Expressiveness and Decidability of Weighted Automata and Weighted Logics

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
eingereichte

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

DOCTOR RERUM NATURALIUM
(Dr. rer. nat.)

im Fachgebiet
Informatik

vorgelegt

von Dipl.-Math. Erik Paul
geboren am 24. Juni 1989 in Schkeuditz

Leipzig, den 08. Januar 2020

Acknowledgments

I want to thank Manfred Droste for supervising this thesis. His guidance has been a tremendous help in improving my abilities to conduct research and he has provided me with many insights into the world of research as a whole. Also, I want to thank Heiko Vogler for his support as the second supervisor of this thesis.

I want to thank Stefan Dück, Andreas Maletti, and Tobias Weihrauch for the many helpful and enjoyable discussion we have had on both scientific and non-scientific topics. In particular, I want to thank Stefan for his great advice on all the small details of everyday research life, and I want to thank Andreas for his invaluable advice on my research on tree automata.

I also want to thank Peter Leupold for guiding me through the very first steps of academic writing, Vitaly Perevoshchikov for helping me get started on the first topic of my PhD studies, and Mirko Schulze for his great support in my teaching duties during the final preparation of this dissertation.

I am grateful to the Deutsche Forschungsgemeinschaft (DFG) for supporting me financially in the scope of its Graduiertenkolleg 1763 "Quantitative Logics and Automata", and of course also for providing me with such a wonderful research environment.

Finally, I want to thank my family who has supported me in many ways over the last years.

Contents

1

Introduction

Automata theory, one of the main branches of theoretical computer science, established its roots in the middle of the 20th century. One of its most fundamental concepts is that of a *finite automaton*, a basic yet powerful model of computation. In essence, finite automata provide a method to finitely represent possibly infinite sets of strings. Such a set of strings is also called a *language*, and the languages which can be described by finite automata are known as *regular* languages. Owing to their versatility, regular languages have received a great deal of attention over the years. Other formalisms were shown to be expressively equivalent to finite automata, most notably regular grammars [22], regular expressions [62], and monadic second order (MSO) logic [16, 38, 105]. To increase expressiveness, the fundamental idea underlying finite automata and regular languages was also extended to describe not only languages of strings, or *words*, but also of *infinite words* by Büchi [16] and Muller [78], *finite trees* by Doner [25] and Thatcher and Wright [104], *infinite trees* by Rabin [89], *nested words* by Alur and Madhusudan [3], and *pictures* [12] by Blum and Hewitt, just to name a few examples. In a parallel line of development, Schützenberger introduced *weighted automata* [98] which allow the description of quantitative properties of regular languages. In subsequent works, many of these descriptive formalisms and extensions were combined and their relationships investigated. This includes regular expressions, regular grammars, and logical characterizations for regular languages of trees and pictures as well as *weighted regular expressions* and *weighted logics*. For surveys on these and many more related topics, we refer to the books by Eilenberg [37], Salomaa and Soittola [97], Kuich and Salomaa [68], Berstel and Reutenauer [8], Rozenberg and Salomaa [92, 93, 94], and Droste, Kuich, and Vogler [29].

In this work, we focus on two of these extensions and their relationship, namely weighted automata and weighted logics. Just as the classical Büchi-Elgot-Trakhtenbrot Theorem [16, 38, 105] established the coincidence of regular languages with languages definable in monadic second order logic, weighted automata have been shown to be expressively equivalent to a specific fragment of a weighted monadic second order logic by Droste and Gastin [27]. We will explore several aspects of

weighted automata and of this weighted logic. More precisely, we will consider the following topics. In Chapter 3, we study a weighted monadic second order logic and prove that a classical model-theoretic result also holds for this logic. In Chapter 4, we study weighted tree automata over the max-plus semiring and prove four decidability results for these automata. In Chapter 5, we investigate the relationship between weighted automata and weighted logics and develop a weighted logic and an expressive equivalence result for *quantitative monitor automata*. In the following, we provide a more in-depth overview of the chapters of this thesis.

In Chapter 2, we introduce some fundamental concepts used throughout this thesis. To avoid an overloaded section of preliminaries, all concepts which are relevant only to a single chapter are introduced at the beginning of the respective chapter. For the most part, the individual chapters are designed to be readable in isolation. The only exception to this is Section 4.2 which uses some definitions of Chapter 3.

In Chapter 3, we extend the classical Feferman-Vaught Theorem [44] to the weighted setting. The Feferman-Vaught Theorem is one of the fundamental theorems in model theory. The theorem describes how the computation of the truth value of a first order sentence in a generalized product of relational structures can be reduced to the computation of truth values of first order sentences in the contributing structures and the evaluation of a monadic second order (MSO) sentence in the index structure. The theorem itself has a long-standing history. It builds upon work of Mostowski [77], and was shown in subsequent works to hold true for MSO logic [36, 51, 52, 69, 101].

Here, we show that under appropriate assumptions, the Feferman-Vaught Theorem also holds true for a weighted MSO logic with arbitrary commutative semirings as weight structure. For this, we consider the logic introduced by Droste and Gastin in [27] and allow the use of arbitrary relational signatures. In comparison to Boolean MSO logic, which is suitable only to specify qualitative properties of a structure, this logic permits the modeling of quantitative properties by employing binary sum and product operators as well as sum and product quantifiers.

For arbitrary commutative semirings, we identify a fragment of this weighted logic whose formulas satisfy a Feferman-Vaught-like theorem over disjoint unions of finite structures. This fragment is characterized by disallowing the second order product quantifier and restricting the first order quantifier to quantify only over formulas not containing any quantifier themselves. We show that for this fragment, the evaluation of a formula on the disjoint union of two finite structures can be reduced to evaluations of formulas from our fragment in the contributing structures and the combination of their results in an elementary way. Moreover, we show that if the product quantifier is removed entirely from the logic, the formulas from the resulting fragment satisfy an analogous statement for products of finite structures. Very importantly, we also show that if the restrictions on the product quantifiers are violated, then the Feferman-Vaught-like theorems do not hold. For this, we employ a weak version of Ramsey's Theorem [90]. We remark that surprisingly, the fragment for disjoint unions coincides with the fragment shown to be expressively equivalent to weighted word automata in [27].

In order to deal with infinite structures, we also consider *bicomplete* semirings which provide infinite sum and product operators. For bicomplete semirings, we show that statements analogous to those above for finite structures also hold for infinite structures. Moreover, for *weakly biaperiodic* semirings and *De Morgan algebras*, we show that Feferman-Vaught-like theorems hold for the full logic, without the need for any restrictions on the product quantifiers as described above. obtain more general versions of our theorems. Translation schemes allow structures over one signature to be "translated" into structures over another signature. In particular, they allow us to extend our theorems to more general combinations of structures than only disjoint unions and products by allowing MSO-defined modifications to these. An extended abstract of some of the results of this chapter appeared as [31].

In Chapter 4, we lift four decidability results from max-plus word automata to max-plus tree automata. Max-plus word and tree automata are weighted automata over the max-plus semiring and assign real numbers to words or trees, respectively. More precisely, a max-plus automaton is a finite automaton whose transitions are weighted by real numbers. To each run of a max-plus automaton, a weight is assigned by summing over the weights of the transitions which constitute the run. The automaton assigns a weight to each word or tree by taking the maximum of the weights of all runs on the given word or tree. We show that, like for max-plus word automata, the *equivalence*, *unambiguity*, and *sequentiality* problems are decidable for *finitely ambiguous* max-plus tree automata, and that the *finite sequentiality* problem is decidable for *unambiguous* max-plus tree automata.

The *equivalence problem* asks whether two max-plus automata are equivalent in the sense that to each input, they both assign the same weight. In his seminal paper, Krob showed that the equivalence problem of max-plus word automata is undecidable in general [66]. For *finitely ambiguous* max-plus automata, on the other hand, Hashiguchi et al. showed that this problem is decidable [55]. Here, a max-plus automaton is called *finitely ambiguous* if the number of accepting runs on each input is bounded by a global constant. In this chapter, we generalize this result to finitely ambiguous max-plus tree automata. We adopt some of the ideas from the proof of [55] but replace the *cycle decompositions* employed therein by an application of Parikh's theorem [81]. This idea was suggested by Mikołaj Bojańczyk after a presentation of our original proof of the statement which still employed cycle decompositions [84]. This approach greatly simplifies the proof, even in comparison to the one for words.

The *sequentiality problem* is a prominent open problem of max-plus automata and asks whether a given automaton is equivalent to a *deterministic* automaton. We call a max-plus word automaton deterministic or *sequential* if it possesses at most one initial state and for each pair of a state and an input letter, there exists at most one valid successor state. Although open in general, the sequentiality problem was shown to be decidable for *unambiguous* max-plus word automata by Mohri [75], finitely ambiguous max-plus word automata by Klimann et al. [63], and later even for *polynomially ambiguous* max-plus word automata by Kirsten and Lombardy [61]. A max-plus automaton is called *unambiguous* if there exists at most one accepting run on every input. It is called *polynomially ambiguous* if the number of accepting runs on each word is bounded polynomially in the size of the input, i.e., in the length

of the word or in the number of nodes of the tree.

The results by Klimann et al. and by Kirsten and Lombardy are in fact more general as they establish the decidability of the *unambiguity problem* for finitely ambiguous and polynomially ambiguous max-plus word automata, respectively. The unambiguity problem is a relaxation of the sequentiality problem and asks for a given automaton whether it is equivalent to an unambiguous automaton. From the respective results on the decidability of the unambiguity problem, the decidability of the sequentiality problem follows by Mohri's result on unambiguous max-plus word automata. In this chapter, we extend the result of Klimann et al. to trees. For this, we adapt the *dominance property*, the criterion for unambiguity identified in [63], to tree automata and show that the resulting criterion allows us to decide the unambiguity problem for finitely ambiguous max-plus tree automata. We then combine results of Büchse et al. [18] and Mohri [75] to also obtain the decidability of the sequentiality problem for these automata.

Finally, the *finite sequentiality problem*, also a relaxation of the sequentiality problem, asks whether for a given max-plus automaton, there exist finitely many deterministic max-plus automata whose pointwise maximum is equivalent to the given automaton. This problem was first posed by Hashiguchi et al. in [55] but was solved only recently by Bala and Koniński for both unambiguous and finitely ambiguous max-plus word automata [5, 4]. In this chapter, we show that finite sequentiality of unambiguous max-plus tree automata is decidable as well. For unambiguous max-plus word automata, the *fork property* was shown to be a criterion for deciding finite sequentiality in [5]. We extend the fork property by an additional criterion accounting for the nonlinear structure of trees and show that the resulting criterion is both necessary and sufficient for finite sequentiality of unambiguous max-plus tree automata. Extended abstracts of the results of this chapter, excluding the one from Section 4.2, appeared as [84] and [86].

In Chapter 5, we develop a logic which is expressively equivalent to *quantitative monitor automata*, a weighted automaton model operating on infinite words introduced very recently by Chatterjee, Henzinger, and Otop [20]. At each transition of a run on an infinite word, a quantitative monitor automaton can activate one of finitely many *monitor counters*. On subsequent transitions, an activated counter can either be modified by being incremented or decremented or the counter can be terminated. A counter may only be activated if it is not already active and counters must be terminated after finitely many steps. The counters only "monitor" the run and do not influence it. The value of a counter when terminated is associated to the position of the word where the counter was activated. The resulting infinite sequence of counter values is aggregated into a real number by a *valuation function* and serves as the weight of the run. The weight assigned to an infinite word is given by the infimum over the weights of all runs on the word. Quantitative monitor automata possess several interesting features. They are expressively equivalent to a subclass of *nested weighted automata* [21], an automaton model which for many valuation functions has decidable emptiness and universality problems. Also, quantitative monitor automata are more expressive than *weighted Büchi-automata* [40, 41] and their extension with valuation functions [30].

In this chapter, we introduce a new logic which we call *monitor logic* and show that it is expressively equivalent to quantitative monitor automata. The logic is equipped with a sum quantifier to handle the counter operations, a valuation quantifier for the aggregation of the counter values, and an infimum quantifier to compute the infimum over all runs on an infinite word. Our expressive equivalence result is effective, i.e., for each formula from our logic, we show how to construct a quantitative monitor automaton whose behavior coincides with the semantics of the formula, and for every automaton, we show how a formula describing precisely this automaton can be constructed. The most challenging aspect of establishing this logical characterization of quantitative monitor automata was to find appropriate quantifiers and, in turn, appropriate restrictions on these quantifiers. In particular, we show that the computations of the sum quantifier need to further depend on an MSO-definable condition and that if this restriction is dropped, the logic becomes strictly more powerful than quantitative monitor automata. In order to obtain our logical characterization, we also prove various closure properties of quantitative monitor automata and the equivalent expressive power of Büchi and Muller acceptance conditions for these automata. An extended abstract of the results of this chapter was published as [85].

2

Preliminaries

> The best books, he perceived,
> are those that tell you what you know already.
>
> GEORGE ORWELL, *1984*

In the following, we introduce basic notions and concepts used throughout this work.

2.1 Basic Definitions

We let $\mathbb{N} = \{0, 1, 2, \ldots\}$ denote the natural numbers, $\mathbb{N}_+ = \{1, 2, 3, \ldots\}$ the natural numbers excluding zero, $\mathbb{Z}$ the integers, $\mathbb{Q}$ the rational numbers, and $\mathbb{R}$ the real numbers.

For a set X, we denote the power set of X by $\mathcal{P}(X)$ and the cardinality of X by $|X|$. For a set Y, we denote the set of all mappings from X to Y by Y^X. For an integer $n \geq 1$ and a tuple $\bar{x} \in X^n$, we will usually assume that its elements are called $x_1, \ldots, x_n$, i.e., that $\bar{x} = (x_1, \ldots, x_n)$. For a mapping $f\colon X \to Y$, we call X the *domain* of f, denoted by $\mathrm{dom}(f)$, and Y the *range* of f, denoted by $\mathrm{range}(f)$. For a subset $X' \subseteq X$, we call the set $f(X') = \{y \in Y \mid \exists x \in X'\colon f(x) = y\}$ the *image* or *range of X' under f*. 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 $x \in X'$. We also call a mapping $g\colon X' \to Y$ a *partial* mapping from X to Y, denoted by $g\colon X \nrightarrow Y$. For a subset $Y' \subseteq Y$, we call the set $f^{-1}(Y') = \{x \in X \mid f(x) \in Y'\}$ the *preimage of Y' under f*. For an element $y \in Y$, we define the preimage of y under f by $f^{-1}(y) = f^{-1}(\{y\})$. For a second mapping $h\colon X \to Y$, we write $f = h$ if for all $x \in X$ we have $f(x) = h(x)$. For a set Z and a mapping $h\colon Y \to Z$, the *composition* of f and h is the mapping $h \circ f\colon X \to Z$ defined by $(h \circ f)(x) = h(f(x))$ for every $x \in X$.

An *alphabet* is a non-empty finite set. An *infinite alphabet* is a non-empty set. Let Σ be an infinite alphabet. By Σ^*, we denote the set of all (finite) words over Σ. The empty word is denoted by ε and the length of a word $w \in \Sigma^*$ by $|w|$. The number of occurrences of a letter $a \in \Sigma$ in w is denoted by $|w|_a$. For two words $u, v \in \Sigma^*$, the

concatenation of u and v is denoted by uv. A subset $L \subseteq \Sigma^*$ is called a *language* over Σ. 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. A subset $L \subseteq \Sigma^\omega$ is called an *infinitary language* over Σ.

2.2 Semirings

A *commutative semiring* is a tuple $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$, abbreviated by K, with operations sum $\oplus$ and product $\odot$ and constants $\mathbb{0}$ and $\mathbb{1}$ such that $(K, \oplus, \mathbb{0})$ and $(K, \odot, \mathbb{1})$ are commutative monoids, multiplication distributes over addition, and $\kappa \odot \mathbb{0} = \mathbb{0} \odot \kappa = \mathbb{0}$ for every $\kappa \in K$. Important examples of commutative semirings include

- the *Boolean semiring* $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ with disjunction $\vee$ and conjunction $\wedge$,
- the semiring of natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$ with the usual addition and multiplication,
- the fields of rational numbers $(\mathbb{Q}, +, \cdot, 0, 1)$ and real numbers $(\mathbb{R}, +, \cdot, 0, 1)$,
- the *max-plus* or *arctic semiring* $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ where the sum and the product operations are max and $+$, respectively, extended to $\mathbb{R} \cup \{-\infty\}$ in the usual way,
- the *min-plus* or *tropical semiring* $\mathbb{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ where the sum and the product operations are min and $+$, respectively,
- the *min-max semiring* $(\mathbb{R} \cup \{\infty, -\infty\}, \min, \max, \infty, -\infty)$,
- the *Viterbi semiring* $([0,1], \max, \cdot, 0, 1)$, where the product operation is the usual multiplication of real numbers,
- the *semiring of formal languages* $(\mathcal{P}(\Sigma^*), \cup, \cdot, \emptyset, \{\varepsilon\})$ over an alphabet Σ, where the sum and product operations are the union and concatenation of languages, respectively. Here, the concatenation of two languages $L_1, L_2 \subseteq \Sigma^*$ is the language $L_1 \cdot L_2 = \{uv \mid u \in L_1, v \in L_2\}$.

For a commutative semiring $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ and a number $n \geq 1$, the *product semiring* $(K^n, \oplus_n, \odot_n, \mathbb{0}_n, \mathbb{1}_n)$ is defined by componentwise operations and the constants $\mathbb{0}_n = (\mathbb{0}, \ldots, \mathbb{0})$ and $\mathbb{1}_n = (\mathbb{1}, \ldots, \mathbb{1})$. We will usually denote $\oplus_n$ and $\odot_n$ simply by $\oplus$ and $\odot$.

Next, assume that the commutative semiring K is equipped, for every index set I, with an infinitary sum operation $\bigoplus_I \colon K^I \to K$ such that for every family $(\kappa_i)_{i \in I}$

of elements of K and $\kappa \in K$ we have

$$\bigoplus_{i\in\emptyset} \kappa_i = 0, \quad \bigoplus_{i\in\{j\}} \kappa_i = \kappa_j, \quad \bigoplus_{i\in\{j,l\}} \kappa_i = \kappa_j \oplus \kappa_l \text{ for } j \neq l, \tag{2.2.1}$$

$$\bigoplus_{j\in J} \Big(\bigoplus_{i\in I_j} \kappa_i \Big) = \bigoplus_{i\in I} \kappa_i, \text{ if } \bigcup_{j\in J} I_j = I \text{ and } I_j \cap I_{j'} = \emptyset \text{ for } j \neq j', \tag{2.2.2}$$

$$\bigoplus_{i\in I} (\kappa \odot \kappa_i) = \kappa \odot \Big(\bigoplus_{i\in I} \kappa_i \Big), \quad \bigoplus_{i\in I} (\kappa_i \odot \kappa) = \Big(\bigoplus_{i\in I} \kappa_i \Big) \odot \kappa. \tag{2.2.3}$$

Then K together with the operations $\bigoplus_I$ is called *complete* [37, 67].

