w, \rho[x \to i]\restriction_{\mathrm{Free}(\beta')}) \models \beta' \\ &\iff (w, \rho\restriction_{\mathrm{Free}(\beta')}[x \to i]) \models \beta'.\end{aligned}$$

It follows that $(w, \rho) \models \beta \iff (w, \rho\restriction_{\mathrm{Free}(\beta')}) \models \beta$. For the the second order existential quantifier, we can proceed in the same way.

Next, we address almost Boolean formulas. For $\psi = k$ with $k \in \mathbb{Z}$, the statement is clear. For $\psi = \beta \mathbin{?} \psi_1 : \psi_2$ with $\beta \in \mathrm{MSO}(\Sigma)$ and $\psi_1, \psi_2 \in \mathrm{mMSO}^{\text{a-bool}}(\Sigma)$, let $(w, \rho) \in \Sigma^\omega_\mathcal{V}$ be valid. Then since $\mathcal{V} \supseteq \mathrm{Free}(\psi) \supseteq \mathrm{Free}(\psi_1)$, $\mathcal{V} \supseteq \mathrm{Free}(\psi) \supseteq \mathrm{Free}(\psi_2)$, and $\mathcal{V} \supseteq \mathrm{Free}(\psi) \supseteq \mathrm{Free}(\beta)$, we have by induction that

$$[\![\psi]\!]_\mathcal{V}(w, \rho) = \begin{cases} [\![\psi_1]\!]_\mathcal{V}(w, \rho) & \text{if } (w, \rho) \models \beta \\ [\![\psi_2]\!]_\mathcal{V}(w, \rho) & \text{otherwise} \end{cases}$$

$$= \begin{cases} [\![\psi_1]\!](w, \rho\restriction_{\mathrm{Free}(\psi_1)}) & \text{if } (w, \rho\restriction_{\mathrm{Free}(\beta)}) \models \beta \\ [\![\psi_2]\!](w, \rho\restriction_{\mathrm{Free}(\psi_2)}) & \text{otherwise} \end{cases}$$
$$= \begin{cases} [\![\psi_1]\!]_{\mathrm{Free}(\psi)}(w, \rho\restriction_{\mathrm{Free}(\psi)}) & \text{if } (w, \rho\restriction_{\mathrm{Free}(\psi)}) \models \beta \\ [\![\psi_2]\!]_{\mathrm{Free}(\psi)}(w, \rho\restriction_{\mathrm{Free}(\psi)}) & \text{otherwise} \end{cases}$$
$$= [\![\psi]\!](w, \rho\restriction_{\mathrm{Free}(\psi)}).$$

For x-summing formulas, we proceed as follows. For $\zeta = \mathbb{1}$, the statement is clear. For $\zeta = \beta \mathbin{?} \zeta_1 : \zeta_2$ with $\beta \in \mathrm{MSO}(\Sigma)$ and $\zeta_1, \zeta_2 \in \mathrm{mMSO}^x(\Sigma)$, we can proceed in the same way as for almost Boolean formulas. Assume that $\zeta = \bigoplus^{x,X} y.\psi$ with $\psi \in \mathrm{mMSO}^{\text{a-bool}}(\Sigma)$ and let $(w, \rho) \in \Sigma^\omega_{\mathcal{V}}$ be valid. We have $\mathcal{V} \supseteq \mathrm{Free}(\zeta) = \mathrm{Free}(\psi) \setminus \{y\} \cup \{x, X\}$. In particular, we have

$$\mathcal{V} \cup \{y\} \supseteq \mathrm{Free}(\zeta) \cup \{y\} = \mathrm{Free}(\psi) \cup \{y, x, X\} \supseteq \mathrm{Free}(\psi).$$

Thus, we see by induction that for every $i \in \mathbb{N}$, we have

$$\begin{aligned} [\![\psi]\!]_{\mathcal{V}}(w, \rho[y \to i]) &= [\![\psi]\!](w, \rho[y \to i]\restriction_{\mathrm{Free}(\psi)}) \\ &= [\![\psi]\!]_{\mathrm{Free}(\zeta)\cup\{y\}}(w, \rho\restriction_{\mathrm{Free}(\zeta)}[y \to i]), \end{aligned}$$

from which the statement follows.

Finally, we consider monitor MSO formulas. For formulas of the form $\beta \mathbin{?} \varphi_1 : \varphi_2$ with $\varphi_1, \varphi_2 \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$, we proceed in the same way as for almost Boolean formulas. For $\varphi = \min(\varphi_1, \varphi_2)$ with $\varphi_1, \varphi_2 \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$, let $(w, \rho) \in \Sigma^\omega_{\mathcal{V}}$ be valid. Then since $\mathcal{V} \supseteq \mathrm{Free}(\varphi) \supseteq \mathrm{Free}(\varphi_1)$ and $\mathcal{V} \supseteq \mathrm{Free}(\varphi) \supseteq \mathrm{Free}(\varphi_2)$, we see by induction that

$$\begin{aligned} [\![\varphi]\!]_{\mathcal{V}}(w, \rho) &= \min\{[\![\varphi_1]\!]_{\mathcal{V}}(w, \rho), [\![\varphi_2]\!]_{\mathcal{V}}(w, \rho)\} \\ &= \min\{[\![\varphi_1]\!](w, \rho\restriction_{\mathrm{Free}(\varphi_1)}), [\![\varphi_2]\!](w, \rho\restriction_{\mathrm{Free}(\varphi_2)})\} \\ &= \min\{[\![\varphi_1]\!]_{\mathrm{Free}(\varphi)}(w, \rho\restriction_{\mathrm{Free}(\varphi)}), [\![\varphi_2]\!]_{\mathrm{Free}(\varphi)}(w, \rho\restriction_{\mathrm{Free}(\varphi)})\} \\ &= [\![\varphi]\!](w, \rho\restriction_{\mathrm{Free}(\varphi)}). \end{aligned}$$

For $\varphi = \mathrm{Val}\, x.\zeta$ with $\zeta \in \mathrm{mMSO}^x(\Sigma)$, let $(w, \rho) \in \Sigma^\omega_{\mathcal{V}}$ be valid. Since $\mathcal{V} \supseteq \mathrm{Free}(\varphi)$, we have

$$\mathcal{V} \cup \{x\} \supseteq \mathrm{Free}(\varphi) \cup \{x\} \supseteq \mathrm{Free}(\zeta) \cup \{x\} \supseteq \mathrm{Free}(\zeta).$$

Thus, we see by induction that

$$\begin{aligned} [\![\varphi]\!]_{\mathcal{V}}(w, \rho) &= \mathrm{Val}(([\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w, \rho[x \to i]))_{i\in\mathbb{N}}) \\ &= \mathrm{Val}(([\![\zeta]\!](w, \rho[x \to i]\restriction_{\mathrm{Free}(\zeta)}))_{i\in\mathbb{N}}) \\ &= \mathrm{Val}(([\![\zeta]\!]_{\mathrm{Free}(\varphi)\cup\{x\}}(w, \rho\restriction_{\mathrm{Free}(\varphi)}[x \to i]))_{i\in\mathbb{N}}) \\ &= [\![\varphi]\!](w, \rho\restriction_{\mathrm{Free}(\varphi)}). \end{aligned}$$

The cases where $\varphi = \inf x.\varphi'$ or $\varphi = \inf X.\varphi'$ with $\varphi' \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ are proved in the same way.

(ii) Consider the homomorphism $h\colon \Sigma^\omega_{\mathcal{V}} \to \Sigma^\omega_{\varphi}$ defined by $(w, \rho) \mapsto (w, \rho\restriction_{\mathrm{Free}(\varphi)})$. If $[\![\varphi]\!]_{\mathcal{V}}$ is Val-MC-recognizable, then by Lemma 4 (i), the series $[\![\varphi]\!] = h([\![\varphi]\!]_{\mathcal{V}})$ is also Val-MC-recognizable.

Conversely, let $[\![\varphi]\!]$ be Val-MC-recognizable and let $N_{\mathcal{V}}$ be the language of all words (w, ρ) where ρ is a valid $(\mathcal{V}, w)$-assignment. $N_{\mathcal{V}}$ is an ω-recognizable language. We have $[\![\varphi]\!]_{\mathcal{V}} = N_{\mathcal{V}} \cap h^{-1}([\![\varphi]\!])$ because of (i). Due to Lemma 4 (ii) and Lemma 6, $N_{\mathcal{V}} \cap h^{-1}([\![\varphi]\!])$ is Val-MC-recognizable. □

In the following lemma, we show that the use of an unrestricted sum quantifier leads to series which are not MC-recognizable.

Lemma 11. *Consider the unrestricted sum quantifier from Remark 8*

$$[\![\bigoplus y.\psi]\!]_{\mathcal{V}}(w, \rho) = \begin{cases} \sum_{i\in\mathbb{N}} [\![\psi]\!]_{\mathcal{V}\cup\{y\}}(w, \rho[y \to i]) & \textit{if this sum converges} \\ \infty & \textit{otherwise,} \end{cases}$$

the ω-valuation function Val *defined by*

$$\mathrm{Val}((z_i)_{i\geq 0}) = \begin{cases} \sum_{i=0}^{\infty} z_i & \textit{if this sum converges} \\ \infty & \textit{otherwise,} \end{cases}$$

and the alphabet $\Sigma = \{a, b\}$. *Then for the almost Boolean formula*

$$\psi = y \leq x \wedge \forall z.(z \leq x \to P_a(z)) \,?\, -1 : 0,$$

the formula

$$\varphi = \mathrm{Val}\, x. \bigoplus y.\psi$$

is not Val*-MC-recognizable.*

Proof. Let $w = a_0 a_1 \ldots \in \Sigma^\omega$, then for $i \in \mathbb{N}$ we have

$$[\![\bigoplus y.\psi]\!]_{\{x\}}(w, [x \to i]) = \begin{cases} -(i+1) & \text{if } a_0 = a_1 = \ldots = a_i = a \\ 0 & \text{otherwise.} \end{cases}$$

By the Gauß summation formula, φ hence describes the series

$$[\![\varphi]\!](w) = \begin{cases} -\frac{m(m+1)}{2} & \text{if } w = a^m b w' \text{ for some } w' \in \Sigma^\omega \\ \infty & \text{if } w = a^\omega. \end{cases}$$

The idea is now that with only finitely many transitions, and therefore only finitely many different weights, this quadratic growth cannot be realized if only transitions up to the first b in each word influence the weight of the runs. But once the automaton has read this first b, it cannot distinguish between the words anymore. Under appropriate assumptions, we can therefore combine runs from different words to obtain a contradiction.