If in addition, K is endowed, for every index set I, with an infinitary product operation $\bigodot_I \colon K^I \to K$ such that for every family $(\kappa_i)_{i\in I}$ of elements of K we have

$$\bigodot_{i\in\emptyset} \kappa_i = \mathbb{1}, \quad \bigodot_{i\in\{j\}} \kappa_i = \kappa_j, \quad \bigodot_{i\in\{j,l\}} \kappa_i = \kappa_j \odot \kappa_l \text{ for } j \neq l, \quad \bigodot_{i\in I} \mathbb{1} = \mathbb{1}, \tag{2.2.4}$$

$$\bigodot_{j\in J} \Big(\bigodot_{i\in I_j} \kappa_i \Big) = \bigodot_{i\in I} \kappa_i, \text{ if } \bigcup_{j\in J} I_j = I \text{ and } I_j \cap I_{j'} = \emptyset \text{ for } j \neq j', \tag{2.2.5}$$

then we call K *bicomplete*. We just want to mention here that there exists a different notion of semirings with infinite sums and products, namely the notion of *totally complete* semirings [42]. The main difference between these two notions lies in the definition of infinite products. For totally complete semirings, only products over countable index sets need to be defined, but the infinite products are required to be completely distributive over the infinite sums. We do not require this infinitary distributivity here. Examples of bicomplete semirings include

- the min-max semiring,
- the min-plus semiring $(\mathbb{R}_{\geq 0} \cup \{\infty\}, \min, +, \infty, 0)$ restricted to the positive reals, i.e., where $\mathbb{R}_{\geq 0} = \{r \in \mathbb{R} \mid r \geq 0\}$,
- the semiring of extended natural numbers $(\mathbb{N} \cup \{\infty\}, +, \cdot, 0, 1)$ where $0 \cdot \infty = 0$,
- the Boolean semiring, or more generally every complete distributive lattice $(L, \vee, \wedge, 0, 1)$ which satisfies the distributivity law (2.2.3). For instance, every complete Boolean algebra B satisfies (2.2.3) (see [9, page 167]), so then B is bicomplete but may not be completely distributive and therefore not totally complete.

2.3 First Order and Monadic Second Order Logic

A *signature* σ is a pair $(\mathrm{Rel}_\sigma, \mathrm{ar}_\sigma)$ where Rel_σ is a set of relation symbols and $\mathrm{ar}_\sigma \colon \mathrm{Rel}_\sigma \to \mathbb{N}_+$ the arity function. A *σ-structure* $\mathfrak{A}$ is a pair $(\mathcal{U}_\mathfrak{A}, \mathcal{I}_\mathfrak{A})$ where $\mathcal{U}_\mathfrak{A}$ is a set, called the *universe of* $\mathfrak{A}$, and $\mathcal{I}_\mathfrak{A}$ is an *interpretation*, which maps every symbol $R \in \mathrm{Rel}_\sigma$ to a set $R^\mathfrak{A} \subseteq \mathcal{U}_\mathfrak{A}^{\mathrm{ar}_\sigma(R)}$. A structure is called *finite* if its universe is a finite set. By $\mathrm{Str}(\sigma)$ we denote the class of all σ-structures.

Example 2.1. For an alphabet Σ, we can interpret every finite or infinite word w over Σ as a structure $\mathfrak{w} = (W, \mathcal{I}_\mathfrak{w})$ over the signature $\sigma = (\{P_a \mid a \in \Sigma\} \cup \{\leq\}, \mathrm{ar}_\sigma)$ where $\mathrm{ar}_\sigma(\leq) = 2$ and $\mathrm{ar}_\sigma(P_a) = 1$ for every $a \in \Sigma$. For this, let $w = a_0 a_1 \ldots \in \Sigma^* \cup \Sigma^\omega$. If w is finite, we let $W = \{0, \ldots, |w| - 1\}$, and otherwise, we let $W = \mathbb{N}$. Furthermore, we let $\mathcal{I}_\mathfrak{w}(\leq) = (W \times W) \cap \leq$, where the latter $\leq$ denotes the usual less-than-or-equal relation on the natural numbers, and for each letter $a \in \Sigma$ define the interpretation of P_a by $\mathcal{I}_\mathfrak{w}(P_a) = \{i \in W \mid a_i = a\}$.

For two σ-structures $\mathfrak{A} = (A, \mathcal{I}_\mathfrak{A})$ and $\mathfrak{B} = (B, \mathcal{I}_\mathfrak{B})$, we define the *product* $\mathfrak{A} \times \mathfrak{B} \in \mathrm{Str}(\sigma)$ *of* $\mathfrak{A}$ *and* $\mathfrak{B}$ and the *disjoint union* $\mathfrak{A} \sqcup \mathfrak{B} \in \mathrm{Str}(\sigma)$ *of* $\mathfrak{A}$ *and* $\mathfrak{B}$ as follows. For the product, we let $\mathfrak{A} \times \mathfrak{B} = (A \times B, \mathcal{I}_{\mathfrak{A}\times\mathfrak{B}})$ with $R^{\mathfrak{A}\times\mathfrak{B}} = \{((a_1, b_1), \ldots, (a_n, b_n)) \mid (a_1, \ldots, a_n) \in R^\mathfrak{A} \text{ and } (b_1, \ldots, b_n) \in R^\mathfrak{B}\}$ for each $R \in \mathrm{Rel}_\sigma$ and $n = \mathrm{ar}_\sigma(R)$. For the disjoint union, we let $A \sqcup B$ be the disjoint union (i.e., the set theoretic coproduct) of A and B with inclusions ι_A and ι_B. Then we define $\mathfrak{A} \sqcup \mathfrak{B} = (A \sqcup B, \mathcal{I}_{\mathfrak{A}\sqcup\mathfrak{B}})$ by $R^{\mathfrak{A}\sqcup\mathfrak{B}} = \{(\iota_A(a_1), \ldots, \iota_A(a_k)) \mid (a_1, \ldots, a_k) \in R^\mathfrak{A}\} \cup \{(\iota_B(b_1), \ldots, \iota_A(b_k)) \mid (b_1, \ldots, b_k) \in R^\mathfrak{B}\}$ for each $R \in \mathrm{Rel}_\sigma$ and $n = \mathrm{ar}_\sigma(R)$. Throughout this chapter, we identify $a \in A$ with $\iota_A(a) \in A \sqcup B$ and $b \in B$ with $\iota_B(b) \in A \sqcup B$.

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 first order formulas β over a signature σ by the grammar

$$\beta ::= \texttt{false} \mid R(x_1, \ldots, x_n) \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\beta,$$

where $R \in \mathrm{Rel}_\sigma$, $n = \mathrm{ar}_\sigma(R)$, and $x, x_1, \ldots, x_n \in \mathcal{V}$ are first order variables. Likewise, we define monadic second order formulas β over σ through

$$\beta ::= \texttt{false} \mid R(x_1, \ldots, x_n) \mid x \in X \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\beta \mid \exists X.\beta,$$

where $R \in \mathrm{Rel}_\sigma$, $n = \mathrm{ar}_\sigma(R)$, $x, x_1, \ldots, x_n \in \mathcal{V}$ are first order variables, and $X \in \mathcal{V}$ is a second order variable. We also allow the usual abbreviations $\wedge$, $\forall$, $\rightarrow$, $\longleftrightarrow$, and $\texttt{true}$. By $\mathrm{FO}(\sigma)$, we denote the set of all first order formulas over σ, and by $\mathrm{MSO}(\sigma)$, we denote the set of all monadic second order formulas over σ.

The notion of *free variables* is defined as usual, i.e., the operators $\exists$ and $\forall$ bind variables. We let $\mathrm{Free}(\beta)$ be the set of all free variables of β. A formula β with $\mathrm{Free}(\beta) = \emptyset$ is called a *sentence*. For a tuple $\bar{\beta} = (\beta_1, \ldots, \beta_n) \in \mathrm{MSO}(\sigma)^n$, we define the set of free variables of $\bar{\beta}$ as $\mathrm{Free}(\bar{\beta}) = \bigcup_{i=1}^n \mathrm{Free}(\beta_i)$.

We now define the semantics of MSO. Let σ be a signature, $\mathfrak{A} = (A, \mathcal{I}_\mathfrak{A})$ a σ-structure, and $\mathcal{V}$ a set of first and second order variables. A $(\mathcal{V}, \mathfrak{A})$-assignment ρ is a partial function $\rho\colon \mathcal{V} \rightharpoonup A \cup \mathcal{P}(A)$ such that, whenever $x \in \mathcal{V}$ is a first order variable and $\rho(x)$ is defined, we have $\rho(x) \in A$, and whenever $X \in \mathcal{V}$ is a second order variable, $\rho(X)$ is defined and we have $\rho(X) \subseteq A$. By $\mathfrak{A}_\mathcal{V}$, we denote the set of all $(\mathcal{V}, \mathfrak{A})$-assignments. The reason we consider partial functions is that in Chapter 3, we want to be able to restrict the range of a variable assignment to a subset of the universe. For a first order variable, this restriction may cause the variable to become undefined.

For a $(\mathcal{V}, \mathfrak{A})$-assignment ρ, a first order variable $x \in \mathcal{V}$, and an element $a \in A$, the *update* $\rho[x \to a]$ is defined through

$$\rho[x \to a] \colon \mathrm{dom}(\rho) \cup \{x\} \to A \cup \mathcal{P}(A), \qquad \mathcal{X} \mapsto \begin{cases} a & \text{if } \mathcal{X} = x \\ \rho(\mathcal{X}) & \text{otherwise.} \end{cases}$$

For a second order variable $X \in \mathcal{V}$ and a set $I \subseteq A$, the update $\rho[X \to I]$ is defined through

$$\rho[X \to I] \colon \mathrm{dom}(\rho) \cup \{X\} \to A \cup \mathcal{P}(A), \qquad \mathcal{X} \mapsto \begin{cases} I & \text{if } \mathcal{X} = X \\ \rho(\mathcal{X}) & \text{otherwise.} \end{cases}$$

For a tuple $\bar{\mathcal{X}} = (\mathcal{X}_1, \ldots, \mathcal{X}_n) \in \mathcal{V}^n$ of pairwise distinct variables and (matching) values $\bar{a} = (a_1, \ldots, a_n) \in (A \cup \mathcal{P}(A))^n$, we abbreviate the repeated update $\rho[\mathcal{X}_1 \to a_1] \cdots [\mathcal{X}_n \to a_n]$ by $\rho[\mathcal{X}_1 \to a_1, \ldots, \mathcal{X}_n \to a_n]$ or simply $\rho[\bar{\mathcal{X}} \to \bar{a}]$.

For $\rho \in \mathfrak{A}_{\mathcal{V}}$ and a formula $\beta \in \mathrm{MSO}(\sigma)$, the relation "$(\mathfrak{A}, \rho)$ satisfies β", denoted by $(\mathfrak{A}, \rho) \models \beta$, is defined as

$$\begin{array}{lcl}
(\mathfrak{A}, \rho) \models \texttt{false} & & \text{never holds} \\
(\mathfrak{A}, \rho) \models R(x_1, \ldots, x_n) & \iff & x_1, \ldots, x_n \in \mathrm{dom}(\rho) \text{ and } (\rho(x_1), \ldots, \rho(x_n)) \in R^{\mathfrak{A}} \\
(\mathfrak{A}, \rho) \models x \in X & \iff & x \in \mathrm{dom}(\rho) \text{ and } \rho(x) \in \rho(X) \\
(\mathfrak{A}, \rho) \models \neg\beta & \iff & (\mathfrak{A}, \rho) \models \beta \text{ does not hold} \\
(\mathfrak{A}, \rho) \models \beta_1 \vee \beta_2 & \iff & (\mathfrak{A}, \rho) \models \beta_1 \text{ or } (\mathfrak{A}, \rho) \models \beta_2 \\
(\mathfrak{A}, \rho) \models \exists x.\beta & \iff & (\mathfrak{A}, \rho[x \to a]) \models \beta \text{ for some } a \in A \\
(\mathfrak{A}, \rho) \models \exists X.\beta & \iff & (\mathfrak{A}, \rho[X \to I]) \models \beta \text{ for some } I \subseteq A.
\end{array}$$

We will usually identify a pair $(\mathfrak{A}, \emptyset)$ with $\mathfrak{A}$.

Example 2.2. Let σ be the signature of a graph, i.e., $\mathrm{Rel}_\sigma = \{\mathrm{edge}\}$ with edge binary. We call a graph $\mathfrak{G} \in \mathrm{Str}(\sigma)$ *undirected* if its interpretation of edge is a symmetric relation on the universe of $\mathfrak{G}$. For every undirected graph $\mathfrak{G} \in \mathrm{Str}(\sigma)$ and a subset I of its universe, we can check whether the nodes from I form a clique in $\mathfrak{G}$ using the MSO formula

$$\mathrm{clique}(X) = \forall x \forall y \Big(\big(x \in X \wedge y \in X \wedge x \neq y \big) \to \mathrm{edge}(x, y) \Big).$$

Here, the formula $x \neq y$ is an abbreviation for $\exists Y (y \in Y \wedge \neg(x \in Y))$. We have that $(\mathfrak{G}, [X \to I])$ satisfies $\mathrm{clique}(X)$ if and only if I is a clique in $\mathfrak{G}$.

3

Feferman-Vaught Theorems

If his forces are united, separate them.

Sun Tzu, *The Art of War*

In this chapter, we extend the Feferman-Vaught Theorem [44] for Boolean logic to a weighted logic. The classical Feferman-Vaught Theorem shows how the evaluation of a sentence in first order or monadic second order logic on a generalized product of relational structures can be reduced to the evaluation of sentences on the contributing structures. For a survey and more background information on the Feferman-Vaught Theorem, see [70]. The logic we employ is based on the weighted logic by Droste and Gastin [27]. In this logic, formulas can take values which convey a quantitative meaning. The logic's connectives and quantifiers hence also adopt quantitative roles. The disjunction becomes a sum, the conjunction a product. The existential quantifier,

instead of only verifying whether some element with a certain property exists, now takes the truth value of this property for every element in the universe and sums over these values. Under appropriate assumptions, the result of this summation can for instance be the exact number of elements that satisfy the given property. One example of a property which can be expressed using this logic is the size of the largest clique of an undirected graph. In [27], the authors prove a Büchi-like result for a specific fragment of this logic, showing that for finite and infinite words, this fragment is expressively equivalent to semiring-weighted automata [98]. The study of a weighted Feferman-Vaught Theorem for disjoint unions, employing the same logic as we do, was initiated by Ravve et al. in [91], where the authors also point out several algorithmic uses and possible applications of a weighted Feferman-Vaught Theorem.

The classical Feferman-Vaught Theorem considers finite and infinite structures without any need for distinction between them. This results from the fact that, in the Boolean setting, infinite joins and meets are well-defined. In particular, existential and universal quantification, which are essentially joins and meets ranging over the whole universe of a structure, are well-defined for finite and infinite structures alike. However, for arbitrary semirings, infinite sums and products are usually not defined. To allow for infinite structures, we therefore also consider *bicomplete* semirings, which are equipped with infinite sum and product operations. Our main results are the following.

- We provide a Feferman-Vaught Theorem for disjoint unions of structures with our weighted MSO logic, where the second order product quantifier is removed and the first order product quantifier is restricted to quantify only over formulas which do not contain any sum or product quantifier themselves. Surprisingly, this restriction and the resulting fragment are the same as the one working for the Büchi-like result of [27].

- We show that no similar theorem can hold for disjoint unions if the first order product quantifier is not restricted. The formulas we employ for this in fact also occurred in [27] and [33] as examples of weighted formulas whose semantics cannot be described by weighted automata. While in these papers, it was elementary to show that the formulas given define weighted languages not recognizable by weighted automata, here proving that they do not allow for a Feferman-Vaught-like decomposition is more complex and employs a weak version of Ramsey's theorem [90].

- We show that a Feferman-Vaught Theorem also holds for products of structures for the product-quantifier-free first order fragment of our logic.

- We show that no similar theorem can hold for products if we include the first order product quantifier.

- We show that the restrictions on the product quantifiers are not necessary if the semiring is *weakly biaperiodic* or forms a *De Morgan algebra*.

- We show that our theorems are also true for more general disjoint unions and products defined by *translation schemes* and *transductions* [70, 79, 23].

With respect to our proofs, here we just note that in comparison to the universal quantifier of the Boolean setting, the product quantifier requires a separate and new consideration. While universal quantification can simply be expressed using negation and existential quantification, it is in general not possible to express multiplication by addition.

Translation schemes are a model theoretic tool to "translate" structures over one logical signature into structures over another signature in a well behaved fashion, namely in an MSO-defined fashion. They can be applied, for example, to translate between *texts* and *trees* [56], and between *nested words*, *alternating texts*, and *hedges* [73, 72, 71]. These particular translations were employed in [71] to prove that weighted automata over texts, hedges, and nested words are expressively equivalent to weighted logics over these structures. Translation schemes are a rather natural concept and therefore they have been frequently rediscovered and named differently [70, 79, 23]. Our notion of a translation scheme is mostly due to [70].

Related work. A concept related to weighted logics is that of many-valued logics. In both models the evaluation of a formula on a structure produces a quantitative piece of information. In many approaches to many-valued logics, values are taken in the interval $[0, 1]$, cf. [53, 50]. In contrast to this, weights in weighted logics are taken from a semiring and may occur as atomic formulas which enables the modeling of quantitative properties.

An extended abstract of the results of this chapter appeared at the 43rd International Symposium on Mathematical Foundations of Computer Science (MFCS) in 2018 [31].