Assume there was a BMCA $\mathcal{A} = (Q, \Sigma, I, \delta, F, n, \mathrm{Val})$ with $[\![\mathcal{A}]\!] = [\![\varphi]\!]$. We consider the special words $w_m = a^m b a^\omega$. For $m \geq 0$ and $r \in \mathrm{Acc}_{\mathcal{A}}(w_m)$, we have $\mathrm{Val}(r) \in \mathbb{Z} \cup \{\infty\}$ and $\mathrm{Val}(r) \geq [\![\varphi]\!](w_m)$. Therefore, there must be a minimal run in $\mathrm{Acc}_{\mathcal{A}}(w_m)$, i.e., $r \in \mathrm{Acc}_{\mathcal{A}}(w_m)$ with $[\![\varphi]\!](w_m) = \mathrm{Val}(r)$.

For a minimal run $r = (d_i)_{i \geq 0}$ of $\mathcal{A}$ on w_m with $d_i = (q_i, a_i, q_{i+1}, \bar{u}^i)$, we define the *counter pattern* $\mathrm{CP}(r)$ and the *effective weights* $\mathrm{EW}^{\leq}(r)$ and $\mathrm{EW}^{>}(r)$ of r as follows. Intuitively, the counter pattern tells us for each counter whether it is active or not at the letter b of w_m. For $j \in \{1, \ldots, n\}$, we let $k_j = 1$ if there is an $i \leq m$ such that $u_j^i = s$ and for all i' with $i < i' \leq m$ we have $u_j^{i'} \in \mathbb{Z}$. Otherwise we let $k_j = 0$. Then we define $\mathrm{CP}(r) = (k_1, \ldots, k_n) \in \{0, 1\}^n$. The effective weights will be partial computations of $\mathrm{Val}(r)$ on $a^m b$ on the one hand and on a^ω on the other hand. For $j \in \{1, \ldots, n\}$, we let

$$E_j^{\leq} = \sum_{\substack{i=0 \\ u_j^i = s}}^{m} \ \sum_{i'=i+1}^{\min(\{m\} \cup \{k \geq i | u_j^{k+1} = t\})} u_j^{i'}$$

$$E_j^{>} = \begin{cases} \sum_{\substack{i=m+1 \\ u_j^i = s}}^{\infty} \sum_{i'=i+1}^{\min\{k \geq i | u_j^{k+1} = t\}} u_j^{i'} & \text{if } k_j = 0 \\ \sum_{\substack{i=m+1 \\ u_j^i = s}}^{\infty} \sum_{i'=i+1}^{\min\{k \geq i | u_j^{k+1} = t\}} u_j^{i'} + \sum_{i'=m+1}^{\min\{k \geq m | u_j^{k+1} = t\}} u_j^{i'} & \text{if } k_j = 1. \end{cases}$$

The empty sum is simply defined as 0. We let $\mathrm{EW}^{\leq}(r) = \sum_{j=1}^n E_j^{\leq}$ and $\mathrm{EW}^{>}(r) = \sum_{j=1}^n E_j^{>}$. We have

$$\mathrm{Val}(r) = \mathrm{EW}^{\leq}(r) + \mathrm{EW}^{>}(r).$$

Now for every $m \geq 0$, let $r_m = (d_i^m)_{i \geq 0}$ be a minimal run of $\mathcal{A}$ on w_m. We consider the pairs $\mathfrak{p}_m = (\mathrm{CP}(r_m), d_m^m)$. Since there are only finitely many different such pairs, namely not more than $2^n \cdot |\delta|$, there must be such a pair $\mathfrak{p}$ and a subsequence $(r_{m_k})_{k \geq 0}$ of $(r_m)_{m \geq 0}$ with $\mathfrak{p}_{m_k} = \mathfrak{p}$ for all $k \geq 0$.

Now let $M > 0$ such that all weights occurring in δ are in $[-M, M]$. Then we have

$$|\mathrm{EW}^{\leq}(r_m)| \leq nM(m+1).$$

For the special index $m_1 \geq 1$, we choose k sufficiently large to ensure that

$$-nM(m_1 + 1) - \frac{m_1(m_1+1)}{2} > -\frac{m_k(m_k+1)}{2} + nM(m_k + 1).$$

This is possible since if we treat the right hand side of this inequality as a polynomial in m_k, the leading coefficient of this polynomial is negative, thus the polynomial tends to minus infinity for $k \to \infty$. We know that

$$\begin{aligned}\mathrm{EW}^{\leq}(r_{m_1}) &\leq nM(m_1+1)\\ \mathrm{EW}^{\leq}(r_{m_k}) &\geq -nM(m_k+1).\end{aligned}$$

If we had

$$\mathrm{EW}^{>}(r_{m_k}) \geq -nM(m_1+1) - \frac{m_1(m_1+1)}{2},$$

then we would have

$$\begin{aligned}\mathrm{Val}(r_{m_k}) &= \mathrm{EW}^{\leq}(r_{m_k}) + \mathrm{EW}^{>}(r_{m_k})\\ &\geq -nM(m_k+1) - nM(m_1+1) - \frac{m_1(m_1+1)}{2}\\ &> -\frac{m_k(m_k+1)}{2},\end{aligned}$$

which is a contradiction to the choice of r_{m_k} as minimal. We therefore have

$$\mathrm{EW}^{\leq}(r_{m_1}) + \mathrm{EW}^{>}(r_{m_k}) < -\frac{m_1(m_1+1)}{2}.$$

We now consider the sequence of transitions $r = d_0^{m_1} \dots d_{m_1}^{m_1} d_{m_k+1}^{m_k} d_{m_k+2}^{m_k} \dots$ and claim that it is an accepting run of $\mathcal{A}$ on w_{m_1}. The first state is initial as r_{m_1} is an accepting run and the transitions are well matched since $d_{m_1}^{m_1} = d_{m_k}^{m_k}$, which are also the only b-transitions of these runs. Since r_{m_k} is an accepting run, it is clear that the Büchi acceptance condition is fulfilled and infinitely often some counter is activated. Finally, that the starts and stops form well matched pairs follows from the fact that $\mathrm{CP}(r_{m_1}) = \mathrm{CP}(r_{m_k})$.

To conclude, we have

$$\begin{aligned}[\![\mathcal{A}]\!](w_{m_1}) &\leq \mathrm{Val}(r)\\ &= \mathrm{EW}^{\leq}(r) + \mathrm{EW}^{>}(r)\\ &= \mathrm{EW}^{\leq}(r_{m_1}) + \mathrm{EW}^{>}(r_{m_k})\\ &< -\frac{m_1(m_1+1)}{2}\\ &= [\![\varphi]\!](w_{m_1}).\end{aligned}$$

Obviously, the behavior of $\mathcal{A}$ does not coincide with the semantics of φ, which is a contradiction to the choice of $\mathcal{A}$. □

The next lemma shows that in order to ensure MC-recognizability, the first order variable x provided to the sum quantifier is necessarily the variable which Val quantifies.

Lemma 12. *Consider the ω-valuation function* Val *defined by*

$$\mathrm{Val}((z_i)_{i\geq 0}) = \begin{cases} \frac{1}{z_0} & \text{if } 0 < z_0 = z_1 = z_2 = \dots \\ -1 & \text{otherwise.}\end{cases}$$

and the alphabet $\Sigma = \{a\}$. We define the abbreviation

$$(y = x+1) = x \leq y \wedge \neg(y \leq x) \wedge \forall z.(z \leq x \vee y \leq z).$$

Then for the Boolean formula

$$\beta(X) = \forall x_1.\forall x_2.((x_1 \in X \wedge x_2 = x_1 + 1) \to \neg(x_2 \in X)),$$

the formula

$$\varphi = \inf X.\inf z.\left(\beta(X)\ ?\ \mathrm{Val}\, x. \bigoplus\nolimits^{z,X} y.1 : \infty\right)$$

is not Val*-MC-recognizable.*

Proof. For $i \in \mathbb{N}$ and $I \subseteq \mathbb{N}$ we have

$$[\![\bigoplus^{z,X} y.1]\!]_{\{z,X\}}(a^\omega,[z\to i, X\to I]) =$$
$$\begin{cases} \mathbb{1} & \text{if } i\notin I \text{ or for all } j>i \text{ we have } j\notin I \\ \min\{j\geq i \mid j+1\in I\}-i & \text{otherwise}\end{cases}$$

and therefore

$$[\![\mathrm{Val}\, x.\bigoplus^{z,X} y.1]\!]_{\{z,X\}}(a^\omega,[z\to i, X\to I]) =$$
$$\begin{cases} \infty & \text{if } i\notin I \text{ or for all } j>i \text{ we have } j\notin I \\ -1 & \text{if } i\in I \text{ and } i+1\in I \\ (\min\{j\geq i \mid j+1\in I\}-i)^{-1} & \text{otherwise.}\end{cases}$$

We obtain

$$[\![\beta(X)\ ?\ \mathrm{Val}\, x.\bigoplus^{z,X} y.1 : \infty]\!]_{\{z,X\}}(a^\omega,[z\to i, X\to I]) =$$
$$\begin{cases} \infty & \text{if } i\notin I \text{ or for all } j>i \text{ we have } j\notin I \\ & \text{or } j,j+1\in I \text{ for some } j\geq 0 \\ (\min\{j\geq i \mid j+1\in I\}-i)^{-1} & \text{otherwise,}\end{cases}$$

in particular $[\![\varphi]\!](a^\omega)\geq 0$. For $m\geq 2$ and the special choices $I=\{0,m+1\}$ and $i=0$, we obtain

$$[\![\beta(X)\ ?\ \mathrm{Val}\, x.\bigoplus^{z,X} y.1 : \infty]\!]_{\{z,X\}}(a^\omega,[z\to i, X\to I]) = \frac{1}{m}$$

and therefore $[\![\varphi]\!](a^\omega)=\inf\{m^{-1}\mid m\geq 2\}=0$.