3.1 Weighted First Order and Monadic Second Order Logic

The following definitions are due to [27] in the form of [14]. Let σ be a signature, $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ a commutative semiring, and $\mathcal{V}$ a countable set of first and second order variables. We define weighted first order formulas φ over σ and K by the grammar

$$\varphi ::= \beta \mid \kappa \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi,$$

where $\beta \in \mathrm{FO}(\sigma)$, $\kappa \in K$, and $x \in \mathcal{V}$ is a first order variable. Likewise, we define weighted monadic second order formulas φ over σ and K through

$$\varphi ::= \beta \mid \kappa \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes X.\varphi,$$

where $\beta \in \mathrm{MSO}(\sigma)$, $\kappa \in K$, $x \in \mathcal{V}$ is a first order variable, and $X \in \mathcal{V}$ is a second order variable. By $\mathrm{wFO}(\sigma, K)$, we denote the set of all weighted first order formulas over σ and K, and by $\mathrm{wMSO}(\sigma, K)$, we denote the set of all weighted monadic second order formulas over σ and K. The notions of free variables and sentences are defined

like for MSO formulas, with the addition that the operators $\bigoplus$ and $\bigotimes$ also bind variables.

We define the semantics of wMSO as follows. Let $\mathfrak{A} = (A, \mathcal{I}_\mathfrak{A})$ be a σ-structure. In the following, for all sums and products to be well-defined, we assume that either the universe A is finite, or that K is bicomplete. For a formula $\varphi \in \mathrm{wMSO}(\sigma, K)$, the *(weighted) semantics* of φ is a mapping $[\![\varphi]\!](\mathfrak{A}, \cdot)\colon \mathfrak{A}_\mathcal{V} \to K$ inductively defined as

$$\begin{aligned}
[\![\beta]\!](\mathfrak{A}, \rho) &= \begin{cases} \mathbb{1} & \text{if } (\mathfrak{A}, \rho) \models \beta \\ \mathbb{0} & \text{otherwise} \end{cases} \\
[\![\kappa]\!](\mathfrak{A}, \rho) &= \kappa \\
[\![\varphi_1 \oplus \varphi_2]\!](\mathfrak{A}, \rho) &= [\![\varphi_1]\!](\mathfrak{A}, \rho) \oplus [\![\varphi_2]\!](\mathfrak{A}, \rho) \\
[\![\varphi_1 \otimes \varphi_2]\!](\mathfrak{A}, \rho) &= [\![\varphi_1]\!](\mathfrak{A}, \rho) \odot [\![\varphi_2]\!](\mathfrak{A}, \rho) \\
[\![\textstyle\bigoplus x.\varphi]\!](\mathfrak{A}, \rho) &= \bigoplus_{a \in A} [\![\varphi]\!](\mathfrak{A}, \rho[x \to a]) \\
[\![\textstyle\bigotimes x.\varphi]\!](\mathfrak{A}, \rho) &= \bigodot_{a \in A} [\![\varphi]\!](\mathfrak{A}, \rho[x \to a]) \\
[\![\textstyle\bigoplus X.\varphi]\!](\mathfrak{A}, \rho) &= \bigoplus_{I \subseteq A} [\![\varphi]\!](\mathfrak{A}, \rho[X \to I]) \\
[\![\textstyle\bigotimes X.\varphi]\!](\mathfrak{A}, \rho) &= \bigodot_{I \subseteq A} [\![\varphi]\!](\mathfrak{A}, \rho[X \to I]).
\end{aligned}$$

For a tuple of formulas $\bar{\varphi} = (\varphi_1, \ldots, \varphi_n) \in \mathrm{wMSO}(\sigma, K)^n$, we define $[\![\bar{\varphi}]\!](\mathfrak{A}, \rho) = ([\![\varphi_1]\!](\mathfrak{A}, \rho), \ldots, [\![\varphi_n]\!](\mathfrak{A}, \rho)) \in K^n$.

We give some examples of how weighted formulas can be interpreted. For more examples, see also [91].

Example 3.1. If $K = \mathbb{B}$ is the Boolean semiring, we obtain the classical Boolean logic.

Example 3.2. Using the max-plus semiring $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$, we can describe the size of the largest clique in a graph as follows. We reuse the signature σ of a graph and the MSO formula $\mathrm{clique}(X)$ from Example 2.2 and define a wMSO formula as follows.

$$\varphi = \bigoplus X.\Big(\mathrm{clique}(X) \otimes \bigotimes x.\big(0 \oplus (1 \otimes x \in X)\big)\Big)$$

Then for every undirected graph $\mathfrak{G} \in \mathrm{Str}(\sigma)$, we have that $[\![\varphi]\!](\mathfrak{G})$ is the size of the largest clique in $\mathfrak{G}$.

Example 3.3. Assume that $K = (\mathbb{Q}, +, \cdot, 0, 1)$ is the field of rational numbers and that σ is the signature from the previous example. Then for every fixed $n \in \mathbb{N}_+$, we can count the number of n-cliques of an undirected graph $\mathfrak{G} \in \mathrm{Str}(\sigma)$ using the wMSO formula

$$\varphi_n = \frac{1}{n!} \otimes \bigoplus x_1 \ldots \bigoplus x_n. \bigwedge_{i \neq j} \Big((x_i \neq x_j) \wedge \mathrm{edge}(x_i, x_j)\Big).$$

Here, $x_i \neq x_j$ again is an abbreviation for $\exists Y(x_j \in Y \wedge \neg(x_i \in Y))$.

Example 3.4. We consider the *minimum cut* of directed acyclic graphs. For this, we interpret these graphs as *flow networks* in the following way. Every vertex which does not have a predecessor is considered a *source*, every vertex without successors is considered a *drain*, and every edge is assumed to have a capacity of 1. Let $G = (V, E)$ be a directed acyclic graph where V is the set of vertices and $E \subseteq V \times V$ the set of edges. A cut (S, D) of G is a partition of V, i.e., $S \cup D = V$ and $S \cap D = \emptyset$, such that all sources of G are in S, and all drains of G are in D. The *minimum cut* of G is the smallest number $|E \cap (S \times D)|$ such that (S, D) is a cut of G.

We can express the minimum cut of directed acyclic graphs by a weighted formula as follows. We let σ be the signature from the previous two examples and as our semiring, we choose the min-plus semiring $\mathbb{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$. Then using the abbreviation

$$\begin{aligned}\mathrm{cut}(X, Y) = \forall x.\Big((x \in X \leftrightarrow \neg(x \in Y)) \wedge (\exists y.\mathrm{edge}(y, x) \vee x \in X) \\ \wedge (\exists y.\mathrm{edge}(x, y) \vee x \in Y)\Big)\end{aligned}$$

we can express the minimum cut of a directed acyclic graph $\mathfrak{G} \in \mathrm{Str}(\sigma)$ using the formula

$$\varphi = \bigoplus X.\bigoplus Y.\Big(\mathrm{cut}(X, Y) \otimes \bigotimes x.\bigotimes y.(1 \oplus \neg(x \in X \wedge y \in Y \wedge \mathrm{edge}(x, y)))\Big).$$

Example 3.5 ([27])**.** Let $K = (\mathbb{N}, +, \cdot, 0, 1)$ be the semiring of natural numbers and let $\varphi \in \mathrm{wMSO}(\sigma, K)$ be a formula which does not contain any constants $\kappa \in K$. Then we may understand $[\![\varphi]\!](\mathfrak{A}, \rho)$ as the number of proofs we have that $(\mathfrak{A}, \rho)$ satisfies φ assuming that we interpret the weighted operators in the following way. For Boolean formulas, we simply consider satisfaction to give us one proof, and otherwise we have no proof. The sum $[\![\varphi_1 \oplus \varphi_2]\!]$ is the number of proofs we have that $\varphi_1 \vee \varphi_2$ is true. This says that, if we have n proofs for φ_1 and m proofs for φ_2, then we interpret this as having $n + m$ proofs for the fact that $\varphi_1 \vee \varphi_2$ is true. Likewise, we interpret the product $[\![\varphi_1 \otimes \varphi_2]\!]$ as the number of proofs we have that $\varphi_1 \wedge \varphi_2$ is true. Similar interpretations apply for the weighted quantifiers.

For $\varphi \in \mathrm{wMSO}(\sigma, K)$ and a first order variable x which does not appear in φ as a bound variable, we define φ^{-x} as the formula obtained from φ by replacing all atomic subformulas containing x, i.e., all subformulas of the form $x \in X$ and $R(\ldots, x, \ldots)$ for $R \in \mathrm{Rel}_\sigma$, by `false`. It is easy to show by induction that for all σ-structures $\mathfrak{A} = (A, \mathcal{I}_\mathfrak{A})$ and $(\mathcal{V}, \mathfrak{A})$-assignments ρ with $x \notin \mathrm{dom}(\rho)$ we have

$$[\![\varphi]\!](\mathfrak{A}, \rho) = [\![\varphi^{-x}]\!](\mathfrak{A}, \rho).$$

As in the sequel we will deal with disjoint unions and products of structures, we need to define the restrictions of a variable assignment to the contributing structures of the disjoint union or product. Let $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ be two structures with universes A and B. For a $(\mathcal{V}, \mathfrak{A} \sqcup \mathfrak{B})$-assignment ρ, we define the restriction $\rho{\restriction_\mathfrak{A}} \colon \mathcal{V} \nrightarrow A$ as

$$\rho{\restriction_\mathfrak{A}}(\mathcal{X}) = \begin{cases} \rho(\mathcal{X}) \cap A & \text{if } \mathcal{X} \text{ is a second order variable} \\ \rho(\mathcal{X}) & \text{if } \mathcal{X} \text{ is a first order variable and } \rho(\mathcal{X}) \in A \\ \text{undefined} & \text{if } \mathcal{X} \text{ is a first order variable and } \rho(\mathcal{X}) \notin A. \end{cases}$$

The restriction $\rho{\restriction_{\mathfrak{B}}}$ is defined similarly.

For a $(\mathcal{V}, \mathfrak{A}\times\mathfrak{B})$-assignment ρ, we define the restrictions $\rho{\restriction_{\mathfrak{A}}}$ and $\rho{\restriction_{\mathfrak{B}}}$ by projection on the corresponding entries. That is, we let π_A be the projection on the first and π_B be the projection on the second entry of $A \times B$ and let $\rho{\restriction_{\mathfrak{A}}} = \pi_A \circ \rho$ and $\rho{\restriction_{\mathfrak{B}}} = \pi_B \circ \rho$.

The *union* of two assignments ρ and ς with $\mathrm{dom}(\rho) \cap \mathrm{dom}(\varsigma) = \emptyset$, denoted by $\rho \cup \varsigma$, is defined by $\mathrm{dom}(\rho \cup \varsigma) = \mathrm{dom}(\rho) \cup \mathrm{dom}(\varsigma)$, $(\rho \cup \varsigma)(\mathcal{X}) = \rho(\mathcal{X})$ for $\mathcal{X} \in \mathrm{dom}(\rho)$ and $(\rho \cup \varsigma)(\mathcal{X}) = \varsigma(\mathcal{X})$ for $\mathcal{X} \in \mathrm{dom}(\varsigma)$.

We fix two disjoint sets of variables $(x_i)_{i\in\mathbb{N}}$ and $(y_i)_{i\in\mathbb{N}}$. For $n \in \mathbb{N}_+$, we define the set of *expressions* $\mathrm{Exp}_n(K)$ over a semiring K by the grammar

$$E ::= x_i \mid y_i \mid E \oplus E \mid E \otimes E,$$

where $i \in \{1, \ldots, n\}$. The *(weighted) semantics* of an expression $E \in \mathrm{Exp}_n(K)$ is a mapping $\langle\!\langle E \rangle\!\rangle \colon K^n \times K^n \to K$ defined for $\bar\kappa, \bar\lambda \in K^n$ inductively by

$$\begin{aligned}
\langle\!\langle x_i \rangle\!\rangle(\bar\kappa, \bar\lambda) &= \kappa_i \\
\langle\!\langle y_i \rangle\!\rangle(\bar\kappa, \bar\lambda) &= \lambda_i \\
\langle\!\langle E_1 \oplus E_2 \rangle\!\rangle(\bar\kappa, \bar\lambda) &= \langle\!\langle E_1 \rangle\!\rangle(\bar\kappa, \bar\lambda) \oplus \langle\!\langle E_2 \rangle\!\rangle(\bar\kappa, \bar\lambda) \\
\langle\!\langle E_1 \otimes E_2 \rangle\!\rangle(\bar\kappa, \bar\lambda) &= \langle\!\langle E_1 \rangle\!\rangle(\bar\kappa, \bar\lambda) \odot \langle\!\langle E_2 \rangle\!\rangle(\bar\kappa, \bar\lambda).
\end{aligned}$$

For expressions over the Boolean semiring $\mathbb{B} = (\{0,1\}, \vee, \wedge, 0, 1)$ we will usually write $\vee$ instead of $\oplus$ and $\wedge$ instead of $\otimes$.

Construction 3.6. We call an expression $E \in \mathrm{Exp}_n(K)$ a *pure product* if

$$E = \mathrm{x}_1 \otimes \ldots \otimes \mathrm{x}_l \otimes \mathrm{y}_1 \otimes \ldots \otimes \mathrm{y}_m$$

with $\mathrm{x}_i \in \{x_1, \ldots, x_n\}$ for $i \in \{1, \ldots, l\}$ and $\mathrm{y}_j \in \{y_1, \ldots, y_n\}$ for $j \in \{1, \ldots, m\}$. We define a substitution procedure as follows. Let $\bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}} \in \mathrm{wMSO}(\sigma, K)^n$ be given. Let $i \in \{1, \ldots, l\}$ and assume $\mathrm{x}_i = x_k$ for some k, then we define $\xi_i = \varphi^{\mathbb{1}}_k$. Likewise, for $j \in \{1, \ldots, m\}$ and $\mathrm{y}_j = y_k$, we define $\theta_j = \varphi^{\mathbb{2}}_k$. We let $\xi = \xi_1 \otimes \ldots \otimes \xi_l$ and $\theta = \theta_1 \otimes \ldots \otimes \theta_m$. Then for $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, every $(\mathcal{V}, \mathfrak{A})$-assignment ρ, and every $(\mathcal{V}, \mathfrak{B})$-assignment ς we have

$$\begin{aligned}
&\langle\!\langle E \rangle\!\rangle([\![\bar\varphi^{\mathbb{1}}]\!](\mathfrak{A}, \rho), [\![\bar\varphi^{\mathbb{2}}]\!](\mathfrak{B}, \varsigma)) \\
&= [\![\xi_1]\!](\mathfrak{A}, \rho) \odot \ldots \odot [\![\xi_l]\!](\mathfrak{A}, \rho) \odot [\![\theta_1]\!](\mathfrak{B}, \varsigma) \odot \ldots \odot [\![\theta_m]\!](\mathfrak{B}, \varsigma) \\
&= [\![\xi]\!](\mathfrak{A}, \rho) \odot [\![\theta]\!](\mathfrak{B}, \varsigma).
\end{aligned}$$

We define $\mathrm{PRD}^{\mathbb{1}}(E, \bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}}) = \xi$ and $\mathrm{PRD}^{\mathbb{2}}(E, \bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}}) = \theta$.

Pure products $B \in \mathrm{Exp}_n(\mathbb{B})$ are also called *pure conjunctions*. For a pure conjunction $B \in \mathrm{Exp}_n(\mathbb{B})$, formulas $\bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}} \in \mathrm{MSO}(\sigma)$ and ξ_i, θ_j as above, we define the $\mathrm{MSO}(\sigma)$-formulas $\mathrm{CON}^{\mathbb{1}}(B, \bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}}) = \xi = \xi_1 \wedge \ldots \wedge \xi_l$ and $\mathrm{CON}^{\mathbb{2}}(B, \bar\varphi^{\mathbb{1}}, \bar\varphi^{\mathbb{2}}) = \theta = \theta_1 \wedge \ldots \wedge \theta_m$. We then have

$$\langle\!\langle B \rangle\!\rangle([\![\bar\varphi^{\mathbb{1}}]\!](\mathfrak{A}, \rho), [\![\bar\varphi^{\mathbb{2}}]\!](\mathfrak{B}, \varsigma)) = 1 \text{ iff } (\mathfrak{A}, \rho) \models \xi \text{ and } (\mathfrak{B}, \varsigma) \models \theta. \qquad \Diamond$$

We say that an expression $E \in \mathrm{Exp}_n(K)$ is in *normal form* if

$$E = E_1 \oplus \ldots \oplus E_m$$

for some $m \geq 1$ and pure products E_i. By applying the laws of distributivity of the semiring K, every expression $E \in \mathrm{Exp}_n(K)$ can be transformed into normal form. More precisely, we have the following lemma.

Lemma 3.7. *For every $E \in \mathrm{Exp}_n(K)$ there exists an expression $E' \in \mathrm{Exp}_n(K)$ in normal form with the same semantics as E.*

Proof. We proceed by induction. Let $E \in \mathrm{Exp}_n(K)$. If $E = x_i$ or $E = y_i$ for some $i \in \{1, \ldots, n\}$, then E is in normal form. If E is of the form $E_1 \oplus E_2$ or $E_1 \otimes E_2$ for two expressions $E_1, E_2 \in \mathrm{Exp}_n(K)$, we can find by induction two expressions $E_1', E_2' \in \mathrm{Exp}_n(K)$ in normal form with $\langle\langle E_1 \rangle\rangle = \langle\langle E_1' \rangle\rangle$ and $\langle\langle E_2 \rangle\rangle = \langle\langle E_2' \rangle\rangle$. In the first case, we see that $E' = E_1' \oplus E_2'$ is also in normal form and we have $\langle\langle E \rangle\rangle = \langle\langle E' \rangle\rangle$.

For the case that $E = E_1 \otimes E_2$, we write $E_1' = E_1^{(1)} \oplus \cdots \oplus E_l^{(1)}$ and $E_2' = E_1^{(2)} \oplus \cdots \oplus E_m^{(2)}$ with $E_1^{(1)}, \cdots, E_l^{(1)}$ and $E_1^{(2)}, \cdots, E_m^{(2)}$ pure products. Then we see that $E' = \bigoplus_{i=1}^{l} \bigoplus_{j=1}^{m} E_i^{(1)} \otimes E_j^{(2)}$ is in normal form and due to the distributivity of K, we have $\langle\langle E \rangle\rangle = \langle\langle E' \rangle\rangle$. □

3.2 The Classical Feferman-Vaught Theorem

In this section, we recall the Feferman-Vaught Theorem for disjoint unions and products of two structures. For convenience, we also provide the proofs for both cases. If the reader is familiar with the classical Feferman-Vaught Theorem, the proofs are safe to skip. For the rest of this section, let σ be a signature.

A Feferman-Vaught Theorem for disjoint unions

First, we state and prove the classical Feferman-Vaught Theorem for disjoint unions in the framework we will also employ for our weighted extension.

Theorem 3.8 ([44])**.** *Let $\mathcal{V}$ be a set of first and second order variables and $\beta \in \mathrm{MSO}(\sigma)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\beta}^{\mathbb{1}}, \bar{\beta}^{\mathbb{2}} \in \mathrm{MSO}(\sigma)^n$, and an expression $B_\beta \in \mathrm{Exp}_n(\mathbb{B})$ such that $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{\mathbb{2}}) \subseteq \mathrm{Free}(\beta)$ and for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ and all $(\mathcal{V}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ρ:*

$$(\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta \text{ iff } \langle\langle B_\beta \rangle\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{\mathbb{2}}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 1.$$

Proof. We proceed by induction.

■ $\beta = R(x_1, \ldots, x_k)$ for a relation symbol $R \in \mathrm{Rel}_\sigma$ of arity k
In this case, we let $n = 1$, $\beta_1^{\mathbb{1}} = \beta_1^{\mathbb{2}} = R(x_1, \ldots, x_k)$, and $B_\beta = x_1 \vee y_1$.

■ $\beta = (x \in X)$
In this case, we let $n = 1$, $\beta_1^{\mathbb{1}} = \beta_1^{\mathbb{2}} = (x \in X)$, and $B_\beta = x_1 \vee y_1$.

■ $\beta = \neg\alpha$
Assume the theorem is true for α with $\bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{2} \in \mathrm{MSO}(\sigma)^l$ and $B_\alpha \in \mathrm{Exp}_l(\mathbb{B})$. We may assume that $B_\alpha = B_1 \vee \ldots \vee B_m$ is in normal form with all B_i pure conjunctions. We let $\gamma_i = \mathrm{CON}^{\mathbb{1}}(B_i, \bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{2})$ and $\delta_i = \mathrm{CON}^{2}(B_i, \bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{2})$ (see Construction 3.6) and set

$$\begin{aligned}\bar{\beta}^{\mathbb{1}} &= (\neg\gamma_1, \ldots, \neg\gamma_m)\\ \bar{\beta}^{2} &= (\neg\delta_1, \ldots, \neg\delta_m)\\ B_\beta &= \bigwedge_{i=1}^{m}(x_i \vee y_i).\end{aligned}$$

Then we have

$$\begin{aligned}(\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta &\Leftrightarrow (\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \alpha \text{ does not hold}\\ &\Leftrightarrow \langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 0\\ &\Leftrightarrow \langle\!\langle B_i \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 0 \text{ for all } i \in \{1, \ldots, m\}\\ &\Leftrightarrow (\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \models \gamma_i \text{ does not hold or } (\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \models \delta_i \text{ does not hold}\\ &\quad \text{for all } i \in \{1, \ldots, m\}\\ &\Leftrightarrow \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.\end{aligned}$$

Furthermore, we have

$$\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\bar{\alpha}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\alpha}^{2}) \subseteq \mathrm{Free}(\alpha) = \mathrm{Free}(\beta).$$

■ $\beta = \alpha \vee \gamma$
Assume the theorem is true for α with $\bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{2} \in \mathrm{MSO}(\sigma)^l$ and $B_\alpha \in \mathrm{Exp}_l(\mathbb{B})$, and for γ with $\bar{\gamma}^{\mathbb{1}}, \bar{\gamma}^{2} \in \mathrm{MSO}(\sigma)^m$ and $B_\gamma \in \mathrm{Exp}_m(\mathbb{B})$. Then we set

$$\begin{aligned}\bar{\beta}^{\mathbb{1}} &= (\alpha_1^{\mathbb{1}}, \ldots, \alpha_l^{\mathbb{1}}, \gamma_1^{\mathbb{1}}, \ldots, \gamma_m^{\mathbb{1}})\\ \bar{\beta}^{2} &= (\alpha_1^{2}, \ldots, \alpha_l^{2}, \gamma_1^{2}, \ldots, \gamma_m^{2})\\ B_\beta &= B_\alpha \vee B'_\gamma,\end{aligned}$$

where B'_γ is obtained from B_γ by replacing every variable x_i by x_{i+l} and every variable y_i by y_{i+l}. Then we have

$$\begin{aligned}&(\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta\\ \Leftrightarrow\ &(\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \alpha \text{ or } (\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \gamma\\ \Leftrightarrow\ &\langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1 \text{ or } \langle\!\langle B_\gamma \rangle\!\rangle([\![\bar{\gamma}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\gamma}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1\\ \Leftrightarrow\ &\langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1 \text{ or } \langle\!\langle B'_\gamma \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1\\ \Leftrightarrow\ &\langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.\end{aligned}$$

Also, we have

$$\mathrm{Free}(\bar{\alpha}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\alpha}^{2}) \cup \mathrm{Free}(\bar{\gamma}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\gamma}^{2}) \subseteq \mathrm{Free}(\alpha) \cup \mathrm{Free}(\gamma) = \mathrm{Free}(\beta).$$

■ $\beta = \exists x.\alpha$
Assume the theorem is true for α with $\bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{\mathbb{2}} \in \mathrm{MSO}(\sigma)^l$ and $B_\alpha \in \mathrm{Exp}_l(\mathbb{B})$. We may assume that $B_\alpha = B_1 \vee \ldots \vee B_m$ is in normal form with all B_i pure conjunctions and that x does no occur as a bound variable in any of the $\alpha_i^{\mathbb{1}}$ or $\alpha_i^{\mathbb{2}}$. We let $\gamma_i = \mathrm{CON}^{\mathbb{1}}(B_i, \bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{\mathbb{2}})$ and $\delta_i = \mathrm{CON}^{\mathbb{2}}(B_i, \bar{\alpha}^{\mathbb{1}}, \bar{\alpha}^{\mathbb{2}})$ and set

$$\begin{aligned}
\bar{\beta}^{\mathbb{1}} &= (\exists x.\gamma_1, \ldots, \exists x.\gamma_m, \gamma_1^{-x}, \ldots, \gamma_m^{-x}) \\
\bar{\beta}^{\mathbb{2}} &= (\exists x.\delta_1, \ldots, \exists x.\delta_m, \delta_1^{-x}, \ldots, \delta_m^{-x}) \\
B_\beta &= \bigvee_{i=1}^{m} ((x_i \wedge y_{m+i}) \vee (x_{m+i} \wedge y_i)).
\end{aligned}$$

Then we have

$$\begin{aligned}
& (\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta \\
\Leftrightarrow\ & (\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \models \alpha \text{ for some } c \in A \sqcup B && \text{let } \rho_c = \rho[x \to c] \\
\Leftrightarrow\ & \langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_c{\restriction}_{\mathfrak{A}}), [\![\bar{\alpha}^{\mathbb{2}}]\!](\mathfrak{B}, \rho_c{\restriction}_{\mathfrak{B}})) = 1 \text{ for some } c \in A \sqcup B \\
\Leftrightarrow\ & \langle\!\langle B_i \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_c{\restriction}_{\mathfrak{A}}), [\![\bar{\alpha}^{\mathbb{2}}]\!](\mathfrak{B}, \rho_c{\restriction}_{\mathfrak{B}})) = 1 \text{ for some } c \in A \sqcup B \text{ and } i \in \{1, \ldots, m\} \\
\Leftrightarrow\ & (\mathfrak{A}, \rho_c{\restriction}_{\mathfrak{A}}) \models \gamma_i \text{ and } (\mathfrak{B}, \rho_c{\restriction}_{\mathfrak{B}}) \models \delta_i \text{ for some } c \in A \sqcup B \text{ and } i \in \{1, \ldots, m\} \\
\Leftrightarrow\ & (\mathfrak{A}, \rho{\restriction}_{\mathfrak{A}}[x \to a]) \models \gamma_i \text{ and } (\mathfrak{B}, \rho{\restriction}_{\mathfrak{B}}) \models \delta_i^{-x} \text{ for some } a \in A \text{ and } i \in \{1, \ldots, m\} \text{ or} \\
& (\mathfrak{A}, \rho{\restriction}_{\mathfrak{A}}) \models \gamma_i^{-x} \text{ and } (\mathfrak{B}, \rho{\restriction}_{\mathfrak{B}}[x \to b]) \models \delta_i \text{ for some } b \in B \text{ and } i \in \{1, \ldots, m\} \\
\Leftrightarrow\ & (\mathfrak{A}, \rho{\restriction}_{\mathfrak{A}}) \models \exists x.\gamma_i \text{ and } (\mathfrak{B}, \rho{\restriction}_{\mathfrak{B}}) \models \delta_i^{-x} \text{ for some } i \in \{1, \ldots, m\} \text{ or} \\
& (\mathfrak{A}, \rho{\restriction}_{\mathfrak{A}}) \models \gamma_i^{-x} \text{ and } (\mathfrak{B}, \rho{\restriction}_{\mathfrak{B}}) \models \exists x.\delta_i \text{ for some } i \in \{1, \ldots, m\} \\
\Leftrightarrow\ & \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction}_{\mathfrak{A}}), [\![\bar{\beta}^{\mathbb{2}}]\!](\mathfrak{B}, \rho{\restriction}_{\mathfrak{B}})) = 1.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{\mathbb{2}}) &\subseteq (\mathrm{Free}(\bar{\alpha}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\alpha}^{\mathbb{2}})) \setminus \{x\} \\
&\subseteq \mathrm{Free}(\alpha) \setminus \{x\} \\
&= \mathrm{Free}(\beta).
\end{aligned}$$

■ $\beta = \exists X.\alpha$
We reuse the notation from first order existential quantification and set

$$\begin{aligned}
\bar{\beta}^{\mathbb{1}} &= (\exists X.\gamma_1, \ldots, \exists X.\gamma_m) \\
\bar{\beta}^{\mathbb{2}} &= (\exists X.\delta_1, \ldots, \exists X.\delta_m) \\
B_\beta &= \bigvee_{i=1}^{m} (x_i \wedge y_i).
\end{aligned}$$

Then we have

$$\begin{aligned}
& (\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta \\
\Leftrightarrow\ & (\mathfrak{A} \sqcup \mathfrak{B}, \rho[X \to I]) \models \alpha \text{ for some } I \subseteq A \sqcup B && \text{let } \rho_I = \rho[X \to I] \\
\Leftrightarrow\ & \langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_I{\restriction}_{\mathfrak{A}}), [\![\bar{\alpha}^{\mathbb{2}}]\!](\mathfrak{B}, \rho_I{\restriction}_{\mathfrak{B}})) = 1 \text{ for some } I \subseteq A \sqcup B
\end{aligned}$$

$\Leftrightarrow \langle\!\langle B_i \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_I{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho_I{\restriction_{\mathfrak{B}}})) = 1$ for some $I \subseteq A \sqcup B$ and $i \in \{1, \dots, m\}$
$\Leftrightarrow (\mathfrak{A}, \rho_I{\restriction_{\mathfrak{A}}}) \models \gamma_i$ and $(\mathfrak{B}, \rho_I{\restriction_{\mathfrak{B}}}) \models \delta_i$ for some $I \subseteq A \sqcup B$ and $i \in \{1, \dots, m\}$
$\Leftrightarrow (\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[X \to I]) \models \gamma_i$ and $(\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[X \to J]) \models \delta_i$ for some $I \subseteq A$, $J \subseteq B$, and $i \in \{1, \dots, m\}$
$\Leftrightarrow (\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \models \exists X.\gamma_i$ and $(\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \models \exists X.\delta_i$ for some $i \in \{1, \dots, m\}$
$\Leftrightarrow \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.$

In the same way as for first order existential quantification, we obtain $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\beta)$. □

A Feferman-Vaught Theorem for products

Here, we state and prove the classical Feferman-Vaught Theorem for products in the framework we will also employ for our weighted extension.

Theorem 3.9 ([44]). *Let $\mathcal{V}$ be a set of first and second order variables and $\beta \in \mathrm{FO}(\sigma)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\beta}^{\mathbb{1}}, \bar{\beta}^{2} \in \mathrm{FO}(\sigma)^n$, and an expression $B_\beta \in \mathrm{Exp}_n(\mathbb{B})$ such that $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\beta)$ and for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ and all $(\mathcal{V}, \mathfrak{A} \times \mathfrak{B})$-assignments ρ:*

$$(\mathfrak{A} \times \mathfrak{B}, \rho) \models \beta \text{ iff } \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.$$

Proof. We proceed by induction.

■ $\beta = R(x_1, \dots, x_k)$ for a relation symbol $R \in \mathrm{Rel}_\sigma$ of arity k
In this case, we let $n = 1$, $\beta^{\mathbb{1}}_1 = \beta^{2}_1 = R(x_1, \dots, x_k)$, and $B_\beta = x_1 \wedge y_1$.

■ The proofs for $\beta = \neg\alpha$ and $\beta = \alpha \vee \gamma$ are the same as in Theorem 3.8.

■ $\beta = \exists x.\alpha$
We reuse the notation from first order existential quantification of Theorem 3.8 and set

$$\begin{aligned} \bar{\beta}^{\mathbb{1}} &= (\exists x.\gamma_1, \dots, \exists x.\gamma_m) \\ \bar{\beta}^{2} &= (\exists x.\delta_1, \dots, \exists x.\delta_m) \\ B_\beta &= \bigvee_{i=1}^{m} (x_i \wedge y_i). \end{aligned}$$

Then we have

$(\mathfrak{A} \times \mathfrak{B}, \rho) \models \beta$
$\Leftrightarrow (\mathfrak{A} \times \mathfrak{B}, \rho[x \to c]) \models \alpha$ for some $c \in A \times B$ let $\rho_c = \rho[x \to c]$
$\Leftrightarrow \langle\!\langle B_\alpha \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_c{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho_c{\restriction_{\mathfrak{B}}})) = 1$ for some $c \in A \times B$
$\Leftrightarrow \langle\!\langle B_i \rangle\!\rangle([\![\bar{\alpha}^{\mathbb{1}}]\!](\mathfrak{A}, \rho_c{\restriction_{\mathfrak{A}}}), [\![\bar{\alpha}^{2}]\!](\mathfrak{B}, \rho_c{\restriction_{\mathfrak{B}}})) = 1$ for some $c \in A \times B$ and $i \in \{1, \dots, m\}$
$\Leftrightarrow (\mathfrak{A}, \rho_c{\restriction_{\mathfrak{A}}}) \models \gamma_i$ and $(\mathfrak{B}, \rho_c{\restriction_{\mathfrak{B}}}) \models \delta_i$ for some $c \in A \times B$ and $i \in \{1, \dots, m\}$

$\Leftrightarrow (\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[x \to a]) \models \gamma_i$ and $(\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[x \to b]) \models \delta_i$ for some $a \in A$, $b \in B$, and $i \in \{1, \ldots, m\}$

$\Leftrightarrow (\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \models \exists x.\gamma_i$ and $(\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \models \exists x.\delta_i$ for some $i \in \{1, \ldots, m\}$

$\Leftrightarrow \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.$

Again, we easily obtain $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\beta)$. □

3.3 Translation Schemes

Theorems 3.8 and 3.9 consider disjoint unions and products only. So far, there is no interaction between the two constituting structures. *Translation schemes* allow us to create such interactions in an MSO-defined manner. More precisely, translation schemes "translate" structures over one signature into structures over another signature. Applying this to disjoint unions and products, we can extend Theorems 3.8 and 3.9 to more complex constructs. The usefulness of such extensions by translation schemes was discussed in [70], which we follow here.

Let σ and τ be two signatures, $\mathcal{Z} = \{z, z_1, z_2, \ldots\}$ be a set of distinguished first order variables, and $\mathcal{W}$ be a set of first and second order variables with $\mathcal{W} \cap \mathcal{Z} = \emptyset$. A *$\sigma$-$\tau$-translation scheme Φ over $\mathcal{W}$ and $\mathcal{Z}$* is a pair $(\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ where $\phi_{\mathcal{U}}, \phi_T \in \mathrm{MSO}(\sigma)$ for each $T \in \mathrm{Rel}_\tau$, $\phi_{\mathcal{U}}$ has variables from $\mathcal{W} \cup \{z\}$, and each ϕ_T has variables from $\mathcal{W} \cup \{z_1, \ldots, z_{\mathrm{ar}_\tau(T)}\}$. The variables from $\mathcal{Z}$ may not be used for quantification, i.e., all variables from $\mathcal{Z}$ must be free.

Intuitively, the formula $\phi_{\mathcal{U}}$ is a filter for the new universe, i.e., the universe of our new τ-structure will contain all elements a of our σ-structure which satisfy $\phi_{\mathcal{U}}$ when z is mapped to a. Likewise, every formula ϕ_T defines the relation T of our new τ-structure, i.e., the interpretation of T will contain all tuples $(a_1, \ldots, a_{\mathrm{ar}_\tau(T)})$ which satisfy ϕ_T when each z_i is mapped to a_i.

We set $\mathrm{Free}(\Phi) = \mathrm{Free}(\phi_{\mathcal{U}}) \cup \bigcup_{T \in \mathrm{Rel}_\tau} \mathrm{Free}(\phi_T)$. The formulas $\phi_{\mathcal{U}}$ and $(\phi_T)_{T \in \mathrm{Rel}_\tau}$ depend on $\mathcal{Z}$ in the following way. For a first order variable x not occurring in $\phi_{\mathcal{U}}$, the formula $\phi_{\mathcal{U}}(x)$ is obtained from $\phi_{\mathcal{U}}$ by replacing all occurrences of z by x. Similarly, for $T \in \mathrm{Rel}_\tau$ and first order variables $x_1, \ldots, x_{\mathrm{ar}_\tau(T)}$ not occurring in ϕ_T, the formula $\phi_T(x_1, \ldots, x_{\mathrm{ar}_\tau(T)})$ is obtained from ϕ_T by replacing all occurrences of z_i by x_i for $i \in \{1, \ldots, \mathrm{ar}_\tau(T)\}$.