For a BMCA $\mathcal{A}$ realizing this series, the weight-sequence associated to each run has to be constant, and there must be a sequence of runs such that this constant grows arbitrarily large. We exploit the latter fact to show that there must be a run whose associated weight-sequence is not constant, which leads to the contradiction $[\![\mathcal{A}]\!](a^\omega)=-1$.

Assume there was a BMCA $\mathcal{A}=(Q,\Sigma,I,\delta,F,n,\mathrm{Val})$ with $[\![\mathcal{A}]\!]=[\![\varphi]\!]$. Let $r\in\mathrm{Acc}_{\mathcal{A}}(a^\omega)$ and let $(z_i)_{i\geq 0}$ be the associated weight-sequence. Clearly, we must have $\mathrm{Val}(r)\geq 0$ and therefore $0<z_0=z_1=z_2=\ldots$, i.e., $\mathrm{Val}(r)=z_0^{-1}>0$. Since $\inf_{r\in\mathrm{Acc}_{\mathcal{A}}(a^\omega)}\mathrm{Val}(r)=0$, there must be a sequence $(r_m)_{m\geq 1}$ in $\mathrm{Acc}_{\mathcal{A}}(a^\omega)$ with $\mathrm{Val}(r_m)<\frac{1}{m}$. We write $r_m=(d_i^m)_{i\geq 0}$.

Now similarly to the proof of Lemma 11, we associate to each $m\geq 1$ and $i\geq 0$ a quantifier pattern $\mathrm{CP}_m(i)=(k_1,\ldots,k_n)\in\{0,1\}^n$, which tells us whether in run r_m at transition i, a counter $j\in\{1,\ldots,n\}$ is active or not. More precisely, k_j is 1 if for some $i'\leq i$ counter j is started at i' but not terminated on the positions $\{i'+1,\ldots,i\}$, and 0 otherwise.

For each $m\geq 1$, let $N_m>0$ such that in run r_m, at least one counter was terminated before reading the N_m-th letter of a^ω. For every $\mathfrak{c}\in\{0,1\}^n$ and $d\in\delta$, let $M(d,\mathfrak{c})=\{m\geq 1\mid \text{there exists } i>N_m \text{ with } d_i^m=d \text{ and } \mathrm{CP}_m(i)=\mathfrak{c}\}$. We have $\bigcup_{(d,\mathfrak{c})\in\delta\times\{0,1\}^n}M(d,\mathfrak{c})=\mathbb{N}_+$, and since $\delta\times\{0,1\}^n$ is a finite set, at least one $M(d,\mathfrak{c})$ must be infinite.

Consider such an infinite $M(d,\mathfrak{c})$. Let $m_1\in M(d,\mathfrak{c})$, let $i_1>N_{m_1}$ with $d_{i_1}^{m_1}=d$ and $\mathrm{CP}_{m_1}(i_1)=\mathfrak{c}$ and let $\varepsilon_1=\mathrm{Val}(r_{m_1})$. Since $M(d,\mathfrak{c})$ is infinite, there is $m_2\in M(d,\mathfrak{c})$ with $m_2^{-1}<\varepsilon_1$. Then we have $\mathrm{Val}(r_{m_2})<m_2^{-1}<\varepsilon_1=\mathrm{Val}(r_{m_1})$. Let $i_2>N_{m_2}$ with $d_{i_2}^{m_2}=d$ and $\mathrm{CP}_{m_2}(i_2)=\mathfrak{c}$ and let $\varepsilon_2=\mathrm{Val}(r_{m_2})$. We know that the associated weight-sequence of r_{m_1} is constantly ε_1^{-1} and that of r_{m_2} is constantly ε_2^{-1}.

Now consider the sequence of transitions $r=d_0^{m_1}\ldots d_{i_1}^{m_1}d_{i_2+1}^{m_2}d_{i_2+2}^{m_2}\ldots$, we claim that r is an accepting run of $\mathcal{A}$ on a^ω and that $\mathrm{Val}(r)=-1$. The first state is initial as r_{m_1} is an accepting run and the transitions are well matched as $d_{i_1}^{m_1}=d_{i_2}^{m_2}$. That the Büchi acceptance condition is fulfilled and infinitely often some counter is activated is clear as r_{m_2} is an accepting run. Finally, that the starts and stops form well matched pairs follows from the fact that $\mathrm{CP}_{m_1}(i_1)=\mathrm{CP}_{m_2}(i_2)$.

To see that $\mathrm{Val}(r)=-1$, note that $i_1>N_{m_1}$, i.e., there is at least one counter in r which is started at a position $l_1<i_1$ and also terminated before position i_1. Therefore, for the weight-sequence $(z_i)_{i\geq 0}$ associated to r, we have $z_{l_1}=\varepsilon_1^{-1}$. Furthermore, there must also be a counter that is started at a position $l_2>i_2$ in r_{m_2}, which means $z_{l_2-i_2+i_1}=\varepsilon_2^{-1}$. Since $\varepsilon_2<\varepsilon_1$, $(z_i)_{i\geq 0}$ is not constant and therefore $\mathrm{Val}(r)=-1$. It follows that $[\![\mathcal{A}]\!](a^\omega)=-1$, which is a contradiction to the choice of $\mathcal{A}$. □

5 The main result

In this section, we want to show that the MC-recognizable series coincide with the series definable by monitor MSO formulas from our logic. In Lemma 14, we show how a given MMCA can be described by a monitor MSO formula. To show that every series definable by a monitor MSO formula is also MC-recognizable, we show by induction on the structure of the formula how to construct an MMCA whose behavior coincides with the semantics of the formula. We first formulate our main theorem.

Theorem 13. *Let* Σ *be an alphabet and* Val *an* ω*-valuation function. A series* $S\colon \Sigma^\omega \to \mathbb{R} \cup \{-\infty, \infty\}$ *is* Val*-MC-recognizable if and only if there exists a monitor MSO sentence* $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ *with* $[\![\varphi]\!] = S$.

In the following lemma, we show the first direction, namely how to obtain a formula for a given MMCA.

Lemma 14. *For every* Val*-MMCA* $\mathcal{A}$ *over* Σ*, there exists a sentence* $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ *with* $[\![\mathcal{A}]\!] = [\![\varphi]\!]$.

Proof. For first order variables x and y and second order variables $X_1, \dots, X_k$ we define the MSO formulas

$$\begin{aligned}
\mathrm{first}(x) &= \forall y. x \leq y \\
(x < y) &= x \leq y \wedge \neg(y \leq x) \\
(y = x+1) &= x < y \wedge \forall z. (z \leq x \vee y \leq z) \\
\mathrm{partition}(X_1, \dots, X_k) &= \forall x. \bigvee_{i=1}^{k} \left(x \in X_i \wedge \bigwedge_{j \neq i} \neg(x \in X_j) \right).
\end{aligned}$$

Now let $\mathcal{A} = (Q, \Sigma, I, \delta, \mathcal{F}, n, \mathrm{Val})$ be an n-MMCA. For every $(p, a, q, \bar{u}) \in \delta$ we choose a second order variable $X_{(p,a,q,\bar{u})}$ and with $k = |\delta|$ we fix a bijection $v\colon \{1, \dots, k\} \to \delta$. For $i \in \{1, \dots, k\}$ we write X_i for $X_{v(i)}$ and $\bar{X}$ for $(X_1, \dots, X_k)$. Furthermore, we fix second order variables $Y_1, \dots, Y_n$ and write $\bar{Y}$ for $(Y_1, \dots, Y_n)$. For $j \in \{1, \dots, n\}$ and $\star \in \{s, t\}$ we abbreviate

$$(u_j(x) = \star) = \bigvee_{\substack{(p,a,q,\bar{u}) \in \delta \\ u_j = \star}} x \in X_{(p,a,q,\bar{u})}.$$

Intuitively, we use the variables $\bar{X}$ to encode runs, i.e., by assigning the transition $v(i)$ to every position in X_i. The variables $\bar{Y}$ are used to mark the starts and stops of the counters in the run $\bar{X}$. In the following, we define the MSO formula muller$(\bar{X})$ which checks that $\bar{X}$ encodes a run of $\mathcal{A}$ satisfying the Muller acceptance condition, and the MSO formula accept$(\bar{X})$ which checks that $\bar{X}$ encodes an accepting run. The MSO formula accept$^*(\bar{X}, \bar{Y})$ asserts that the positions in $\bar{Y}$ conform to the starts and stops of the counters in $\bar{X}$. The precise formulas are as follows.

$$\begin{aligned}
\mathrm{matched}(\bar{X}) = &\bigwedge_{(p,a,q,\bar{u}) \in \delta} \forall x. \big(x \in X_{(p,a,q,\bar{u})} \to P_a(x) \big) \\
&\wedge \forall x. \forall y. \left(y = x+1 \to \bigvee_{q \in Q} \left(\bigvee_{(p,a,q,\bar{u}),(q,a',p',\bar{u}') \in \delta} (x \in X_{(p,a,q,\bar{u})} \wedge y \in X_{(q,a',p',\bar{u}')}) \right) \right)
\end{aligned}$$

$$\begin{aligned}
\mathrm{muller}(\bar{X}) = &\,\mathrm{partition}(\bar{X}) \wedge \mathrm{matched}(\bar{X}) \wedge \exists x. \left(\mathrm{first}(x) \wedge \bigvee_{\substack{(p,a,q,\bar{u}) \in \delta \\ p \in I}} x \in X_{(p,a,q,\bar{u})} \right) \\
&\wedge \bigvee_{F \in \mathcal{F}} \left(\exists x. \forall y. x \leq y \to \left(\left(\bigvee_{\substack{(p,a,q,\bar{u}) \in \delta \\ q \in F}} y \in X_{(p,a,q,\bar{u})} \right) \wedge \bigwedge_{q \in F} \exists z. \left(y \leq z \wedge \bigvee_{(p,a,q,\bar{u}) \in \delta} z \in X_{(p,a,q,\bar{u})} \right) \right) \right)
\end{aligned}$$

$$\begin{aligned}
\mathrm{accept}(\bar{X}) = \mathrm{muller}(\bar{X}) \wedge \forall x.\exists y.(x \le y \wedge \bigvee_{j=1}^{n} u_j(y) = s) \\
\wedge \bigwedge_{j=1}^{n} \forall x.\Big(\Big((u_j(x) = s) \to \exists y.\Big(x < y \wedge u_j(y) = t \wedge \forall z.((x < z \wedge z < y) \to \neg(u_j(z) = s))\Big)\Big)\wedge \\
\Big((u_j(x) = t) \to \exists y.\Big(y < x \wedge u_j(y) = s \wedge \forall z.((y < z \wedge z < x) \to \neg(u_j(z) = t))\Big)\Big)\Big)
\end{aligned}$$

$$\mathrm{accept}^*(\bar{X}, \bar{Y}) = \mathrm{accept}(\bar{X}) \wedge \bigwedge_{j=1}^{n} \forall x.(x \in Y_j \leftrightarrow (u_j(x) = s \vee u_j(x) = t)).$$