For a σ-structure $\mathfrak{A} = (A, \mathcal{I}_{\mathfrak{A}})$ and a $(\mathcal{W}, \mathfrak{A})$-assignment ς, we define the *Φ-induced τ-structure of $\mathfrak{A}$ and ς*, denoted by $\Phi^\star(\mathfrak{A}, \varsigma)$, as a τ-structure with universe $\mathcal{U}_{\mathfrak{C}}$ and interpretation $\mathcal{I}_{\mathfrak{C}}$ as follows.

$$\mathcal{U}_{\mathfrak{C}} = \{a \in A \mid (\mathfrak{A}, \varsigma[z \to a]) \models \phi_{\mathcal{U}}\}$$
$$\mathcal{I}_{\mathfrak{C}}(T) = \{\bar{c} \in \mathcal{U}_{\mathfrak{C}}^{\mathrm{ar}_\tau(T)} \mid (\mathfrak{A}, \varsigma[\bar{z} \to \bar{c}]) \models \phi_T\}$$

Here, $\bar{z}$ always denotes the tuple $(z_1, \ldots, z_{\mathrm{ar}_\tau(T)})$.

Example 3.10. We can use a translation scheme to connect a specified vertex in a graph to a set of vertices of the graph. For this let $\sigma = \tau = (\{\mathrm{edge}\}, \mathrm{edge} \mapsto 2)$ be

the signature of a directed graph like in Example 2.2. We define a σ-σ-translation scheme $\Phi = (\phi_{\mathcal{U}}, \phi_{\text{edge}})$ through

$$\begin{aligned}\phi_{\mathcal{U}} &= \texttt{true}\\ \phi_{\text{edge}} &= \text{edge}(z_1, z_2) \vee (z_1 = x \wedge z_2 \in X),\end{aligned}$$

where $z_1 = x$ is an abbreviation for $\forall Y(z_1 \in Y \to x \in Y)$. Let $\mathfrak{G} = (V, \text{edge} \mapsto E) \in \text{Str}(\sigma)$ be a graph, $v \in V$ a vertex, and $I \subseteq V$ a set of vertices. Then the graph $\Phi^*(\mathfrak{G}, \{x \mapsto v, X \mapsto I\})$ is exactly the graph $\mathfrak{G}$ with an edge added between v and every vertex $v' \in I$.

Example 3.11. A translation scheme can also be used to cut a subtree from a given tree at a specified vertex in the tree. As in the previous example, let σ be the signature of a directed graph. For a graph $\mathfrak{G} = (V, \text{edge} \mapsto E) \in \text{Str}(\sigma)$, let E' be the transitive closure of the relation $E \subseteq V \times V$. We say that $\mathfrak{G}$ is a directed rooted tree with root $r \in V$ if (1) E' is irreflexive, (2) $(r, v) \in E'$ for all $v \in V \setminus \{r\}$ and (3) for all $v \in V \setminus \{r\}$ there is exactly one $v' \in V$ with $(v', v) \in E$. We define the following abbreviation which describes the reflexive transitive closure of E.

$$(x \leq y) = \forall X\Big(\big(x \in X \wedge \big(\forall z.(\exists z'.z' \in X \wedge \text{edge}(z', z)) \to z \in X\big)\big) \to y \in X\Big)$$

We define a σ-σ-translation scheme $\Phi = (\phi_{\mathcal{U}}, \phi_{\text{edge}})$ through

$$\begin{aligned}\phi_{\mathcal{U}} &= (x \leq z)\\ \phi_{\text{edge}} &= \text{edge}(z_1, z_2).\end{aligned}$$

Then with $\mathfrak{G}$ as above and $v \in V$, the graph $\mathfrak{C} = \Phi^*(\mathfrak{G}, x \mapsto v)$ is the subtree of $\mathfrak{G}$ at the vertex v, i.e.,

$$\begin{aligned}\mathcal{U}_{\mathfrak{C}} &= \{v\} \cup \{v' \in V \mid (v, v') \in E'\}\\ \mathcal{I}_{\mathfrak{C}} &= E \cap (\mathcal{U}_{\mathfrak{C}} \times \mathcal{U}_{\mathfrak{C}}).\end{aligned}$$

We have the following fundamental property of translation schemes [70].

Lemma 3.12 ([70])**.** *Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \text{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\beta \in \text{MSO}(\tau)$ with variables from $\mathcal{V}$. Then there exists a formula $\alpha \in \text{MSO}(\sigma)$ such that $\text{Free}(\alpha) \subseteq \text{Free}(\beta) \cup \text{Free}(\Phi)$ and for all structures $\mathfrak{A} \in \text{Str}(\sigma)$, all $(\mathcal{W}, \mathfrak{A})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A}, \varsigma))$-assignments ρ:*

$$(\Phi^\star(\mathfrak{A}, \varsigma), \rho) \models \beta \text{ iff } (\mathfrak{A}, \varsigma \cup \rho) \models \alpha.$$

Proof. We indicate the proof for the convenience of the reader. We proceed by induction. In the following, we will assume that for formulas β', β_1, and β_2, the theorem holds by induction with the formulas α', α_1 and α_2, respectively.

For $\beta = (x \in X)$, we let $\alpha = (x \in X)$. For $\beta = T(x_1, \ldots, x_k)$ for some $T \in \text{Rel}_\tau$, we let $\alpha = \phi_T(x_1, \ldots, x_k)$. For $\beta = \neg\beta'$, we let $\alpha = \neg\alpha'$. For $\beta = \beta_1 \vee \beta_2$, we let $\alpha = \alpha_1 \vee \alpha_2$. For $\beta = \exists x.\beta'$, we let $\alpha = \exists x.(\alpha' \wedge \phi_{\mathcal{U}}(x))$ and for $\beta = \exists X.\beta'$, we let $\alpha = \exists X.(\alpha' \wedge \forall x.(x \in X \to \phi_{\mathcal{U}}(x)))$. □

Together with Theorems 3.8 and 3.9, this gives us the following Feferman-Vaught decomposition theorems for disjoint unions and products with translations schemes.

Theorem 3.13 ([70]). *Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\beta \in \mathrm{MSO}(\tau)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\beta}^{\mathbb{1}}, \bar{\beta}^{2} \in \mathrm{MSO}(\sigma)^n$, and an expression $B_\beta \in \mathrm{Exp}_n(\mathbb{B})$ such that $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\beta) \cup \mathrm{Free}(\Phi)$ and for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, all $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma))$-assignments ρ:*

$$(\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) \models \beta \text{ iff } \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})) = 1.$$

Proof. By Lemma 3.12 we know that there is a formula $\alpha \in \mathrm{MSO}(\sigma)$ such that

$$(\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) \models \beta \text{ iff } (\mathfrak{A} \sqcup \mathfrak{B}, \varsigma \cup \rho) \models \alpha.$$

We then use Theorem 3.8 for the formula α to obtain $n \geq 1$, tuples of formulas $\bar{\beta}^{\mathbb{1}}, \bar{\beta}^{2} \in \mathrm{MSO}(\sigma)^n$, and an expression $B_\beta \in \mathrm{Exp}_n(\mathbb{B})$ as required. □

Theorem 3.14 ([70]). *Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\beta \in \mathrm{FO}(\tau)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\beta}^{\mathbb{1}}, \bar{\beta}^{2} \in \mathrm{FO}(\sigma)^n$, and an expression $B_\beta \in \mathrm{Exp}_n(\mathbb{B})$ such that $\mathrm{Free}(\bar{\beta}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\beta}^{2}) \subseteq \mathrm{Free}(\beta) \cup \mathrm{Free}(\Phi)$ and for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, all $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma))$-assignments ρ:*

$$(\Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma), \rho) \models \beta \text{ iff } \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})) = 1.$$

Proof. We proceed as in the proof of Theorem 3.13 and combine Lemma 3.12 and Theorem 3.9. □

Example 3.15. We consider the signature σ of a labeled graph, i.e., $\mathrm{Rel}_\sigma = \{\mathrm{edge}, \mathrm{label}_a, \mathrm{label}_b\}$ where edge has arity 2 and $\mathrm{label}_a, \mathrm{label}_b$ both have arity 1. Given two directed rooted labeled trees $\mathfrak{G}_1, \mathfrak{G}_2$ in this signature (see Example 3.11), we can use a translation scheme to add edges between all leaves of $\mathfrak{G}_1$ and the root of $\mathfrak{G}_2$ in $\mathfrak{G}_1 \sqcup \mathfrak{G}_2$. For this scenario, we have to distinguish between the vertices from the first and the second graph, so the use of an intermediate signature is necessary. We define the signature σ' to be σ extended by the relation symbols G_1 and G_2 of arity 1. Then for $i \in \{1, 2\}$, we define a σ-σ'-translation scheme $\Phi_i = (\phi_{\mathcal{U}}, \phi'_{\mathrm{edge}}, \phi_{\mathrm{label}_a}, \phi_{\mathrm{label}_b}, \phi^i_{G_1}, \phi^i_{G_2})$ as

$$\begin{aligned}
\phi_{\mathcal{U}} &= \mathtt{true} \\
\phi'_{\mathrm{edge}} &= \mathrm{edge}(z_1, z_2) \\
\phi_{\mathrm{label}_a} &= \mathrm{label}_a(z_1) \\
\phi_{\mathrm{label}_b} &= \mathrm{label}_b(z_1) \\
\phi^i_{G_j} &= \begin{cases} \mathtt{true} & \text{if } i = j \\ \mathtt{false} & \text{otherwise.} \end{cases}
\end{aligned}$$

With the abbreviations

$$\mathrm{root}(x) = \neg\exists y.\mathrm{edge}(y,x)$$
$$\mathrm{leaf}(x) = \neg\exists y.\mathrm{edge}(x,y)$$

we then define the σ'-σ-translation scheme $\Phi = (\phi_{\mathcal{U}}, \phi_{\mathrm{edge}}, \phi_{\mathrm{label}_a}, \phi_{\mathrm{label}_b})$ through

$$\phi_{\mathrm{edge}} = \mathrm{edge}(z_1,z_2) \vee (G_1(z_1) \wedge G_2(z_2) \wedge \mathrm{leaf}(z_1) \wedge \mathrm{root}(z_2)).$$

Then $\mathfrak{G} = \Phi^*(\Phi_1^*(\mathfrak{G}_1) \sqcup \Phi_2^*(\mathfrak{G}_2))$ is exactly $\mathfrak{G}_1 \sqcup \mathfrak{G}_2$ with the leaves of $\mathfrak{G}_1$ connected to the root of $\mathfrak{G}_2$. We now consider the formula

$$\beta = \exists x.\exists y.(\mathrm{edge}(x,y) \wedge \mathrm{label}_a(x) \wedge \mathrm{label}_b(y))$$

which asks whether there is some edge between an a-labeled and a b-labeled vertex. We can apply Lemma 3.12 and Theorem 3.13 to obtain the following decomposition of β. Let

$$\bar{\beta}^{\mathbb{1}} = (\beta, \exists x.\mathrm{label}_a(x) \wedge \mathrm{leaf}(x))$$
$$\bar{\beta}^{2} = (\beta, \exists y.\mathrm{label}_b(y) \wedge \mathrm{root}(y))$$
$$B_\beta = x_1 \vee y_1 \vee (x_2 \wedge y_2).$$

Then we have

$$\mathfrak{G} \models \beta \text{ iff } \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{G}_1), [\![\bar{\beta}^{2}]\!](\mathfrak{G}_2)) = 1.$$

3.4 Weighted Feferman-Vaught Theorems

Our goal is to prove weighted versions of Theorems 3.13 and 3.14. That is, we would like to replace FO by wFO and MSO by wMSO in those theorems. This, however, is not possible as we will see in Sections 3.4 and 3.4. For disjoint unions, we have to restrict the use of the first order product quantifier and entirely remove the second order product quantifier in wMSO. For products, it is not possible to include the first order product quantifier at all.

Formulation of the theorems

Let σ be a signature and K a commutative semiring. We define two fragments of our logic and formulate our weighted versions of Theorems 3.13 and 3.14 for these fragments.

Definition 3.16 (Product-free weighted first order logic)**.** We define the *product-free* first order fragment of our logic through the grammar

$$\varphi ::= \beta \mid \kappa \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi,$$

where $\beta \in \mathrm{FO}(\sigma)$ is a first order formula, $\kappa \in K$, and x is a first order variable. By $\mathrm{wFO}^{\bigotimes\text{-free}}(\sigma, K)$, we denote the set of all such formulas. In fact, $\mathrm{wFO}^{\bigotimes\text{-free}}(\sigma, K)$ is the set of all formulas from $\mathrm{wFO}(\sigma, K)$ which do not contain any first order product quantifier. Using this fragment, we will formulate a weighted Feferman-Vaught decomposition theorem for products of structures.

Definition 3.17 (Product-restricted weighted monadic second order logic)**.** In order to define the *product-restricted* fragment of our weighted monadic second order logic, we first define the fragment of so-called *almost Boolean* formulas through the grammar

$$\psi ::= \beta \mid \kappa \mid \psi \oplus \psi \mid \psi \otimes \psi,$$

where $\beta \in \mathrm{MSO}(\sigma)$ is a monadic second order formula and $\kappa \in K$. This fragment, which we denote by $\mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$, already appeared in [27] in the form of *recognizable step functions*. To obtain the main theorem of [27], the product quantifier was restricted to quantify only over recognizable step functions. We employ the same restriction and define the product-restricted fragment of our logic through the grammar

$$\varphi ::= \beta \mid \kappa \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\psi \mid \bigoplus X.\varphi,$$

where $\beta \in \mathrm{MSO}(\sigma)$ is a monadic second order formula, $\kappa \in K$, x is a first order variable, X is a second order variable, and $\psi \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$ is an almost Boolean formula. By $\mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)$ we denote the set of all such formulas. The set $\mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)$ contains all formulas from $\mathrm{wMSO}(\sigma, K)$ which do not contain any second order quantifier and where for every subformula of the form $\bigotimes x.\psi$ we have that ψ is an almost Boolean formula. Our weighted Feferman-Vaught decomposition theorem for disjoint unions of structures will be formulated for this fragment. In [27] it was shown that for finite and infinite words, this fragment is expressively equivalent to weighted finite automata.

We note that the restrictions we impose on the product quantifier are necessary as we will show in Sections 3.4 and 3.4. We formulate the weighted versions of Theorems 3.13 and 3.14 as follows.[1] Let τ, $\mathcal{W}$, and $\mathcal{Z}$ be as in Section 3.3.

Theorem 3.18. *Let K be a commutative semiring. Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wMSO}^{\bigotimes\text{-res}}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in \mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)^n$ with $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^{2}) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$ such that the following holds. For all finite structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, or, for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ if K is bicomplete, all $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})).$$

Theorem 3.19. *Let K be a commutative semiring. Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wFO}^{\bigotimes\text{-free}}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in \mathrm{wFO}^{\bigotimes\text{-free}}(\sigma, K)^n$ with $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^{2}) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$*

[1] In [91] a weighted version of Theorem 3.13 similar to ours is stated (without proof) to hold without any restriction on the first order product quantifier. However, in Section 3.4 we show that a restriction on the product quantifier is necessary.

such that the following holds. For all finite structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$*, or, for all structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ *if* K *is bicomplete, all* $(\mathcal{W}, \mathfrak{A} \times \mathfrak{B})$*-assignments* ς*, and all* $(\mathcal{V}, \Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma))$*-assignments* ρ *we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{\mathbb{2}}]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})).$$

The proofs of both theorems are deferred to Section 3.4. For formulas without free variables and a trivial translation scheme, i.e., $\sigma = \tau$, $\phi_{\mathcal{U}} = \mathtt{true}$, and $\phi_T = T(z_1, \ldots, z_{\mathrm{ar}_\tau(T)})$ for all $T \in \mathrm{Rel}_\tau$, the theorems reduce to the following, simplified versions.

Theorem 3.20. *Let* K *be a commutative semiring and* $\varphi \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)$ *be a sentence. Then there exist* $n \geq 1$*, tuples of sentences* $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{\mathbb{2}} \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)^n$*, and an expression* $E_\varphi \in \mathrm{Exp}_n(K)$ *such that the following holds. For all finite structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$*, or, for all structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ *if* K *is bicomplete, we have*

$$[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}), [\![\bar{\varphi}^{\mathbb{2}}]\!](\mathfrak{B})).$$

Theorem 3.21. *Let* K *be a commutative semiring and* $\varphi \in \mathrm{wFO}^{\otimes\text{-free}}(\sigma, K)$ *be a sentence. Then there exist* $n \geq 1$*, tuples of sentences* $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{\mathbb{2}} \in \mathrm{wFO}^{\otimes\text{-free}}(\sigma, K)^n$*, and an expression* $E_\varphi \in \mathrm{Exp}_n(K)$ *such that the following holds. For all finite structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$*, or, for all structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ *if* K *is bicomplete, we have*

$$[\![\varphi]\!](\mathfrak{A} \times \mathfrak{B}) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}), [\![\bar{\varphi}^{\mathbb{2}}]\!](\mathfrak{B})).$$

Example 3.22. To illustrate Theorem 3.20, we consider the semiring of natural numbers $(\mathbb{N}, +, \cdot, 0, 1)$ and the signature σ of a labeled graph, i.e., $\mathrm{Rel}_\sigma = \{\mathrm{edge}, \mathrm{label}_a, \mathrm{label}_b\}$ with edge binary and $\mathrm{label}_a, \mathrm{label}_b$ both unary. Consider the following formula which multiplies the number of vertices labeled a with the number of edges between two vertices labeled b.

$$\left(\underbrace{\bigoplus x.\mathrm{label}_a(x)}_{=\varphi_a}\right) \otimes \left(\underbrace{\bigoplus x.\bigoplus y.\mathrm{edge}(x,y) \wedge \mathrm{label}_b(x) \wedge \mathrm{label}_b(y)}_{=\varphi_b}\right)$$

The formula can be decomposed as follows. Let

$$\bar{\varphi}^{\mathbb{1}} = \bar{\varphi}^{\mathbb{2}} = (\varphi_a, \varphi_b)$$
$$E_\varphi = (x_1 \oplus y_1) \otimes (x_2 \oplus y_2).$$

Then for every two σ-structures $\mathfrak{G}_1, \mathfrak{G}_2$ we have

$$[\![\varphi]\!](\mathfrak{G}_1 \sqcup \mathfrak{G}_2) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{G}_1), [\![\bar{\varphi}^{\mathbb{2}}]\!](\mathfrak{G}_2)).$$

Example 3.23. In Example 3.5, we interpreted $[\![\varphi]\!](\mathfrak{A}, \rho)$ as the number of proofs we have that $(\mathfrak{A}, \rho)$ satisfies φ, assuming that φ does not contain constants. Applying Theorem 3.18 in this scenario means that the number of proofs that $(\mathfrak{A} \sqcup \mathfrak{B}, \rho)$ satisfies a formula φ can be computed from the number of proofs we have that $(\mathfrak{A}, \rho\restriction_{\mathfrak{A}})$ satisfies some formulas $\varphi_1^{\mathbb{1}}, \ldots, \varphi_n^{\mathbb{1}}$ and the number of proofs we have that $(\mathfrak{B}, \rho\restriction_{\mathfrak{B}})$ satisfies some formulas $\varphi_1^{\mathbb{2}}, \ldots, \varphi_n^{\mathbb{2}}$ by combining these numbers only through an expression.