For $(p, a, q, \bar{u}) \in \delta$ and $j \in \{1, \ldots, n\}$, we let $\mathrm{wt}_j(p, a, q, \bar{u}) = u_j$ if $u_j \in \mathbb{Z}$, and $\mathrm{wt}_j(p, a, q, \bar{u}) = 0$ otherwise. Then for $i \in \{1, \ldots, k-2\}$ and $j \in \{1, \ldots, n\}$ we define inductively

$$\begin{aligned}
\psi_{k-1}^{j} &= (y \in X_{k-1} \mathbin{?} \mathrm{wt}_j(v(k-1)) : \mathrm{wt}_j(v(k))) \\
\psi_i^j &= \Big(y \in X_i \mathbin{?} \mathrm{wt}_j(v(i)) : \psi_{i+1}^j\Big) \\
\zeta_{n+1} &= \mathbb{1} \\
\zeta_j &= \Big((u_j(x) = s) \mathbin{?} \bigoplus\nolimits^{x,Y_j} y.\psi_1^j : \zeta_{j+1}\Big).
\end{aligned}$$

Then with

$$\varphi = \inf \bar{X}. \inf \bar{Y}.(\mathrm{accept}^*(\bar{X}, \bar{Y}) \mathbin{?} \mathrm{Val}\, x.\zeta_1 : \infty),$$

we have $[\![\mathcal{A}]\!] = [\![\varphi]\!]$. The formula ψ_1^j evaluates to the weight for counter j of the transition at position y, i.e., it is $\mathrm{wt}_j(v(i))$ iff y is in X_i. The formula ζ_1 evaluates to the output of the counter started at position x in the run encoded by $\bar{X}$. More precisely, ζ_1 evaluates to $\bigoplus^{x,Y_j} y.\psi_1^j$ if counter j is started at position x, and to $\mathbb{1}$ if no counter is started at x. Finally, the formula φ takes the infimum over the weights of all runs $\bar{X}$, in the sense that assignments to $\bar{X}$ and $\bar{Y}$ only influence the value of φ if $\bar{X}$ encodes an accepting run and $\bar{Y}$ mirrors its counter starts and stops. □

The remainder of this section is dedicated to showing the converse, namely, that for every monitor MSO formula there exists an MMCA whose behavior coincides with the semantics of the formula. Let Σ be an alphabet and Val an ω-valuation function. We proceed by induction. For the base case, we show that for an x-summing formula $\zeta \in \mathrm{mMSO}^x(\Sigma)$, the semantics of $\mathrm{Val}\, x.\zeta$ is Val-MC-recognizable. For the inductive part, we show that if we have mMSO formulas $\varphi_1, \varphi_2 \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ whose semantics are Val-MC-recognizable and an MSO formula $\beta \in \mathrm{MSO}(\Sigma)$, then the semantics of $\beta \mathbin{?} \varphi_1 : \varphi_2$, $\min(\varphi_1, \varphi_2)$, $\inf x.\varphi_1$, and $\inf X.\varphi_1$ are all recognizable. We will actually show the base case last, as it has the most involved proof. We begin with the Val-MC-recognizability of $\beta \mathbin{?} \varphi_1 : \varphi_2$.

Lemma 15. *Let $\beta \in \mathrm{MSO}(\Sigma)$ be an MSO formula and $\varphi_1, \varphi_2 \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ such that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are Val-MC-recognizable. Then with $\varphi = \beta \mathbin{?} \varphi_1 : \varphi_2$, the series $[\![\varphi]\!]$ is also Val-MC-recognizable.*

Proof. Let $\mathcal{V} = \mathrm{Free}(\varphi)$. Then we have $\mathrm{Free}(\varphi_1) \subseteq \mathcal{V}$ and $\mathrm{Free}(\varphi_2) \subseteq \mathcal{V}$ and hence by Lemma 10 $[\![\varphi_1]\!]_{\mathcal{V}}$ and $[\![\varphi_2]\!]_{\mathcal{V}}$ are Val-MC-recognizable. Due to $\mathrm{Free}(\beta) \subseteq \mathcal{V}$, the classical Büchi theorem tells us that both $\mathcal{L}_{\mathcal{V}}(\beta)$ and $\mathcal{L}_{\mathcal{V}}(\neg\beta)$ are ω-recognizable. Hence by Lemma 5 and Lemma 6, $[\![\varphi]\!] = \min(\mathcal{L}_{\mathcal{V}}(\beta) \cap [\![\varphi_1]\!]_{\mathcal{V}}, \mathcal{L}_{\mathcal{V}}(\neg\beta) \cap [\![\varphi_2]\!]_{\mathcal{V}})$ is also Val-MC-recognizable. □

Next, we show that for two mMSO formulas φ_1 and φ_2 whose semantics are Val-MC-recognizable, the semantics of $\min(\varphi_1, \varphi_2)$ is also Val-MC-recognizable.

Lemma 16. *Let $\varphi_1, \varphi_2 \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ be such that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are Val-MC-recognizable. Then with $\varphi = \min(\varphi_1, \varphi_2)$, the series $[\![\varphi]\!]$ is also Val-MC-recognizable.*

Proof. Let $\mathcal{V} = \mathrm{Free}(\varphi_1) \cup \mathrm{Free}(\varphi_2)$, then by Lemma 10, $[\![\varphi_1]\!]_{\mathcal{V}}$ and $[\![\varphi_2]\!]_{\mathcal{V}}$ are Val-MC-recognizable. Hence by Lemma 5, $[\![\varphi]\!] = \min([\![\varphi_1]\!]_{\mathcal{V}}, [\![\varphi_2]\!]_{\mathcal{V}})$ is also Val-MC-recognizable. □

For the last step of the inductive part, we show that for an mMSO formula φ whose semantics is Val-MC-recognizable, the semantics of $\inf x.\varphi$ and $\inf X.\varphi$ are also Val-MC-recognizable.

Lemma 17. *Let $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ such that $[\![\varphi]\!]$ is* Val*-MC-recognizable. Then the series $[\![\inf x.\varphi]\!]$ and $[\![\inf X.\varphi]\!]$ are also* Val*-MC-recognizable.*

Proof. We show the lemma for $\inf x.\varphi$. The proof for $\inf X.\varphi$ is similar. Let $\mathcal{V} = \mathrm{Free}(\inf x.\varphi)$, then $x \notin \mathcal{V}$. We consider the homomorphism

$$h \colon \Sigma^{\omega}_{\mathcal{V}\cup\{x\}} \to \Sigma^{\omega}_{\mathcal{V}}$$

which erases the x-row. Then for every valid $(w, \rho) \in \Sigma^{\omega}_{\mathcal{V}}$, we have that

$$[\![\inf x.\varphi]\!]_{\mathcal{V}}(w, \rho) = \inf\{[\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(w, \rho[x \to i]) \mid i \geq 0\} = h([\![\varphi]\!]_{\mathcal{V}\cup\{x\}})(w, \rho).$$

As $\mathrm{Free}(\varphi) \subseteq \mathcal{V} \cup \{x\}$, Lemma 10 shows that $[\![\varphi]\!]_{\mathcal{V}\cup\{x\}}$ is Val-MC-recognizable and therefore by Lemma 4 (i), the series $[\![\inf x.\varphi]\!]_{\mathcal{V}} = h([\![\varphi]\!]_{\mathcal{V}\cup\{x\}})$ is Val-MC-recognizable as well. $\square$

Before we turn to the proof of the base case, we prove two technical lemmata for almost Boolean and x-summing formulas.

Lemma 18. *Let $\psi \in \mathrm{mMSO}^{\text{a-bool}}(\Sigma)$ be an almost Boolean formula and $\mathcal{V} \supseteq \mathrm{Free}(\psi)$. Then there exist MSO formulas $\beta_1, \ldots, \beta_n \in \mathrm{MSO}(\Sigma)$ and weights $z_1, \ldots, z_n \in \mathbb{Z}$ such that $\mathrm{Free}(\psi) = \bigcup_{i=1}^{n} \mathrm{Free}(\beta_i)$, $N_{\mathcal{V}} = \bigcup_{i=1}^{n} \mathcal{L}_{\mathcal{V}}(\beta_i)$, for $i \neq j$ we have $\mathcal{L}_{\mathcal{V}}(\beta_i) \cap \mathcal{L}_{\mathcal{V}}(\beta_j) = \emptyset$, and for $(w, \rho) \in N_{\mathcal{V}}$ we have $[\![\psi]\!]_{\mathcal{V}}(w, \rho) = z_i$ if and only if $(w, \rho) \in \mathcal{L}_{\mathcal{V}}(\beta_i)$.*

Proof. For $\psi = k$ with $k \in \mathbb{Z}$, we choose $n = 1$, β_1 as any tautology, for example $\beta_1 = \exists x.x \leq x$, and $z_1 = k$.