Example 3.24. In [91], it is discussed how translation schemes can be applied for Feferman-Vaught-like decompositions of weighted properties. Theorems 3.18 and 3.19 show that this is possible for all properties which can be expressed by formulas in our weighted logic fragments.

Necessity of restricting the logic for disjoint unions

In this section, we show that the restrictions we impose on the product quantifiers are indeed necessary. For disjoint unions, we will prove that already Theorem 3.20 does not hold over the min-plus semiring $\mathbb{R}_{\min} = (\mathbb{R} \cup \{\infty\}, \min, +, \infty, 0)$ and over the max-plus semiring $\mathbb{R}_{\max} = (\mathbb{R} \cup \{-\infty\}, \max, +, -\infty, 0)$ for the formulas $\bigotimes x.\bigotimes y.1$ and $\bigotimes X.1$. To prove this, we employ Ramsey's theorem. Then we show that for the formula $\bigotimes x.\bigoplus y.1$, Theorem 3.20 does not hold over the semiring $(\mathbb{N}, +, \cdot, 0, 1)$. We note that these types of formulas also occurred in [27] and [33] as examples of weighted formulas whose semantics cannot be described by weighted automata.

We will employ the following version of Ramsey's theorem. For a set X, we denote by $\left[\frac{X}{2}\right]$ the set of all subsets of X of size 2.

Theorem 3.25 ([90])**.** *Let X be an infinite set, $k \geq 1$ a positive integer, and $f\colon \left[\frac{\mathbb{N}}{2}\right] \to \{1, \ldots, k\}$ a mapping. Then there exists an infinite subset $E \subseteq \mathbb{N}$ such that $f\restriction_{\left[\frac{E}{2}\right]} \equiv i$ for some $i \in \{1, \ldots, k\}$.*

Theorem 3.26. *Let $K \in \{\mathbb{R}_{\min}, \mathbb{R}_{\max}\}$, $\sigma = (\emptyset, \emptyset)$ be the empty signature, and for $l \in \mathbb{N}_+$ consider the σ-structures $\mathfrak{S}_l = (\{1, \ldots, l\}, \emptyset)$. Then for $\varphi = \bigotimes x.\bigotimes y.1$ there do not exist $n \geq 1$, $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in (\mathrm{wMSO}(\sigma, K))^n$, and $E_\varphi \in \mathrm{Exp}_n(K)$ such that for all $l, m \in \mathbb{N}_+$ we have*

$$[\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{S}_l), [\![\bar{\varphi}^{2}]\!](\mathfrak{S}_m)). \tag{3.4.1}$$

Proof. First, consider $K = \mathbb{R}_{\min}$. For contradiction, suppose that $n, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2}$, and E_φ as above satisfying (3.4.1) exist. We may assume that $E_\varphi = E_1 \oplus \ldots \oplus E_k$ is in normal form with all E_i pure products. For $l \geq 1$ and $i \in \{1, \ldots, k\}$ we let $a_{li} = [\![\mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2})]\!](\mathfrak{S}_l)$ and $b_{li} = [\![\mathrm{PRD}^{2}(E_i, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2})]\!](\mathfrak{S}_l)$. Then by assumption we have

$$(l+m)^2 = [\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \min_{i=1}^{k}(a_{li} + b_{mi}). \tag{3.4.2}$$

Given $l \geq 1$ and $m \geq 1$, for at least one index $j \in \{1, \ldots, k\}$ we have $(l+m)^2 = a_{lj} + b_{mj}$. We define j_{lm} as the smallest such index. Then we define a function $f\colon \left[\frac{\mathbb{N}_+}{2}\right] \to \{1, \ldots, k\}$ by $f(\{l, m\}) = j_{lm}$ for $l < m$. Then we take $E \subseteq \mathbb{N}_+$ according to Ramsey's theorem. As E is infinite, there are $l, \lambda, m, \mu \in E$ with $l < \lambda < m < \mu$. With $j = j_{lm}$, we thus have

$$\begin{aligned}(l+m)^2 &= a_{lj} + b_{mj}\\ (\lambda+m)^2 &= a_{\lambda j} + b_{mj}\\ (l+\mu)^2 &= a_{lj} + b_{\mu j}\\ (\lambda+\mu)^2 &= a_{\lambda j} + b_{\mu j}.\end{aligned}$$

This implies that

$$\begin{aligned}(\lambda+\mu)^2 &= (\lambda+m)^2 + (l+\mu)^2 - (l+m)^2\\ &= \lambda^2+\mu^2+2\lambda m+2l\mu-2lm\\ &= (\lambda+\mu)^2 - 2(\lambda-l)(\mu-m)\\ &< (\lambda+\mu)^2,\end{aligned}$$

a contradiction. Therefore, $n, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2}$, and E_φ as chosen cannot exist.

The proof for the max-plus semiring is in fact identical, the only difference is that equations (3.4.2) become

$$(l+m)^2 = [\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \max_{i=1}^{k}(a_{li}+b_{mi}).$$

□

Theorem 3.27. *Let $K \in \{\mathbb{R}_{\min}, \mathbb{R}_{\max}\}$, $\sigma = (\emptyset, \emptyset)$ be the empty signature, and for $l \in \mathbb{N}_+$ consider the σ-structures $\mathfrak{S}_l = (\{1,\dots,l\}, \emptyset)$. Then for $\varphi = \bigotimes X.1$ there do not exist $n \geq 1$, $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in (\mathrm{wMSO}(\sigma, K))^n$, and $E_\varphi \in \mathrm{Exp}_n(K)$ such that for all $l, m \in \mathbb{N}_+$ we have*

$$[\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{S}_l), [\![\bar{\varphi}^{2}]\!](\mathfrak{S}_m)).$$

Proof. We proceed as in the proof of Theorem 3.26 and by contradiction obtain a system of equations

$$2^{l+m} = [\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \min_{i=1}^{k}(a_{li}+b_{mi}).$$

Also employing Ramsey's theorem in the same way, we obtain $l < \lambda < m < \mu$ and $j \in \{1,\dots,k\}$ such that

$$\begin{aligned}2^{l+m} &= a_{lj} + b_{mj}\\ 2^{\lambda+m} &= a_{\lambda j} + b_{mj}\\ 2^{l+\mu} &= a_{lj} + b_{\mu j}\\ 2^{\lambda+\mu} &= a_{\lambda j} + b_{\mu j},\end{aligned}$$

which gives us the equality

$$2^{\lambda+\mu} = 2^{\lambda+m} + 2^{l+\mu} - 2^{l+m}.$$

By dividing by 2^{l+m} we obtain

$$2^{(\lambda-l)+(\mu-m)} = 2^{\lambda-l} + 2^{\mu-m} - 1. \tag{3.4.3}$$

However, we have

$$\begin{aligned}2^{(\lambda-l)+(\mu-m)} &\geq 2^{\lambda-l} + 2^{\mu-m}\\ &> 2^{\lambda-l} + 2^{\mu-m} - 1,\end{aligned}$$

which contradicts equation (3.4.3). □

Theorem 3.28. *Let $K = (\mathbb{N}, +, \cdot, 0, 1)$, $\sigma = (\emptyset, \emptyset)$ be the empty signature, and for $l \in \mathbb{N}_+$ consider the σ-structures $\mathfrak{S}_l = (\{1, \dots, l\}, \emptyset)$. Then for $\varphi = \bigotimes x. \bigoplus y. 1$ there do not exist $n \geq 1$, $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^2 \in (\mathrm{wMSO}(\sigma, \mathbb{N}))^n$, and $E_\varphi \in \mathrm{Exp}_n(\mathbb{N})$ such that for all $l, m \in \mathbb{N}_+$ we have*

$$[\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{S}_l), [\![\bar{\varphi}^2]\!](\mathfrak{S}_m)). \tag{3.4.4}$$

Proof. We proceed by contradiction and assume $n, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^2$, and E_φ as above satisfying (3.4.4) exist. We may assume that $E_\varphi = E_1 \oplus \ldots \oplus E_k$ is in normal form with all E_i pure products. For $l \geq 1$ and $i \in \{1, \dots, k\}$ we let $a_{li} = [\![\mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^2)]\!](\mathfrak{S}_l)$ and $b_{li} = [\![\mathrm{PRD}^2(E_i, \bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^2)]\!](\mathfrak{S}_l)$. Then by assumption we have

$$(l+m)^{(l+m)} = [\![\varphi]\!](\mathfrak{S}_l \sqcup \mathfrak{S}_m) = \sum_{i=1}^{k} (a_{li} \cdot b_{mi}). \tag{3.4.5}$$

For every $j \in \{1, \dots, k\}$ we choose $L_j \geq 1$ such that $a_{L_j j} \neq 0$, or let $L_j = 0$ if for all $l \geq 1$ we have $a_{lj} = 0$. Assume $m \geq 1$ and $j \in \{1, \dots, k\}$ with $L_j \neq 0$, then $a_{L_j j} \geq 1$, hence

$$(L_j + m)^{(L_j+m)} = \sum_{i=1}^{k} (a_{L_j i} \cdot b_{mi}) \geq (a_{L_j j} \cdot b_{mj}) \geq b_{mj}.$$

In particular, with $L = \max\{L_i \mid i \in \{1, \dots, k\}\}$, we have that for every $j \in \{1, \dots, k\}$ either (i) $b_{mj} \leq (L+m)^{(L+m)}$ for all $m \geq 1$ or (ii) $a_{lj} = 0$ for all $l \geq 1$. Note that from equation (3.4.5) it follows that $L = 0$ is impossible. In the same fashion, we can find $M \geq 1$ such that for every $l \geq 1$ and every $j \in \{1, \dots, k\}$ either (i) $a_{lj} \leq (l+M)^{(l+M)}$ for all $l \geq 1$ or (ii) $b_{mj} = 0$ for all $m \geq 1$.

Now, for arbitrary $l \geq 1$, consider the special case

$$(l+l)^{(l+l)} = \sum_{i=1}^{k} (a_{li} \cdot b_{li}).$$

If $j \in \{1, \dots, k\}$ such that either $a_{lj} = 0$ for all $l \geq 1$ or $b_{mj} = 0$ for all $m \geq 1$, then clearly also $(a_{lj} \cdot b_{lj}) = 0$. If j is not like this, we have

$$(a_{lj} \cdot b_{lj}) \leq (l+M)^{(l+M)} \cdot (L+l)^{(L+l)} \leq (l+C)^{2(l+C)}$$

for $C = \max\{L, M\}$. In summary, we have

$$(2l)^{2l} \leq k(l+C)^{2(l+C)}$$

for every $l \geq 1$. Now if l is of the form NC for some $N \in \mathbb{N}$, we have

$$\begin{aligned}
& (2l)^{2l} \leq k(l+C)^{2(l+C)} \\
\Leftrightarrow \quad & (2NC)^{NC} \leq \sqrt{k}((N+1)C)^{(N+1)C} \\
\Leftrightarrow \quad & (2N)^N \leq \sqrt[2C]{k} C (N+1)^{(N+1)} \\
\Leftrightarrow \quad & \frac{2^N}{N+1} \left(\frac{N}{N+1}\right)^N \leq \sqrt[2C]{k} C.
\end{aligned}$$

However, this inequality cannot hold for all $N \in \mathbb{N}$, as

$$\frac{2^N}{N+1} \xrightarrow{N\to\infty} +\infty \quad \text{and} \quad \left(\frac{N}{N+1}\right)^N \xrightarrow{N\to\infty} e^{-1}.$$

□

Necessity of restricting the logic for products

The proof of Theorem 3.26 can also be used to show that no Feferman-Vaught-like theorem holds for products if the first order product quantifier is included in the weighted logic. More precisely, already Theorem 3.21 does not hold over the min-plus and max-plus semirings for the formula $\varphi = \bigotimes x.1$ even if $\bar{\varphi}^{\mathbb{1}}$ and $\bar{\varphi}^{2}$ are allowed to be from $\mathrm{wMSO}(\sigma, K)$.

Theorem 3.29. *Let $K \in \{\mathbb{R}_{\min}, \mathbb{R}_{\max}\}$, $\sigma = (\emptyset, \emptyset)$ be the empty signature, and for $l \in \mathbb{N}_+$ consider the σ-structures $\mathfrak{S}_l = (\{1, \dots, l\}, \emptyset)$. Then for $\varphi = \bigotimes x.1$ there do not exist $n \geq 1$, $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in (\mathrm{wMSO}(\sigma, K))^n$, and $E_\varphi \in \mathrm{Exp}_n(K)$ such that for all $l, m \in \mathbb{N}_+$ we have*

$$[\![\varphi]\!](\mathfrak{S}_l \times \mathfrak{S}_m) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{S}_l), [\![\bar{\varphi}^{2}]\!](\mathfrak{S}_m)).$$

Proof. Like in the proof of Theorem 3.26, for $K = \mathbb{R}_{\min}$ we reduce the problem to a system of equations

$$lm = [\![\varphi]\!](\mathfrak{S}_l \times \mathfrak{S}_m) = \min_{i=1}^{k}(a_{li} + b_{mi}).$$

Employing Ramsey's theorem, we again obtain $l < \lambda < m < \mu$ and $j \in \{1, \dots, k\}$ such that

$$\begin{aligned} lm &= a_{lj} + b_{mj} \\ \lambda m &= a_{\lambda j} + b_{mj} \\ l\mu &= a_{lj} + b_{\mu j} \\ \lambda\mu &= a_{\lambda j} + b_{\mu j}. \end{aligned}$$

Thus, we have

$$\begin{aligned} \lambda\mu &= \lambda m + l\mu - lm \\ &= \lambda\mu - (\lambda - l)(\mu - m) \\ &< \lambda\mu, \end{aligned}$$

which is a contradiction. For $K = \mathbb{R}_{\max}$, the proof is again analogous. □

Proofs of the theorems

We now turn to the proofs of Theorems 3.18 and 3.19. First, we show that we can reduce the proofs to the case where the translation scheme is the identity.

Lemma 3.30. *Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wMSO}(\tau, K)$ with variables from $\mathcal{V}$. Then there exists a formula $\psi \in \mathrm{wMSO}(\sigma, K)$ with $\mathrm{Free}(\psi) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$ such that the following holds. For all finite structures $\mathfrak{A} \in \mathrm{Str}(\sigma)$, or, for all structures $\mathfrak{A} \in \mathrm{Str}(\sigma)$ if K is bicomplete, all $(\mathcal{W}, \mathfrak{A})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A}, \varsigma), \rho) = [\![\psi]\!](\mathfrak{A}, \varsigma \cup \rho).$$

If φ is from $\mathrm{wMSO}^{\bigotimes\text{-res}}(\tau, K)$ or $\mathrm{wFO}^{\bigotimes\text{-free}}(\tau, K)$, then ψ can also be chosen as a formula from $\mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)$ or $\mathrm{wFO}^{\bigotimes\text{-free}}(\sigma, K)$, respectively.

Proof. We proceed by induction. In the sequel we will assume that for formulas φ', φ_1, and φ_2, the lemma holds by induction with the formulas ψ', ψ_1, and ψ_2, respectively.

For $\varphi = \beta \in \mathrm{MSO}(\tau)$, we obtain ψ by applying Lemma 3.12 to β. For $\varphi = \kappa \in K$, we let $\psi = \kappa$. For $\varphi = \varphi_1 \oplus \varphi_2$ or $\varphi = \varphi_1 \otimes \varphi_2$, we define $\psi = \psi_1 \oplus \psi_2$ or $\psi = \psi_1 \otimes \psi_2$, respectively.
For $\varphi = \bigoplus x.\varphi'$, we let $\psi = \bigoplus x.(\psi' \otimes \phi_{\mathcal{U}}(x))$.
For $\varphi = \bigoplus X.\varphi'$, we let $\psi = \bigoplus X.(\psi' \otimes \forall x.(x \in X \to \phi_{\mathcal{U}}(x)))$.
For $\varphi = \bigotimes x.\varphi'$, we let $\psi = \bigotimes x.((\psi' \otimes \phi_{\mathcal{U}}(x)) \oplus \neg\phi_{\mathcal{U}}(x))$.
For $\varphi = \bigotimes X.\varphi'$, we define $\beta = \forall x.(x \in X \to \phi_{\mathcal{U}}(x))$ and let $\psi = \bigotimes X.((\psi' \otimes \beta) \oplus \neg\beta)$.

Note that for the cases of infinite sums and products, we need that $\bigodot_I \mathbb{1} = \mathbb{1}$ and $\bigoplus_I \mathbb{0} = \mathbb{0}$ for every index set I. The first is an axiom of our infinite products, the latter follows from the distributivity of the infinite sum. □

Proof of Theorem 3.18. By Lemma 3.30, it suffices to prove the case $\tau = \sigma$ and $\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma) = \mathfrak{A} \sqcup \mathfrak{B}$. We proceed by induction on the structure of φ. We note that the idea for the case $\varphi = \bigotimes x.\zeta$ for some almost Boolean formula ζ was suggested by Vitaly Perevoshchikov.

■ $\varphi = \beta$ for some $\beta \in \mathrm{MSO}(\sigma)$
We apply Theorem 3.8 to the formula β and obtain $l \geq 1$, tuples of formulas $\bar\beta^{\mathbb{1}}, \bar\beta^{\mathbb{2}} \in \mathrm{MSO}(\sigma)^l$, and an expression $B_\beta \in \mathrm{Exp}_l(\mathbb{B})$ such that

$$(\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta \text{ iff } \langle\!\langle B_\beta \rangle\!\rangle([\![\bar\beta^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar\beta^{\mathbb{2}}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) = 1.$$

We may assume that $B_\beta = B_1 \vee \ldots \vee B_m$ is in normal form with all B_i pure conjunctions. We let $\gamma_i = \mathrm{CON}^{\mathbb{1}}(B_i, \bar\beta^{\mathbb{1}}, \bar\beta^{\mathbb{2}})$ and $\delta_i = \mathrm{CON}^{\mathbb{2}}(B_i, \bar\beta^{\mathbb{1}}, \bar\beta^{\mathbb{2}})$ for $i \in \{1, \ldots, m\}$ (see Construction 3.6). We set $n = 2m$ and define

$$\bar\varphi^{\mathbb{1}} = (\gamma_1, \ldots, \gamma_m, \neg\gamma_1, \ldots, \neg\gamma_m)$$
$$\bar\varphi^{\mathbb{2}} = (\delta_1, \ldots, \delta_m, \neg\delta_1, \ldots, \neg\delta_m).$$

Intuitively, we would now define the expression E_φ as $(x_1 \otimes y_1) \oplus \ldots \oplus (x_m \otimes y_m)$, but this expression is not necessarily evaluated to $\mathbb{1}$ in K if $\gamma_i \wedge \delta_i$ is true for more

than one index i. Instead, we define expressions $E_k \in \mathrm{Exp}_n(K)$ for $k \in \{1, \dots, m\}$ inductively by $E_1 = x_1 \otimes y_1$ and

$$E_k = (E_{k-1} \otimes ((x_{k+m} \otimes y_k) \oplus y_{k+m})) \oplus (x_k \otimes y_k)$$

for $k \geq 2$ and set $E_\varphi = E_m$. The expression E_k is evaluated to $\mathbb{1}$ if $\gamma_k \wedge \delta_k$ holds, and otherwise, if either γ_k or δ_k does not hold, it is evaluated to E_{k-1}. We show by induction that for all $k \in \{1, \dots, m\}$ we have

$$\langle\!\langle E_k \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) =$$
$$\begin{cases} \mathbb{1} & \text{if } \langle\!\langle B_i \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 1 \text{ for some } i \in \{1, \dots, k\} \\ \mathbb{0} & \text{otherwise.} \end{cases}$$

Let $\bar{\kappa} = [\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}})$ and $\bar{\lambda} = [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})$. For $k = 1$ we have, due to the fact that $\bar{\kappa}, \bar{\lambda} \in \{\mathbb{0}, \mathbb{1}\}^{2m}$,

$$\begin{aligned} &\langle\!\langle x_1 \otimes y_1 \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{1} \\ \Leftrightarrow\ & \kappa_1 = \mathbb{1} \text{ and } \lambda_1 = \mathbb{1} \\ \Leftrightarrow\ & (\mathfrak{A}, \rho\restriction_{\mathfrak{A}}) \models \gamma_1 \text{ and } (\mathfrak{B}, \rho\restriction_{\mathfrak{A}}) \models \delta_1 \\ \Leftrightarrow\ & \langle\!\langle B_1 \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 1 \end{aligned}$$

and $\langle\!\langle E_1 \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{0}$ otherwise. For $k > 1$ and $\langle\!\langle B_i \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 0$ for all $i \in \{1, \dots, k\}$, we have $\langle\!\langle E_{k-1} \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{0}$ by induction and at least one of κ_k, λ_k is $\mathbb{0}$. It is easy to see that in this case $\langle\!\langle E_k \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{0}$. Otherwise either $\langle\!\langle E_{k-1} \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{1}$ by induction or $\kappa_k = \lambda_k = \mathbb{1}$. Taking into account that the values for x_k, x_{k+m} and y_k, y_{k+m} are always "dual", a simple case distinction shows that in this case $\langle\!\langle E_k \rangle\!\rangle(\bar{\kappa}, \bar{\lambda}) = \mathbb{1}$. In conclusion, we have $[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \in \{\mathbb{0}, \mathbb{1}\}$ and

$$\begin{aligned} &[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) = \mathbb{1} \\ \Leftrightarrow\ & (\mathfrak{A} \sqcup \mathfrak{B}, \rho) \models \beta \\ \Leftrightarrow\ & \langle\!\langle B_\beta \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 1 \\ \Leftrightarrow\ & \langle\!\langle B_i \rangle\!\rangle([\![\bar{\beta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\beta}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = 1 \text{ for some } i \in \{1, \dots, m\} \\ \Leftrightarrow\ & \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})) = \mathbb{1}, \end{aligned}$$

hence $[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}}))$.

■ $\varphi = \kappa$ for some $\kappa \in K$
We let $n = 1$, $\varphi_1^{\mathbb{1}} = \varphi_1^{2} = \kappa$ and $E_\varphi = x_1$.

■ $\varphi = \zeta \oplus \eta$
We assume the theorem is true for ζ with $\bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2} \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)^l$ and $E_\zeta \in \mathrm{Exp}_l(K)$, and for η with $\bar{\eta}^{\mathbb{1}}, \bar{\eta}^{2} \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)^m$ and $E_\eta \in \mathrm{Exp}_m(K)$. We let $\bar{\varphi}^{\mathbb{1}} = (\zeta_1^{\mathbb{1}}, \dots, \zeta_l^{\mathbb{1}}, \eta_1^{\mathbb{1}}, \dots, \eta_m^{\mathbb{1}})$, $\bar{\varphi}^{2} = (\zeta_1^{2}, \dots, \zeta_l^{2}, \eta_1^{2}, \dots, \eta_m^{2})$, and $E_\varphi = E_\zeta \oplus E'_\eta$, where E'_η is obtained from E_η by replacing every variable x_i by x_{i+l} and every variable

y_i by y_{i+l}. We have

$$\begin{aligned}
&[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \\
&= [\![\zeta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \oplus [\![\eta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \\
&= \langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) \oplus \langle\!\langle E_\eta \rangle\!\rangle([\![\bar{\eta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\eta}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) \\
&= \langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) \oplus \langle\!\langle E'_\eta \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})) \\
&= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})).
\end{aligned}$$

■ $\varphi = \zeta \otimes \eta$
The proof is the same as for the previous case, only that here we define $E_\varphi = E_\zeta \otimes E'_\eta$.

■ $\varphi = \bigoplus x.\zeta$
We assume the theorem is true for ζ with $\bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2} \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)^l$ and $E_\zeta \in \mathrm{Exp}_l(K)$. We may assume that $E_\zeta = E_1 \oplus \ldots \oplus E_m$ is in normal form with all E_i pure products and that x does no occur as a bound variable in any of the $\zeta_i^{\mathbb{1}}$ or ζ_i^{2}. We let $\xi_i = \mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$ and $\theta_i = \mathrm{PRD}^{2}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$. We set $n = 2m$ and define

$$\begin{aligned}
\bar{\varphi}^{\mathbb{1}} &= (\textstyle\bigoplus x.\xi_1, \ldots, \bigoplus x.\xi_m, \xi_1^{-x}, \ldots, \xi_m^{-x}) \\
\bar{\varphi}^{2} &= (\textstyle\bigoplus x.\theta_1, \ldots, \bigoplus x.\theta_m, \theta_1^{-x}, \ldots, \theta_m^{-x}) \\
E_\varphi &= \textstyle\bigoplus_{i=1}^{m}((x_i \otimes y_{m+i}) \oplus (x_{m+i} \otimes y_i)).
\end{aligned}$$

Then we have

$$\begin{aligned}
&[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \\
&= \bigoplus_{c \in A \sqcup B} [\![\zeta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \\
&= \bigoplus_{c \in A \sqcup B} \langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho[x \to c]{\restriction_{\mathfrak{A}}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho[x \to c]{\restriction_{\mathfrak{B}}})) \\
&= \bigoplus_{c \in A \sqcup B} \bigoplus_{i=1}^{m} \langle\!\langle E_i \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho[x \to c]{\restriction_{\mathfrak{A}}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho[x \to c]{\restriction_{\mathfrak{B}}})) \\
&= \bigoplus_{c \in A \sqcup B} \bigoplus_{i=1}^{m} [\![\xi_i]\!](\mathfrak{A}, \rho[x \to c]{\restriction_{\mathfrak{A}}}) \odot [\![\theta_i]\!](\mathfrak{B}, \rho[x \to c]{\restriction_{\mathfrak{B}}}) \\
&= \bigoplus_{a \in A} \bigoplus_{i=1}^{m} [\![\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[x \to a]) \odot [\![\theta_i^{-x}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \\
&\quad \oplus \bigoplus_{b \in B} \bigoplus_{i=1}^{m} [\![\xi_i^{-x}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot [\![\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[x \to b]) \\
&= \bigoplus_{i=1}^{m} \left(\bigoplus_{a \in A} [\![\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[x \to a]) \right) \odot [\![\theta_i^{-x}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \\
&\qquad \oplus [\![\xi_i^{-x}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot \left(\bigoplus_{b \in B} [\![\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[x \to b]) \right)
\end{aligned}$$

$$= \bigoplus_{i=1}^{m} [\![\bigoplus x.\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot [\![\theta_i^{-x}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \oplus [\![\xi_i^{-x}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot [\![\bigoplus x.\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})$$
$$= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})).$$

■ $\varphi = \bigoplus X.\zeta$
As for the first order sum quantifier, we assume that the theorem is true for ζ with $\bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2} \in \mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)^l$ and $E_\zeta \in \mathrm{Exp}_l(K)$ such that $E_\zeta = E_1 \oplus \ldots \oplus E_m$ is in normal form with all E_i pure products. We let $\xi_i = \mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$ and $\theta_i = \mathrm{PRD}^{2}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$. We set $n = m$ and define

$$\bar{\varphi}^{\mathbb{1}} = (\bigoplus X.\xi_1, \ldots, \bigoplus X.\xi_m)$$
$$\bar{\varphi}^{2} = (\bigoplus X.\theta_1, \ldots, \bigoplus X.\theta_m)$$
$$E_\varphi = \textstyle\bigoplus_{i=1}^{m} (x_i \otimes y_i).$$

Then we have

$$\begin{aligned}
&[\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \\
&= \bigoplus_{I \subseteq A \sqcup B} [\![\zeta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[X \to I]) \\
&= \bigoplus_{I \subseteq A \sqcup B} \langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho[X \to I]{\restriction_{\mathfrak{A}}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho[X \to I]{\restriction_{\mathfrak{B}}})) \\
&= \bigoplus_{I \subseteq A \sqcup B} \bigoplus_{i=1}^{m} [\![\xi_i]\!](\mathfrak{A}, \rho[X \to I]{\restriction_{\mathfrak{A}}}) \odot [\![\theta_i]\!](\mathfrak{B}, \rho[X \to I]{\restriction_{\mathfrak{B}}}) \\
&= \bigoplus_{i=1}^{m} \bigoplus_{I \subseteq A} \bigoplus_{J \subseteq B} [\![\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[X \to I]) \odot [\![\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[X \to J]) \\
&= \bigoplus_{i=1}^{m} \left(\bigoplus_{I \subseteq A} [\![\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}[X \to I]) \right) \odot \left(\bigoplus_{J \subseteq B} [\![\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}[X \to J]) \right) \\
&= \bigoplus_{i=1}^{m} [\![\bigoplus X.\xi_i]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot [\![\bigoplus X.\theta_i]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \\
&= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})).
\end{aligned}$$

■ $\varphi = \bigotimes x.\zeta$ with $\zeta \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$ almost Boolean
Using the laws of distributivity in K and the fact that for two MSO formulas $\alpha, \beta \in \mathrm{MSO}(\sigma)$ we have $[\![\alpha \otimes \beta]\!] = [\![\alpha \wedge \beta]\!]$, we may assume that $\zeta = (\kappa_1 \otimes \beta_1) \oplus \ldots \oplus (\kappa_l \otimes \beta_l)$ for some $l \geq 1$, $\kappa_i \in K$, and $\beta_i \in \mathrm{MSO}(\sigma)$. First, we will show that we may even assume that $\beta_1, \ldots, \beta_l$ form a partition, i.e., that for every $(\mathcal{V}, \mathfrak{A} \sqcup \mathfrak{B})$-assignment ρ' there is exactly one $i \in \{1, \ldots, l\}$ with $(\mathfrak{A} \sqcup \mathfrak{B}, \rho') \models \beta_i$.

For this, let $\Omega = \{\beta_1, \neg\beta_1\} \times \ldots \times \{\beta_l, \neg\beta_l\}$. For every $\bar{\omega} = (\omega_1, \ldots, \omega_l) \in \Omega$ we define a formula $\alpha_{\bar{\omega}}$ and $\kappa_{\bar{\omega}} \in K$ as follows.

$$\alpha_{\bar{\omega}} = \bigwedge_{i=1}^{l} \omega_i \qquad\qquad \kappa_{\bar{\omega}} = \bigoplus_{\substack{1 \leq i \leq l \\ \omega_i = \beta_i}} \kappa_i$$

The empty sum is $\mathbb{0}$ by convention. It is clear that for every $(\mathcal{V}, \mathfrak{A} \sqcup \mathfrak{B})$-assignment ρ' there exists a unique $\bar{\omega} \in \Omega$ with $(\mathfrak{A} \sqcup \mathfrak{B}, \rho') \models \alpha_{\bar{\omega}}$. Moreover, for $i \in \{1, \dots, l\}$ we have $(\mathfrak{A} \sqcup \mathfrak{B}, \rho') \models \beta_i$ if and only if $(\mathfrak{A} \sqcup \mathfrak{B}, \rho') \models \alpha_{\bar{\omega}}$ for some $\bar{\omega} \in \Omega$ with $\omega_i = \beta_i$, and in this case $\bar{\omega}$ is unique. We therefore have

$$\begin{aligned}
[\![\zeta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho') &= \bigoplus_{i=1}^{l} \kappa_i \odot [\![\beta_i]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho') \\
&= \bigoplus_{i=1}^{l} \kappa_i \odot \bigoplus_{\substack{\bar{\omega} \in \Omega \\ \omega_i = \beta_i}} [\![\alpha_{\bar{\omega}}]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho') \\
&= \bigoplus_{i=1}^{l} \bigoplus_{\substack{\bar{\omega} \in \Omega \\ \omega_i = \beta_i}} \kappa_i \odot [\![\alpha_{\bar{\omega}}]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho') \\
&= \bigoplus_{\bar{\omega} \in \Omega} \left(\bigoplus_{\substack{1 \le i \le l \\ \omega_i = \beta_i}} \kappa_i \right) \odot [\![\alpha_{\bar{\omega}}]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho') \\
&= \bigoplus_{\bar{\omega} \in \Omega} \kappa_{\bar{\omega}} \odot [\![\alpha_{\bar{\omega}}]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho').
\end{aligned}$$

Thus, $[\![\zeta]\!] = [\![\bigoplus_{\bar{\omega} \in \Omega} \kappa_{\bar{\omega}} \otimes \alpha_{\bar{\omega}}]\!]$ and the family $(\alpha_{\bar{\omega}})_{\bar{\omega} \in \Omega}$ forms a partition in the above sense. In the following, we simply assume that $\zeta = (\kappa_1 \otimes \beta_1) \oplus \ldots \oplus (\kappa_l \otimes \beta_l)$ and that $\beta_1, \dots, \beta_l$ form a partition.

For every $i \in \{1, \dots, l\}$, let $X_i \in \mathcal{V}$ be a second order variable not occurring in ζ. We define the abbreviation

$$((x \in X_i) \rhd \kappa_i) = ((x \in X_i) \otimes \kappa_i) \oplus \neg(x \in X_i).$$

We write all of the X_i into a tuple $\bar{X}$ and for sets $I_i \subseteq A \sqcup B$ ($i \in \{1, \dots, l\}$), we let $\bar{I}$ be the corresponding tuple of sets. Then for $c \in A \sqcup B$ and sets $I_i \subseteq A \sqcup B$ we have

$$[\![(x \in X_i) \rhd \kappa_i]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[\bar{X} \to \bar{I}, x \to c]) = \begin{cases} \kappa_i & \text{if } c \in I_i \\ \mathbb{1} & \text{otherwise.} \end{cases}$$

Now consider the formula

$$\left(\bigwedge_{i=1}^{l} \forall x.(x \in X_i \leftrightarrow \beta_i) \right) \otimes \bigotimes_{i=1}^{l} \bigotimes x.((x \in X_i) \rhd \kappa_i).$$

For sets $I_i \subseteq A \sqcup B$ ($i \in \{1, \dots, l\}$) we have

$$[\![\bigwedge_{i=1}^{l} \forall x.(x \in X_i \leftrightarrow \beta_i)]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[\bar{X} \to \bar{I}])$$

$$= \begin{cases} \mathbb{1} & \text{if for all } c \in A \sqcup B \text{ and all } i \in \{1, \dots, l\} \colon c \in I_i \text{ iff } (\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \models \beta_i \\ \mathbb{0} & \text{otherwise.} \end{cases}$$

Hence, the above is evaluated to $\mathbb{1}$ if and only if $I_i = \{c \in A \sqcup B \mid (\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \models \beta_i\}$ for all $i \in \{1, \dots, l\}$. In this case, the family $(I_i)_{1 \leq i \leq l}$ is a partition of $A \sqcup B$, since the family $(\beta_i)_{1 \leq i \leq l}$ forms a partition. Therefore, in this case we have

$$\begin{aligned}
&[\![\bigotimes_{i=1}^{l} \bigotimes x.((x \in X_i) \rhd \kappa_i)]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[\bar{X} \to \bar{I}]) \\
&= \bigodot_{i=1}^{l} \bigodot_{c \in I_i} \kappa_i \\
&= \bigodot_{c \in A \sqcup B} \bigoplus_{i=1}^{l} \kappa_i \odot [\![\beta_i]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \\
&= \bigodot_{c \in A \sqcup B} [\![\zeta]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho[x \to c]) \\
&= [\![\varphi]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho).
\end{aligned}$$

In conclusion, we have

$$[\![\varphi]\!] = [\![\bigoplus X_1 . \bigoplus X_2 \dots \bigoplus X_l . \left(\bigwedge_{i=1}^{l} \forall x.(x \in X_i \leftrightarrow \beta_i) \right) \otimes \bigotimes_{i=1}^{l} \bigotimes x.((x \in X_i) \rhd \kappa_i)]\!].$$

Therefore, it suffices to show this case of the induction for formulas of the form

$$\varphi = \bigotimes x.((x \in X) \rhd \kappa).$$

We let $n = 1$ and define $\bar{\varphi}^{\mathbb{1}} = \bar{\varphi}^{2} = (\bigotimes x.((x \in X) \rhd \kappa))$ and $E_\varphi = x_1 \otimes y_1$. Then we have

$$\begin{aligned}
&[\![\bigotimes x.((x \in X) \rhd \kappa)]\!](\mathfrak{A} \sqcup \mathfrak{B}, \rho) \\
&= [\![\bigotimes x.((x \in X) \rhd \kappa)]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}) \odot [\![\bigotimes x.((x \in X) \rhd \kappa)]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}}) \\
&= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho{\restriction_{\mathfrak{A}}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho{\restriction_{\mathfrak{B}}})).
\end{aligned}$$

□

Proof of Theorem 3.19. Again we proceed by induction and assume that $\tau = \sigma$ and $\Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma) = \mathfrak{A} \times \mathfrak{B}$. The proofs for the cases $\varphi = \beta$, $\varphi = \kappa$, $\varphi = \zeta \oplus \eta$, and $\varphi = \zeta \otimes \eta$ are identical to the ones used in the proof of Theorem 3.18 for the corresponding cases.