For $\psi = \beta \mathbin{?} \psi_1 : \psi_2$ we assume that the lemma is true for ψ_1 with $\beta_1^{(1)}, \ldots, \beta_{n_1}^{(1)}$ and $z_1^{(1)}, \ldots, z_{n_1}^{(1)}$ and for ψ_2 with $\beta_1^{(2)}, \ldots, \beta_{n_2}^{(2)}$ and $z_1^{(2)}, \ldots, z_{n_2}^{(2)}$. Then for ψ we let $n = n_1 + n_2$ and choose $\beta_1, \ldots, \beta_{n_1+n_2}$ and $z_1, \ldots, z_{n_1+n_2}$ as follows. For $i \in \{1, \ldots, n_1\}$, we let $\beta_i = \beta \wedge \beta_i^{(1)}$ and $z_i = z_i^{(1)}$, and for $i \in \{1, \ldots, n_2\}$, we let $\beta_{n_1+i} = \neg\beta \wedge \beta_i^{(2)}$ and $z_{n_1+i} = z_i^{(2)}$. $\square$

Lemma 19. *Let $\zeta \in \mathrm{mMSO}^{x}(\Sigma)$ be an x-summing formula and $\mathcal{V} \supseteq \mathrm{Free}(\zeta)$. Then there exist MSO formulas $\beta_1, \ldots, \beta_n \in \mathrm{MSO}(\Sigma)$ and formulas $\zeta_1, \ldots, \zeta_n$ with $\zeta_i = \bigoplus^{x,Y_i} y.\psi_i$, where ψ_i is almost Boolean, such that $\mathrm{Free}(\zeta) = \bigcup_{i=1}^{n} \mathrm{Free}(\beta_i) \cup \mathrm{Free}(\zeta_i)$, for $i \neq j$ we have $\mathcal{L}_{\mathcal{V}}(\beta_i) \cap \mathcal{L}_{\mathcal{V}}(\beta_j) = \emptyset$, for $(w, \rho) \in N_{\mathcal{V}}$ we have $[\![\zeta]\!]_{\mathcal{V}}(w, \rho) = [\![\zeta_i]\!]_{\mathcal{V}}(w, \rho)$ if and only if $(w, \rho) \in \mathcal{L}_{\mathcal{V}}(\beta_i)$, and if $(w, \rho) \notin \bigcup_{i=1}^{n} \mathcal{L}_{\mathcal{V}}(\beta_i)$ then $[\![\zeta]\!]_{\mathcal{V}}(w, \rho) = \mathbb{1}$. We may assume the variables Y_i to be pairwise distinct.*

Proof. We proceed like in the proof of Lemma 18. For $\zeta = \mathbb{1}$ we choose $n = 0$, i.e., there are no formulas β. For $\zeta = \bigoplus^{x,X} y.\psi$, we choose β_1 as any tautology and $\zeta_1 = \zeta$.

For $\zeta = \beta \mathbin{?} \zeta_1' : \zeta_2'$, we assume that the lemma is true for ζ_1' with $\beta_1^{(1)}, \ldots, \beta_{n_1}^{(1)}$ and $\zeta_1^{(1)}, \ldots, \zeta_{n_1}^{(1)}$ and for ζ_2' with $\beta_1^{(2)}, \ldots, \beta_{n_2}^{(2)}$ and $\zeta_1^{(2)}, \ldots, \zeta_{n_2}^{(2)}$. Then for ζ we let $n = n_1 + n_2$ and choose $\beta_1, \ldots, \beta_{n_1+n_2}$ and $\zeta_1, \ldots, \zeta_{n_1+n_2}$ as follows. For $i \in \{1, \ldots, n_1\}$ we let $\beta_i = \beta \wedge \beta_i^{(1)}$ and $\zeta_i = \zeta_i^{(1)}$, and for $i \in \{1, \ldots, n_2\}$ we let $\beta_{n_1+i} = \neg\beta \wedge \beta_i^{(2)}$ and $\zeta_{n_1+i} = \zeta_i^{(2)}$.

Now in case that for some $i \neq j$ we have $Y_i = Y_j$, we replace β_i and β_j by one formula $\beta' = \beta_i \vee \beta_j$ and we replace ζ_i and ζ_j by $\zeta' = \bigoplus^{x,Y_i} y.(\beta_i \mathbin{?} \psi_i : \psi_j)$. $\square$

We now turn to the proof of the base case of our induction.

Theorem 20. *Let $\zeta \in \mathrm{mMSO}^{x}(\Sigma)$ be an x-summing formula. Then $[\![\mathrm{Val}\, x.\zeta]\!]$ is* Val*-MC-recognizable.*

Proof. We adapt and expand ideas from [6, 9]. Let $\beta_1, \ldots, \beta_n$ and $\zeta_1, \ldots, \zeta_n$ be the formulas we can find for ζ according to Lemma 19. We write $\zeta_i = \bigoplus^{x,Y_i} y.\psi_i$. Then for each $i \in \{1, \ldots, n\}$, let $\beta_{i1}, \ldots, \beta_{in_i}$ and $z_{i1}, \ldots, z_{in_i}$ be the formulas and weights we can find for ψ_i according to Lemma 18.

The proof idea is as follows. For $\mathcal{V} = \mathrm{Free}(\mathrm{Val}\, x.\zeta)$, the mapping $[\![\mathrm{Val}\, x.\zeta]\!]$ assigns values to words from $\Sigma^{\omega}_{\mathcal{V}}$. Consider a valid $(w, \rho) \in \Sigma^{\omega}_{\mathcal{V}}$. We can interpret each ζ_i as a counter which is stopped and then restarted at the k-th letter of w depending on whether $(w, \rho[x \to k])$ satisfies β_i. As our automata cannot stop and start a single counter at the same time, each counter i will correspond to two counters i and i' in the automaton we construct. The computations of counter i depend on $\beta_{i1}, \ldots, \beta_{in_i}$. We extend the alphabet $\Sigma_{\mathcal{V}}$ by adding two entries for each counter to each letter in $\Sigma_{\mathcal{V}}$. The entries for counter i can contain an s to indicate the start of the counter, a t to indicate a stop, a number $j \in \{1, \ldots, n_i\}$ to indicate that the counter is active and should add z_{ij} to its current value, or the new symbol $\perp$ to indicate that the counter is inactive. Let $\tilde{\Sigma}_{\mathcal{V}}$ be this new alphabet. We show that we can define an ω-recognizable language L over $\tilde{\Sigma}_{\mathcal{V}}$ which for every word has all information about the counter operations encoded in the word. For example, if $(w, \rho[x \to k]) \models \beta_i$, then in the word $(w, \rho, v) \in \tilde{\Sigma}_{\mathcal{V}}^{\omega}$ corresponding to (w, ρ), the entry for counter i in the k-th letter should contain an s. Then if $(w, \rho[x \to k, y \to k+1]) \models \beta_{ij}$, the i-entry of the $k+1$-th letter should contain a j. The precise formulation is involved and will be formalized in the sequel.

When we have shown that the language L is ω-recognizable, we can construct a Muller automaton $\tilde{\mathcal{A}}$ which recognizes L. Turning $\tilde{\mathcal{A}}$ into an MMCA and applying a projection, we finally obtain the recognizability of $[\![\mathrm{Val}\, x.\zeta]\!]$.

If $n = 0$ then $[\![\zeta]\!] \equiv \mathbb{1}$, i.e., $[\![\mathrm{Val}\, x.\zeta]\!] \equiv \infty$, which is recognized by every BMCA without final states. Assume $n > 0$ and let $\mathcal{W} = \mathrm{Free}(\zeta)$, then according to Lemma 19 we have

$$\begin{aligned} \mathcal{W} &= \bigcup_{i=1}^{n} \mathrm{Free}(\beta_i) \cup \mathrm{Free}(\zeta_i) \\ &= \bigcup_{i=1}^{n} \mathrm{Free}(\beta_i) \cup \{x, Y_i\} \cup (\mathrm{Free}(\psi_i) \setminus \{y\}). \end{aligned}$$

In particular, we have $\mathcal{W} \supseteq \{x, Y_1, \ldots, Y_n\}$ and for every $i \in \{1, \ldots, n\}$, we have $\mathcal{W} \supseteq \mathrm{Free}(\beta_i)$. According to the classical Büchi theorem, we therefore know that $\mathcal{L}_{\mathcal{W}}(\beta_i)$ is ω-recognizable. This in turn means that there are MSO sentences β_i' over the alphabet $\Sigma_{\mathcal{W}}$ with $\mathcal{L}_{\mathcal{W}}(\beta_i) = \mathcal{L}(\beta_i')$.

Let $H = (\{s, t, \perp, 1, \ldots, n_1\} \times \ldots \times \{s, t, \perp, 1, \ldots, n_n\})^2$, where $\perp$ is a new symbol. An element $h \in H$ can be interpreted as a mapping with $\mathrm{dom}(h) = \{1, \ldots, 2n\}$ such that for $i \in \{1, \ldots, n\}$ we have $h(i), h(i+n) \in \{s, t, \perp, 1, \ldots, n_i\}$. We now consider a new alphabet $\tilde{\Sigma}_{\mathcal{W}} = \Sigma_{\mathcal{W}} \times H$. We represent letters from $\tilde{\Sigma}_{\mathcal{W}}$ as triples (a, g, h) where $a \in \Sigma$, $g \in \{0,1\}^{\mathcal{W}}$, and $h \in H$. Infinite words over $\tilde{\Sigma}_{\mathcal{W}}$ are represented as triples (w, ρ, v) where $(w, \rho) \in \Sigma_{\mathcal{W}}^{\omega}$ and $v \colon \mathbb{N} \to H$.

We transform each β_i' in the following fashion. We obtain β_i'' from β_i' by replacing each atomic formula $P_{(a,g)}(z)$ in β_i' by $\bigvee_{h \in H} P_{(a,g,h)}(z)$. Then for every $(w, \rho, v) \in \tilde{\Sigma}_{\mathcal{W}}^{\omega}$ we have $(w, \rho, v) \models \beta_i''$ if and only if $(w, \rho) \models \beta_i'$.

Now let $\mathcal{V} = \mathcal{W} \setminus \{x\}$. We transform β_i'' into a formula $\beta_i'''(x)$ over the alphabet $\tilde{\Sigma}_{\mathcal{V}} = \Sigma_{\mathcal{V}} \times H$ as follows. Each atomic subformula $P_{(a,g,h)}(z)$ in β_i'' is replaced by $P_{(a,g',h)}(z) \wedge x = z$ if $g(x) = 1$ and by $P_{(a,g',h)}(z) \wedge \neg(x = z)$ if $g(x) = 0$, where g' is the restriction of g to $\mathcal{V}$, i.e., $g' = g{\restriction_{\mathcal{V}}}$. Then $\beta_i'''(x)$ has exactly one free variable, namely x, as β_i' is a sentence. Let $k \in \mathbb{N}$ and $(w, \rho, v) \in \tilde{\Sigma}_{\mathcal{V}}^{\omega}$. Then $((w, \rho, v), [x \to k]) \models \beta_i'''(x)$ if and only if $(w, \rho[x \to k], v) \models \beta_i''$.

Now let $\mathcal{W}' = \mathcal{W} \cup \{y\}$, then by Lemma 18 we have

$$\begin{aligned} \mathcal{W}' &\supseteq \{y\} \cup \bigcup_{i=1}^{n} (\mathrm{Free}(\psi_i) \setminus \{y\}) \\ &= \{y\} \cup \bigcup_{i=1}^{n} \bigcup_{j=1}^{n_i} \mathrm{Free}(\beta_{ij}). \end{aligned}$$

With the same argumentation as above, we find MSO sentences β_{ij}' over the alphabet $\Sigma_{\mathcal{W}'}$ with $\mathcal{L}_{\mathcal{W}'}(\beta_{ij}) = \mathcal{L}(\beta_{ij}')$. Again, we obtain from each β_{ij}' a sentence β_{ij}'' over the alphabet $\tilde{\Sigma}_{\mathcal{W}'} = \Sigma_{\mathcal{W}'} \times H$ by replacing every atomic formula $P_{(a,g)}(z)$ in β_{ij}' by $\bigvee_{h \in H} P_{(a,g,h)}(z)$. Then for every $(w, \rho, v) \in \tilde{\Sigma}_{\mathcal{W}'}^{\omega}$ we have $(w, \rho, v) \models \beta_{ij}''$ if and only if $(w, \rho) \models \beta_{ij}'$.

Next, we obtain from β_{ij}'' a formula $\beta_{ij}'''(x, y)$ over the alphabet $\tilde{\Sigma}_{\mathcal{V}}$ as follows. Each atomic subformula $P_{(a,g,h)}(z)$ in β_{ij}'' is replaced by

- $P_{(a,g',h)}(z) \wedge x = z \wedge y = z$ if $g(x) = 1$ and $g(y) = 1$
- $P_{(a,g',h)}(z) \wedge x = z \wedge \neg(y = z)$ if $g(x) = 0$ and $g(y) = 1$
- $P_{(a,g',h)}(z) \wedge \neg(x = z) \wedge y = z$ if $g(x) = 1$ and $g(y) = 0$
- $P_{(a,g',h)}(z) \wedge \neg(x = z) \wedge \neg(y = z)$ if $g(x) = 0$ and $g(y) = 0$

where g' is the restriction of g to $\mathcal{V}$, i.e., $g' = g{\restriction_{\mathcal{V}}}$. Note the change of roles of x and y. Then $\beta_{ij}'''(x, y)$ has exactly two free variables, namely x and y, as β_{ij}' is a sentence. Let $k, l \in \mathbb{N}$ and $(w, \rho, v) \in \tilde{\Sigma}_{\mathcal{V}}^{\omega}$. Then $((w, \rho, v), [x \to k, y \to l]) \models \beta_{ij}'''(x, y)$ if and only if $(w, \rho[x \to l, y \to k], v) \models \beta_{ij}''$.

Now recall that $\{Y_1,\dots,Y_n\} \subseteq \mathcal{V}$. For $i \in \{1,\dots,n\}$, $\star \in \{s,t,1,\dots,n_i\}$, and $i' \in \{i, i+n\}$, we define the abbreviations

$$\begin{aligned}
Y_i(z) &= \bigvee_{\substack{(a,g,h)\in\tilde{\Sigma}_{\mathcal{V}}\\ g(Y_i)=1}} P_{(a,g,h)}(z)\\
(h_{i'}(z)=\star) &= \bigvee_{\substack{(a,g,h)\in\tilde{\Sigma}_{\mathcal{V}}\\ h(i')=\star}} P_{(a,g,h)}(z)\\
\text{even}(x,X) &= \exists Y.\exists Z.\forall x'.((x'\in X \wedge x' < x) \to (x'\in Y \leftrightarrow \neg x' \in Z)) \wedge\\
&\qquad \forall x'.((x'\in Y \vee x'\in Z) \to (x'\in X \wedge x' < x)) \wedge\\
&\qquad \forall z.(z \in Z \to \exists y.(y\in Y \wedge y < z)) \wedge\\
&\qquad \forall y.(y\in Y \to \exists z.(z\in Z \wedge y < z)) \wedge\\
&\qquad \forall y_1.\forall y_2.((y_1\in Y \wedge y_2 \in Y \wedge y_1 < y_2) \to \exists z.(z\in Z \wedge y_1 < z \wedge z < y_2)) \wedge\\
&\qquad \forall z_1.\forall z_2.((z_1\in Z \wedge z_2 \in Z \wedge z_1 < z_2) \to \exists y.(y\in Y \wedge z_1 < y \wedge y < z_2)).
\end{aligned}$$

For every $w \in \Sigma^\omega$, we have $(w,[x \to k, X \to I]) \models \text{even}(x,X)$ if and only if $\{j \in I \mid j < k\}$ contains evenly many elements. In the following, we will also use the formula $\text{notLast}(x,X)$ defined in Remark 8. Now for $i \in \{1,\dots,n\}$, $j \in \{1,\dots,n_i\}$, and $i' \in \{i,i+n\}$, we define

$$\begin{aligned}
\varphi_{is}(x) &= (h_i(x)=s) \leftrightarrow \Big(\beta_i'''(x) \wedge \exists X.\Big(\forall z.(z\in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x,X) \wedge \text{even}(x,X)\Big)\Big)\\
\varphi_{(i+n)s}(x) &= (h_{i+n}(x)=s) \leftrightarrow \Big(\beta_i'''(x) \wedge \exists X.\Big(\forall z.(z\in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x,X) \wedge \neg\text{even}(x,X)\Big)\Big)\\
\varphi_{i't}(x) &= (h_{i'}(x)=t) \leftrightarrow \Big(Y_i(x) \wedge \exists z.\Big((z<x) \wedge (h_{i'}(z)=s) \wedge \forall z'.\Big((z'<x \wedge Y_i(z')) \to z' \le z\Big)\Big)\Big)\\
\varphi_{i'j}(x) &= (h_{i'}(x)=j) \leftrightarrow \exists y.\exists y'.\Big((y<x)\wedge(h_{i'}(y)=s)\wedge(x<y')\wedge(h_{i'}(y')=t)\wedge\\
&\qquad\qquad \forall z.\Big(Y_i(z) \to (z\le y \vee y' \le z)\Big) \wedge \beta_{ij}'''(x,y)\Big)
\end{aligned}$$

and finally

$$\varphi = \left(\bigwedge_{i=1}^{2n} \forall x.\varphi_{is}(x)\right) \wedge \left(\bigwedge_{i=1}^{2n} \forall x.\varphi_{it}(x)\right) \wedge \left(\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{n_i} \forall x.(\varphi_{ij}(x) \wedge \varphi_{(i+n)j}(x))\right).$$

The formula φ is clearly a sentence over $\tilde{\Sigma}_{\mathcal{V}}$. In the following lemma, we show that $\mathcal{L}(\varphi)$ is exactly the language over $\tilde{\Sigma}_{\mathcal{V}}$ we described in the explanation at the beginning of the proof.

Lemma 21. *Let $(w,\rho) \in \Sigma_{\mathcal{V}}^\omega$ be valid, then there exists exactly one mapping $v\colon \mathbb{N} \to H$ such that $(w,\rho,v) \models \varphi$ and for this v we have the following. For all $k \in \mathbb{N}$, either $v(k)(i') \neq s$ for all $i' \in \{1,\dots,2n\}$ and $[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k]) = \mathbb{1}$, or we have $v(k)(i') = s$ for exactly one $i' \in \{1,\dots,2n\}$ and with $l = \min\{\iota > k \mid v(\iota)(i') = t\}$ we have*

$$[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k]) = \sum_{\iota=k+1}^{l-1} z_{iv(\iota)(i')},$$

where $i \in \{1,\dots,n\}$ is such that $i' \in \{i,i+n\}$. In particular, $\{\iota > k \mid v(\iota)(i') = t\} \neq \emptyset$ in the latter case. Furthermore, for this v and all $l \in \mathbb{N}$ we have that if $v(l)(i') = t$ for some $i' \in \{1,\dots,2n\}$, then for some $k < l$ we have $v(k)(i') = s$.

Proof. **Step 1:** We show the uniqueness first, so let $(w,\rho) \in \Sigma_{\mathcal{V}}^\omega$ be valid and $v_1, v_2\colon \mathbb{N} \to H$ such that $(w,\rho,v_1) \models \varphi$ and $(w,\rho,v_2) \models \varphi$. Let $k \in \mathbb{N}$, $i \in \{1,\dots,n\}$, and $i' \in \{i,i+n\}$. We then have five different cases: (1) $v_1(k)(i') = s$ with $i' \le n$, (2) $v_1(k)(i') = s$ with $i' > n$, (3) $v_1(k)(i') = t$, (4) $v_1(k)(i') \in \mathbb{N}$, and (5) $v_1(k)(i') = \perp$. Assume the first case is true. Then we know that both $((w,\rho,v_1),[x\to k]) \models (h_{i'}(x) = s)$ and $((w,\rho,v_1),[x\to k]) \models \varphi_{i's}(x)$, which implies

$$((w,\rho,v_1),[x\to k]) \models \beta_i'''(x) \wedge \exists X.\Big(\forall z.(z\in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x,X) \wedge \text{even}(x,X)\Big). \tag{1}$$

As we have $((w, \rho, v_1), [x \to k]) \models \beta_i'''(x)$ if and only if $(w, \rho[x \to k]) \models \beta_i$, the validity of (1) does not depend on v_1. Hence, the formula in (1) is also satisfied by $((w, \rho, v_2), [x \to k])$. Since $((w, \rho, v_2), [x \to k]) \models \varphi_{i's}(x)$, we therefore must have $((w, \rho, v_2), [x \to k]) \models (h_{i'}(x) = s)$, i.e., $v_1(k)(i') = v_2(k)(i')$. With similar reasoning, cases (2), (3), and (4) yield the same result. For case (5) it then follows trivially that $v_2(k)(i') \neq \bot$ is impossible.

Step 2: We now construct a mapping $v \colon \mathbb{N} \to H$ with $(w, \rho, v) \models \varphi$. We assume that v is initialized with $\bot$, i.e., $v(k)(i') = \bot$ for all $k \in \mathbb{N}$ and $i' \in \{1, \dots, 2n\}$, and we will gradually redefine v. Let $k \in \mathbb{N}$ and $i \in \{1, \dots, n\}$. We define $i' = i$ if the set $\{j \in \rho(Y_i) \mid j < k\}$ contains evenly many elements, and $i' = i + n$ otherwise. If

$$((w, \rho, v), [x \to k]) \models \beta_i'''(x) \wedge \exists X. \Big(\forall z. (z \in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x, X)\Big), \tag{2}$$

we redefine $v(k)(i') = s$. Note that the validity of (2) does not depend on v. Doing this for all k and i clearly yields $(w, \rho, v) \models \bigwedge_{i=1}^{2n} \forall x. \varphi_{is}(x)$.

Now let $k \in \mathbb{N}$. Then for all $i \in \{1, \dots, n\}$ and $i' \in \{i, i+n\}$, we redefine $v(k)(i') = t$ if

$$((w, \rho, v), [x \to k]) \models Y_i(x) \wedge \exists z. \Big((z < x) \wedge (h_{i'}(z) = s) \wedge \forall z'. \Big((z' < x \wedge Y_i(z')) \to z' \leq z\Big)\Big). \tag{3}$$

We see as follows that $v(k)(i')$ was not redefined to s earlier. Let $l = \max\{j \in \rho(Y_i) \mid j < k\}$. Then we have by (3) that $v(l)(i') = s$. We know that $\{j \in \rho(Y_i) \mid j < k\}$ contains evenly many elements if and only if $\{j \in \rho(Y_i) \mid j < l\}$ contains an odd number of elements. This shows that $v(k)(i')$ and $v(l)(i')$ cannot both be s. Proceeding in the same way for all $k \in \mathbb{N}$, we obtain v such that $(w, \rho, v) \models \Big(\bigwedge_{i=1}^{2n} \forall x. \varphi_{is}(x)\Big) \wedge \Big(\bigwedge_{i=1}^{2n} \forall x. \varphi_{it}(x)\Big)$.

Finally, for $k \in \mathbb{N}$, $i \in \{1, \dots, n\}$, $i' \in \{i, i+n\}$, and $j \in \{1, \dots, n_i\}$ with

$$\begin{aligned} ((w, \rho, v), [x \to k]) \models \exists y. \exists y'. \Big(&(y < x) \wedge (h_{i'}(y) = s) \wedge (x < y') \wedge (h_{i'}(y') = t) \wedge \\ &\forall z. \Big(Y_i(z) \to (z \leq y \vee y' \leq z)\Big) \wedge \beta_{ij}'''(x, y)\Big), \end{aligned}$$

we redefine $v(k)(i') = j$. Since clearly $k \notin \rho(Y_i)$ in this case, $v(k)(i')$ was surely not redefined to s or t earlier. In conclusion, we obtain $(w, \rho, v) \models \varphi$.

Step 3: Assume $(w, \rho, v) \models \varphi$. First, assume that for some $l \in \mathbb{N}$ and $i' \in \{1, \dots, 2n\}$ we have $v(l)(i') = t$. Since $((w, \rho, v), [x \to l]) \models \varphi_{i't}(x)$, it easily follows that $v(k)(i') = s$ for some $k < l$.

Now let $k \in \mathbb{N}$. We show that either $v(k)(i') \neq s$ for all $i' \in \{1, \dots, 2n\}$ and $[\![\zeta]\!]_{\mathcal{V} \cup \{x\}}(w, \rho[x \to k]) = \mathbb{1}$, or we have $v(k)(i') = s$ for some $i \in \{1, \dots, n\}$ and $i' \in \{i, i+n\}$ and with $l = \min\{\iota > k \mid v(\iota)(i') = t\}$ we have

$$[\![\zeta]\!]_{\mathcal{V} \cup \{x\}}(w, \rho[x \to k]) = \sum_{\iota = k+1}^{l-1} z_{iv(\iota)(i')}.$$

Step 3.1: According to the choice of $\beta_1, \dots, \beta_n$ and by Remark 8, we know that $[\![\zeta]\!]_{\mathcal{V} \cup \{x\}}(w, \rho[x \to k]) \neq \mathbb{1}$ if and only if there exists $i \in \{1, \dots, n\}$ such that

$$(w, \rho[x \to k]) \models \beta_i \wedge \text{notLast}(x, Y_i)$$

and in this case the index i is uniquely determined. We have $(w, \rho[x \to k]) \models \text{notLast}(x, Y_i)$ if and only if

$$((w, \rho, v), [x \to k]) \models \exists X. \Big(\forall z. (z \in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x, X)\Big),$$

and by construction we have $(w, \rho[x \to k]) \models \beta_i$ if and only if $((w, \rho, v), [x \to k]) \models \beta_i'''(x)$. We thus have

$$((w, \rho, v), [x \to k]) \models \beta_i'''(x) \wedge \exists X. \Big(\forall z. (z \in X \leftrightarrow Y_i(z)) \wedge \text{notLast}(x, X)\Big)$$

if and only if $(w, \rho[x \to k]) \models \beta_i \wedge \text{notLast}(x, Y_i)$. Since $((w, \rho, v), [x \to k]) \models \varphi_{is}(x) \wedge \varphi_{(i+n)s}(x)$, this is the case if and only if either $v(k)(i) = s$ or $v(k)(i+n) = s$ holds, and both $v(k)(i) = s$ and $v(k)(i+n) = s$ can never hold.

In conclusion, for every $k \in \mathbb{N}$ there is always at most one $i' \in \{1, \dots, 2n\}$ with $v(k)(i') = s$, and furthermore $[\![\zeta]\!]_{\mathcal{V} \cup \{x\}}(w, \rho[x \to k]) \neq \mathbb{1}$ if and only if $v(k)(i') = s$ for some i'.

Step 3.2: Now assume $[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k]) \neq \mathbb{1}$ and let $i\in\{1,\ldots,n\}$ and $i'\in\{i,i+n\}$ with $v(k)(i')=s$ as in Step 3.1. We know that

$$\begin{aligned}[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k]) &= [\![\zeta_i]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k])\\ &= [\![\textstyle\bigoplus^{x,Y_i} y.\psi_i]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k])\\ &= \sum_{\iota=k+1}^{\min\{j\in\rho(Y_i)\mid j>k\}-1} [\![\psi_i]\!]_{\mathcal{V}\cup\{x,y\}}(w,\rho[x\to k,y\to\iota]).\end{aligned}$$

Let $l=\min\{j\in\rho(Y_i)\mid j>k\}$. We show that $l=\min\{\iota>k\mid v(\iota)(i')=t\}$. We have

$$((w,\rho,v),[x\to l]) \models Y_i(x)\wedge \exists z.\Big((z<x)\wedge(h_{i'}(z)=s)\wedge\forall z'.\Big((z'<x\wedge Y_i(z'))\to z'\le z\Big)\Big). \quad (4)$$

Due to $(w,\rho,v)\models\varphi_{i't}(x)$, we obtain $v(l)(i')=t$. Now assume there was l' with $k<l'<l$ and $v(l')(i')=t$, then $((w,\rho,v),[x\to l'])\models(h_{i'}(x)=t)$ and therefore (4) above is also satisfied for l'. In particular, $((w,\rho,v),[x\to l'])\models Y_i(x)$, but $l'\in\rho(Y_i)$ is impossible by definition of l.
Step 3.3: Now let ι with $k<\iota<l$, we show that

$$z_{iv(\iota)(i')} = [\![\psi_i]\!]_{\mathcal{V}\cup\{x,y\}}(w,\rho[x\to k,y\to\iota]).$$

By choice of $\beta_{i1},\ldots,\beta_{in_i}$ there is exactly one $j\in\{1,\ldots,n_i\}$ with $(w,\rho[x\to k,y\to\iota])\models\beta_{ij}$, which is equivalent to $((w,\rho,v),[x\to\iota,y\to k])\models\beta'''_{ij}(x,y)$. Due to Steps 3.1 and 3.2, we therefore see that $((w,\rho,v),[x\to\iota])$ models

$$\exists y.\exists y'.\Big((y<x)\wedge(h_{i'}(y)=s)\wedge(x<y')\wedge(h_{i'}(y')=t)\wedge\forall z.\Big(Y_i(z)\to(z\le y\vee y'\le z)\Big)\wedge\beta'''_{ij}(x,y)\Big),$$

which due to $(w,\rho,v)\models\varphi_{i'j}(x)$ means that $v(\iota)(i')=j$. We obtain

$$[\![\psi_i]\!]_{\mathcal{V}\cup\{x,y\}}(w,\rho[x\to k,y\to\iota]) = z_{ij} = z_{iv(\iota)(i')}.$$

□

Let $\tilde{\mathcal{A}}=(Q,\tilde{\Sigma}_{\mathcal{V}},q_0,\tilde{\delta},\mathcal{F})$ be a Muller automaton which accepts $\mathcal{L}(\varphi)$. We construct the MMCA $\mathcal{A}=(Q,\tilde{\Sigma}_{\mathcal{V}},\{q_0\},\delta,\mathcal{F},2n,\mathrm{Val})$ by defining δ as follows. The set δ contains all transitions $(p,(a,g,h),q,\bar{u})$ such that (1) $(p,(a,g,h),q)\in\tilde{\delta}$ and (2) for all $i\in\{1,\ldots,n\}$ and $i'\in\{i,i+n\}$ we have

$$u_{i'} = \begin{cases} s & \text{if } h(i')=s\\ t & \text{if } h(i')=t\\ z_{ij} & \text{if } h(i')=j\\ 0 & \text{if } h(i')=\bot.\end{cases}$$

By Lemma 21, we see that each transition starts at most one counter.

We show that now for every $(w,\rho,v)\in\tilde{\Sigma}^\omega_{\mathcal{V}}$ we have

$$[\![\mathcal{A}]\!](w,\rho,v) = \begin{cases}[\![\mathrm{Val}\,x.\zeta]\!](w,\rho) & \text{if } (w,\rho,v)\models\varphi\\ \infty & \text{otherwise.}\end{cases}$$

If $(w,\rho,v)\not\models\varphi$, then $\mathrm{Acc}_{\tilde{\mathcal{A}}}(w,\rho,v)=\emptyset$ and thus by construction of δ we also have $\mathrm{Acc}_{\mathcal{A}}(w,\rho,v)=\emptyset$, i.e., $[\![\mathcal{A}]\!](w,\rho,v)=\infty$.

Conversely, assume $(w,\rho,v)\models\varphi$. If $\mathrm{Acc}_{\mathcal{A}}(w,\rho,v)=\emptyset$, we let $\tilde{r}\in\mathrm{Acc}_{\tilde{\mathcal{A}}}(w,\rho,v)$ be an accepting run of $\tilde{\mathcal{A}}$ on (w,ρ,v). By supplying vectors $\bar{u}$ to the transitions of $\tilde{r}$ in the obvious fashion, we obtain a run r of $\mathcal{A}$ on (w,ρ,v). It follows from Lemma 21 that the start and stop symbols s and t for the counters appear in well-formed pairs. Thus, r is accepting if and only if the set $\{k\in\mathbb{N}\mid v(k)(i')=s$ for some $i'\in\{1,\ldots,2n\}\}$ is infinite. Since $\mathrm{Acc}_{\mathcal{A}}(w,\rho,v)=\emptyset$, the run r is not accepting, so we see by Lemma 21 that $\{k\in\mathbb{N}\mid[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k])\neq\mathbb{1}\}$ is finite. Thus, we have $[\![\mathrm{Val}\,x.\zeta]\!](w,\rho)=\infty$. It follows that $[\![\mathcal{A}]\!](w,\rho,v)=[\![\mathrm{Val}\,x.\zeta]\!](w,\rho)$.

Finally, if $(w,\rho,v)\models\varphi$ and $\mathrm{Acc}_{\mathcal{A}}(w,\rho,v)\neq\emptyset$, we let $r\in\mathrm{Acc}_{\mathcal{A}}(w,\rho,v)$, $k\in\mathbb{N}$, and let $(z_j)_{j\ge0}$ be the weight-sequence associated to r. We show that $z_k=[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to k])$. If $z_k=\mathbb{1}$, then by construction of δ we have $v(k)(i')\neq s$ for all $i'\in\{1,\ldots,2n\}$. By Lemma 21 we thus have

$[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w, \rho[x \to k]) = \mathbb{1} = z_k$. If $z_k \neq \mathbb{1}$, we must have $v(k)(i') = s$ for some $i \in \{1, \ldots, n\}$ and $i' \in \{i, i+n\}$ by the definition of $(z_j)_{j\geq 0}$ and the definition of δ. Thus, with $l = \min\{\iota > k \mid v(\iota)(i') = t\}$ we have by Lemma 21 and the definition of δ that

$$[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w, \rho[x \to k]) = \sum_{\iota=k+1}^{l-1} z_{iv(\iota)(i')} = z_k.$$

Therefore, we have $\mathrm{Val}(r) = \mathrm{Val}((z_j)_{j\geq 0}) = [\![\mathrm{Val}\, x.\zeta]\!](w, \rho)$. Since $r \in \mathrm{Acc}_{\mathcal{A}}(w, \rho, v)$ was arbitrary, it follows that $[\![\mathcal{A}]\!](w, \rho, v) = \mathrm{Val}(r) = [\![\mathrm{Val}\, x.\zeta]\!](w, \rho)$.

To conclude, consider the projection $h\colon \tilde{\Sigma}_{\mathcal{V}} \to \Sigma_{\mathcal{V}}, (w, \rho, v) \mapsto (w, \rho)$. For every valid $(w, \rho) \in \Sigma_{\mathcal{V}}^{\omega}$, we know by Lemma 21 that there exists exactly one mapping $v\colon \mathbb{N} \to H$ with $(w, \rho, v) \models \varphi$. Thus we have

$$\begin{aligned} h([\![\mathcal{A}]\!])(w, \rho) &= \inf\{[\![\mathcal{A}]\!](w, \rho, v) \mid v\colon \mathbb{N} \to H\} \\ &= [\![\mathcal{A}]\!](w, \rho, v) && \text{for the unique } v \text{ with } (w, \rho, v) \models \varphi \\ &= [\![\mathrm{Val}\, x.\zeta]\!](w, \rho), \end{aligned}$$

so $h([\![\mathcal{A}]\!]) = [\![\mathrm{Val}\, x.\zeta]\!]$ holds. By Lemma 4, $h([\![\mathcal{A}]\!])$ is Val-MC-recognizable. $\square$

By combining Theorem 20 and Lemmata 15, 16, and 17, we obtain that the semantics of every mMSO formula $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ is Val-MC-recognizable. Together with Lemma 14, this concludes the proof of Theorem 13.

6 A Nivat Theorem for Quantitative Monitor Automata

Theorem 22. *Let Σ be an alphabet and* Val *an ω-valuation function. A series $S\colon \Sigma^{\omega} \to \mathbb{R} \cup \{-\infty, \infty\}$ is* Val*-MC-recognizable if and only if there exists an alphabet Γ, a mapping $h\colon \Gamma \to \Sigma$, a one-state* Val*-BMCA $\mathcal{A}'$ over Γ, and an ω-recognizable language $L \subseteq \Gamma^{\omega}$ such that $S = h(L \cap [\![\mathcal{A}']\!])$.*

Proof. Let S be Val-MC-recognizable and $\mathcal{A} = (Q, \Sigma, I, \delta, \mathcal{F}, n, \mathrm{Val})$ be an MMCA with $S = [\![\mathcal{A}]\!]$. We let $\Gamma = \{(p, a, q) \mid (p, a, q, \bar{u}) \in \delta \text{ for some } \bar{u} \in (\mathbb{Z} \cup \{s, t\})^n\}$, define h by $(p, a, q) \mapsto a$, and define a BMCA $\mathcal{A}' = (\{*\}, \Gamma, \{*\}, \delta', \{*\}, n, \mathrm{Val})$ by $\delta' = \{(*, (p, a, q), *, \bar{u}) \mid (p, a, q, \bar{u}) \in \delta\}$. Finally, for each $q_0 \in I$ we define an NMA $\mathcal{B}_{q_0} = (Q, \Gamma, q_0, \delta'', \mathcal{F})$ by $\delta'' = \{(p, (p, a, q), q) \mid (p, a, q) \in \Gamma\}$ and let $L = \bigcup_{q_0 \in I} \mathcal{L}(\mathcal{B}_{q_0})$. Then $S = h(L \cap [\![\mathcal{A}']\!])$.

Conversely, take Γ, h, $\mathcal{A}'$, and L as in the statement of the theorem. Then $h(L \cap [\![\mathcal{A}']\!])$ is Val-MC-recognizable by combining Lemma 4 and Lemma 6. $\square$

7 Conclusion

We introduced a new logic which is expressively equivalent to Quantitative Monitor Automata. Since our proofs are constructive, we immediately obtain the possibility to reduce the satisfiability and equivalence problems of our logic to the emptiness and equivalence problems of Quantitative Monitor Automata. Future work could therefore focus on the investigation of this automaton model, and on the related model of Nested Weighted Automata [3].

References

[1] J. R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik und Grundl. Math.*, 6:66–92, 1960.

[2] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In M. Kaminski and S. Martini, editors, *Proc. CSL*, volume 5213 of *LNCS*, pages 385–400. Springer, 2008.

[3] K. Chatterjee, T. A. Henzinger, and J. Otop. Nested weighted automata. In *Proc. LICS*, pages 725–737, 2015.

[4] K. Chatterjee, T. A. Henzinger, and J. Otop. Quantitative monitor automata. In X. Rival, editor, *Proc. SAS*, volume 9837 of *LNCS*. Springer, 2016.

[5] M. Droste and S. Dück. Weighted automata and logics on graphs. In G. F. Italiano, G. Pighizzini, and D. T. Sannella, editors, *Proc. MFCS*, volume 9234 of *LNCS*, pages 192–204. Springer, 2015.

[6] M. Droste and P. Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380:69 – 86, 2007.

[7] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. Monogr. Theoret. Comput. Sci. EATCS Ser. Springer, 2009.

[8] M. Droste and I. Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Inform. Comput.*, 220–221:44 – 59, 2012.

[9] M. Droste and G. Rahonis. Weighted automata and weighted logics on infinite words. In *Proc. DLT*, volume 4036 of *LNCS*, pages 49–58. Springer, 2006.

[10] M. Droste and H. Vogler. Weighted automata and multi-valued logics over arbitrary bounded lattices. *Theor. Comput. Sci.*, 418:14 – 36, 2012.

[11] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Am. Math. Soc.*, 98(1):21–51, 1961.

[12] Z. Ésik and W. Kuich. A semiring-semimodule generalization of ω-regular languages I. Special issue on "Weighted automata". In M. Droste and H. Vogler, editors, *J. Autom. Lang. Comb.*, volume 10, pages 203–242. 2005.

[13] Z. Ésik and W. Kuich. A semiring-semimodule generalization of ω-regular languages II. Special issue on "Weighted automata". In M. Droste and H. Vogler, editors, *J. Autom. Lang. Comb.*, volume 10, pages 243–264. 2005.

[14] I. Fichtner. Weighted picture automata and weighted logics. *Theor. Comput. Syst.*, 48(1):48–78, 2011.

[15] C. Mathissen. Weighted logics for nested words and algebraic formal power series. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Proc. ICALP*, volume 5126 of *LNCS*, pages 221–232. Springer, 2008.

[16] B. A. Trakhtenbrot. Finite automata and logic of monadic predicates. *Doklady Akademii Nauk SSSR*, 140:326–329, 1961. In Russian.