For the case $\varphi = \bigoplus x.\zeta$ we proceed as for the case $\varphi = \bigoplus X.\zeta$ in the proof of Theorem 3.18 as follows. We assume that the theorem is true for ζ with $\bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2} \in \mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)^l$ and $E_\zeta \in \mathrm{Exp}_l(K)$ such that $E_\zeta = E_1 \oplus \dots \oplus E_m$ is in normal form with all E_i pure products. We let $\xi_i = \mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$ and $\theta_i = \mathrm{PRD}^{2}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$. We set $n = m$ and define

$$\begin{aligned}
\bar{\varphi}^{\mathbb{1}} &= (\bigoplus x.\xi_1, \dots, \bigoplus x.\xi_m) \\
\bar{\varphi}^{2} &= (\bigoplus x.\theta_1, \dots, \bigoplus x.\theta_m) \\
E_\varphi &= \bigoplus_{i=1}^{m} (x_i \otimes y_i).
\end{aligned}$$

Then we have

$$
\begin{aligned}
&[\![\varphi]\!](\mathfrak{A} \times \mathfrak{B}, \rho) \\
&= \bigoplus_{c \in A \times B} [\![\zeta]\!](\mathfrak{A} \times \mathfrak{B}, \rho[x \to c]) \\
&= \bigoplus_{c \in A \times B} \langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho[x \to c]\restriction_{\mathfrak{A}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho[x \to c]\restriction_{\mathfrak{B}})) \\
&= \bigoplus_{c \in A \times B} \bigoplus_{i=1}^{m} [\![\xi_i]\!](\mathfrak{A}, \rho[x \to c]\restriction_{\mathfrak{A}}) \odot [\![\theta_i]\!](\mathfrak{B}, \rho[x \to c]\restriction_{\mathfrak{B}}) \\
&= \bigoplus_{i=1}^{m} \bigoplus_{a \in A} \bigoplus_{b \in B} [\![\xi_i]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}[x \to a]) \odot [\![\theta_i]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}}[x \to b]) \\
&= \bigoplus_{i=1}^{m} \left(\bigoplus_{a \in A} [\![\xi_i]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}[x \to a]) \right) \odot \left(\bigoplus_{b \in B} [\![\theta_i]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}}[x \to b]) \right) \\
&= \bigoplus_{i=1}^{m} [\![\textstyle\bigoplus x.\xi_i]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}) \odot [\![\textstyle\bigoplus x.\theta_i]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}}) \\
&= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho\restriction_{\mathfrak{B}})).
\end{aligned}
$$

□

This concludes the proofs of Theorems 3.18 and 3.19.

We note here that Theorems 3.18 and 3.19, which consider disjoint unions and products of only two structures, can easily be extended to disjoint unions and products of finitely many structures. This can be shown either by modifying the proofs of this section directly, or by using an induction on the number of structures m, where the base case $m = 2$ is given by Theorems 3.18 and 3.19. Since both proof methods are technical but not too difficult to implement, we omit the formal proof.

3.5 Extensions

In this section, we want to consider several extensions of Theorems 3.18 and 3.19. More precisely, we show two conditions on the semiring under which we can drop the restrictions on the product quantifiers, and we show how to combine Theorems 3.18 and 3.19 with *transductions*.

De Morgan algebras

In this section, we consider the special case where our semiring can be extended by a unary operation $\neg$ to form a *De Morgan algebra* $(L, \vee, \wedge, \neg, 0, 1)$. A tuple $(L, \vee, \wedge, \neg, 0, 1)$ is called a De Morgan algebra if $(L, \vee, \wedge, 0, 1)$ is a bounded distributive lattice and $\neg\colon L \to L$ is an involution satisfying De Morgan's laws, i.e., we have $\neg(x \vee y) = \neg x \wedge \neg y$, $\neg(x \wedge y) = \neg x \vee \neg y$, and $\neg\neg x = x$ for all $x, y \in L$. If $\leq$ is the induced order of the lattice $(L, \vee, \wedge, 0, 1)$, it follows that $\neg\colon (L, \leq) \to (L, \leq)$ is an order-antiisomorphism. In particular, $\neg 0 = 1$ and $\neg 1 = 0$. A De Morgan

algebra is called *complete* if $(L, \vee, \wedge, 0, 1)$ is a complete lattice. Since $\neg$ is an order-antiisomorphism, it follows that the equalities $\neg \bigwedge_{x\in X} x = \bigvee_{x\in X} \neg x$ and $\neg \bigvee_{x\in X} x = \bigwedge_{x\in X} \neg x$ hold for every subset $X \subseteq L$ of a complete lattice L.

Example 3.31. Examples of De Morgan algebras include

- all Boolean algebras, in particular, the two element Boolean algebra $\mathbb{B}$,
- *Kleene* or *Priest logic* $(\{F, I, T\}, \vee, \wedge, \neg, F, T)$ where $F \leq I \leq T$ describes the lattice and $\neg I = I$, $\neg F = T$ the negation,
- *Belnap* or *Dunn logic* $(\{F, B, N, T\}, \vee, \wedge, \neg, F, T)$ where $F \leq B \leq T$, $F \leq N \leq T$, and B and N are incomparable, and the negation is given by $\neg B = B$, $\neg N = N$, $\neg F = T$, and
- the *Łukasiewicz logics*, for example $L_\infty = ([0, 1], \max, \min, \neg, 0, 1)$ where $\neg x = 1 - x$.

Whenever we are dealing with a De Morgan algebra, we can include the operator $\neg$ into our weighted logic.

Definition 3.32 (De Morgan-extension)**.** Let σ be a signature. We define the *De Morgan-extensions* of our weighted first order and monadic second order logics through the grammars

$$\varphi ::= \beta \mid \kappa \mid \neg\varphi \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi,$$

where $\beta \in \mathrm{FO}(\sigma)$ is a first order formula, $\kappa \in L$, and x is a first order variable, and

$$\varphi ::= \beta \mid \kappa \mid \neg\varphi \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus x.\varphi \mid \bigotimes x.\varphi \mid \bigoplus X.\varphi \mid \bigotimes X.\varphi,$$

where $\beta \in \mathrm{MSO}(\sigma)$ is a monadic second order formula, $\kappa \in L$, x is a first order variable, and X is a second order variable, respectively. The semantics of $\neg\varphi$ is defined by $[\![\neg\varphi]\!](\mathfrak{A}, \rho) = \neg[\![\varphi]\!](\mathfrak{A}, \rho)$ for a σ-structure $\mathfrak{A}$ and a variable assignment ρ. By $\mathrm{wFO}^\neg(\sigma, L)$ and $\mathrm{wMSO}^\neg(\sigma, L)$, we denote the sets of all such formulas, respectively. Weighted logics for *words* over bounded lattices were also considered in [35], where the authors showed that Kleene-type and Büchi-like results hold for these logics.

Since L is a De Morgan algebra, it is easy to see that for every formula $\varphi \in \mathrm{wMSO}^\neg(\sigma, L)$ the formulas $\bigotimes x.\varphi$ and $\neg\bigoplus x.\neg\varphi$ are semantically equivalent. The same holds true for the formulas $\bigotimes X.\varphi$ and $\neg\bigoplus X.\neg\varphi$. Therefore, in this scenario we do not need any restriction to formulate weighted Feferman-Vaught decomposition theorems. Let τ, $\mathcal{W}$, and $\mathcal{Z}$ be as in Section 3.3.

Theorem 3.33. *Let L be a De Morgan algebra, $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T\in \mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wMSO}^\neg(\tau, L)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{\mathbb{2}} \in \mathrm{wMSO}^\neg(\sigma, L)^n$ with $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^{\mathbb{2}}) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(L)$ such that the following holds. For all finite structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, or, for all structures*

$\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ *if* L *is complete, all* $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$*-assignments* ς*, and all* $(\mathcal{V}, \Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma))$*-assignments* ρ *we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)|_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)|_{\mathfrak{B}})).$$

Theorem 3.34. *Let* L *be a De Morgan algebra,* $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T \in \mathrm{Rel}_\tau})$ *be a* σ*-*τ*-translation scheme over* $\mathcal{W}$ *and* $\mathcal{Z}$*,* $\mathcal{V}$ *be a set of first and second order variables such that* $\mathcal{V}$*,* $\mathcal{W}$*, and* $\mathcal{Z}$ *are pairwise disjoint, and* $\varphi \in \mathrm{wFO}^\neg(\tau, L)$ *with variables from* $\mathcal{V}$*. Then there exist* $n \geq 1$*, tuples of formulas* $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in \mathrm{wFO}^\neg(\sigma, L)^n$ *with* $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^{2}) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$*, and an expression* $E_\varphi \in \mathrm{Exp}_n(L)$ *such that the following holds. For all finite structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$*, or, for all structures* $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ *if* L *is complete, all* $(\mathcal{W}, \mathfrak{A} \times \mathfrak{B})$*-assignments* ς*, and all* $(\mathcal{V}, \Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma))$*-assignments* ρ *we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)|_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)|_{\mathfrak{B}})).$$

Proof. We proceed as in the proofs of Theorems 3.18 and 3.19. To see that we can assume the translation scheme to be trivial, note that the inductive proof of Lemma 3.30 can easily be extended to $\mathrm{wMSO}^\neg(\sigma, L)$: if $\varphi = \neg\varphi'$ and the lemma is true for φ' with the formula ψ', then we can choose $\psi = \neg\psi'$.

Using the inductive steps of the proofs for Theorems 3.18 and 3.19 and the above rewriting of product quantifiers into sum quantifiers through a double weighted negation, we see that it only remains to show the inductive step for the weighted negation $\varphi = \neg\zeta$ as follows.

We proceed as in the proof for the Boolean case. Also, the proofs for the disjoint union and the product are the same, so in the following let $\mathfrak{C} = \mathfrak{A} \sqcup \mathfrak{B}$ or $\mathfrak{C} = \mathfrak{A} \times \mathfrak{B}$. We assume the theorem is true for ζ with $E_\zeta \in \mathrm{Exp}_l(L)$ and $\bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2}$ from $\mathrm{wFO}^\neg(\sigma, L)^l$ or from $\mathrm{wMSO}^\neg(\sigma, L)^l$. We may assume that $E_\zeta = E_1 \oplus \ldots \oplus E_m$ is in normal form with all E_i pure products. We let $\xi_i = \mathrm{PRD}^{\mathbb{1}}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$ and $\theta_i = \mathrm{PRD}^{2}(E_i, \bar{\zeta}^{\mathbb{1}}, \bar{\zeta}^{2})$ and define

$$\begin{aligned}
\bar{\varphi}^{\mathbb{1}} &= (\neg\xi_1, \ldots, \neg\xi_m) \\
\bar{\varphi}^{2} &= (\neg\zeta_1, \ldots, \neg\zeta_m) \\
E_\varphi &= \bigwedge_{i=1}^{m} (x_i \vee y_i).
\end{aligned}$$

Then we have

$$\begin{aligned}
[\![\varphi]\!](\mathfrak{C}, \rho) &= \neg\langle\!\langle E_\zeta \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho|_{\mathfrak{A}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho|_{\mathfrak{B}})) \\
&= \neg \bigvee_{i=1}^{m} \langle\!\langle E_i \rangle\!\rangle([\![\bar{\zeta}^{\mathbb{1}}]\!](\mathfrak{A}, \rho|_{\mathfrak{A}}), [\![\bar{\zeta}^{2}]\!](\mathfrak{B}, \rho|_{\mathfrak{B}})) \\
&= \neg \bigvee_{i=1}^{m} [\![\xi_i]\!](\mathfrak{A}, \rho|_{\mathfrak{A}}) \wedge [\![\theta_i]\!](\mathfrak{B}, \rho|_{\mathfrak{B}}) \\
&= \bigwedge_{i=1}^{m} [\![\neg\xi_i]\!](\mathfrak{A}, \rho|_{\mathfrak{A}}) \vee [\![\neg\theta_i]\!](\mathfrak{B}, \rho|_{\mathfrak{B}}) \\
&= \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, \rho|_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, \rho|_{\mathfrak{B}})).
\end{aligned}$$

□

Weakly biaperiodic semirings

In this section, we show that Theorems 3.18 and 3.19 hold true without the need for any restriction whenever our weights are taken from a *weakly biaperiodic* commutative semiring. A monoid is called *weakly aperiodic* if for every element x there exists a positive integer n such that $x^n = x^{n+1}$. We call a semiring $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ weakly biaperiodic if both its additive monoid $(K, \oplus, \mathbb{0})$ and its multiplicative monoid $(K, \odot, \mathbb{1})$ are weakly aperiodic. Weighted logics for *words* over weakly biaperiodic semirings were also considered in [27, 35].

Example 3.35. Examples of weakly biaperiodic semirings include

- Every De Morgan algebra, in particular, all semirings from Example 3.31,
- the Łukasiewicz semiring $([0,1], \max, \otimes, 0, 1)$ where $x \otimes y = \max\{0, x + y - 1\}$,
- the truncated min-plus semiring $([0,d], \min, +_d, d, 0)$ for a real number $d > 0$, where $x +_d y = \min\{d, x + y\}$.

For weakly biaperiodic semirings, we can show that every quantifier, when quantifying over an almost Boolean formula, again models an almost Boolean formula. The proof for this employs explicit case distinctions to compute the outcomes of the quantifiers. By induction, it follows that for weakly biaperiodic semirings, every wMSO formula is semantically equivalent to an almost Boolean formula, i.e., a formula containing no weighted quantifiers. We thus have the following lemma.

Lemma 3.36. *Let K be a weakly biaperiodic commutative semiring and σ a signature. Then for every formula $\varphi \in \mathrm{wMSO}(\sigma, K)$, there exists a formula $\psi \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$ with $[\![\varphi]\!] = [\![\psi]\!]$.*

Proof. We proceed by induction. For the cases $\varphi = \beta \in \mathrm{MSO}(\sigma, K)$, $\varphi = \kappa \in K$, $\varphi = \psi_1 \oplus \psi_2$, and $\varphi = \psi_1 \otimes \psi_2$ with $\psi_1, \psi_2 \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$, the statement is clear.

For the cases $\varphi = \bigoplus x.\varphi'$, $\varphi = \bigoplus X.\varphi'$, $\varphi = \bigotimes x.\varphi'$, and $\varphi = \bigotimes X.\varphi'$ with $\varphi' \in \mathrm{wMSO}(\sigma, K)$, we proceed as follows. By induction, we assume that there exists an almost Boolean formula $\psi' \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$ such that $[\![\varphi']\!] = [\![\psi']\!]$. We assume that ψ' is of the form $\psi' = (\kappa_1 \otimes \beta_1) \oplus \ldots \oplus (\kappa_l \otimes \beta_l)$, where $\beta_1, \ldots, \beta_l$ form a partition like in the proof of Theorem 3.18. By the assumption that K is weakly biaperiodic, there exists for every $i \in \{1, \ldots, l\}$ a number $n_i \in \mathbb{N}_+$ such that $\bigoplus_{j=1}^{n_i} \kappa_i = \bigoplus_{j=1}^{n_i+1} \kappa_i$. We let $N_1 = \max_{i=1}^{l} n_i$. Likewise, there exist $n_i \in \mathbb{N}_+$ such that $\bigodot_{j=1}^{n_i} \kappa_i = \bigodot_{j=1}^{n_i+1} \kappa_i$ for every $i \in \{1, \ldots, l\}$. We let $N_2 = \max_{i=1}^{l} n_i$. Then with $N = \max\{N_1, N_2\}$ we have $\bigoplus_{j=1}^{N} \kappa_i = \bigoplus_{j=1}^{N+1} \kappa_i$ and $\bigodot_{j=1}^{N} \kappa_i = \bigodot_{j=1}^{N+1} \kappa_i$ for all $i \in \{1, \ldots, l\}$. Furthermore, we define abbreviations as follows. For first order variables y_1 and y_2 and second order variables Y_1 and Y_2, we let

$$(y_1 = y_2) = \forall Z.(y_1 \in Z \leftrightarrow y_2 \in Z)$$
$$(Y_1 = Y_2) = \forall z.(z \in Y_1 \leftrightarrow z \in Y_2).$$

Now let $\beta \in \mathrm{MSO}(\sigma, K)$ be a monadic second order formula. For a first order variable y, we denote by $\beta(y)$ the formula which results from β by renaming every free occurrence of the first order variable x to y. For a second order variable Y, we denote by $\beta(Y)$ the formula which results from β by renaming every free occurrence of the second order variable X to Y. Then for $m \in \mathbb{N}_+$ and $\mathcal{X} \in \{x, X\}$, we define the abbreviations

$$\exists^{\geq m}\mathcal{X}.\beta = \exists\mathcal{X}_1 \dots \exists\mathcal{X}_m.\Big(\bigwedge_{i=1}^{m} \beta(\mathcal{X}_i) \wedge \bigwedge_{i\neq j} \neg(\mathcal{X}_i = \mathcal{X}_j)\Big)$$
$$\exists^{m}\mathcal{X}.\beta = \exists^{\geq m}\mathcal{X}.\beta \wedge \neg(\exists^{\geq m+1}\mathcal{X}.\beta)$$
$$\bar{\exists}^{m}\mathcal{X}.\beta = \begin{cases} \exists^{m}\mathcal{X}.\beta & \text{if } m < N \\ \exists^{\geq N}\mathcal{X}.\beta & \text{if } m \geq N, \end{cases}$$

where $\mathcal{X}_1, \dots, \mathcal{X}_m \notin \mathrm{Free}(\psi')$ are first order variables if $\mathcal{X} = x$, and they are second order variables if $\mathcal{X} = X$. For every $\bar{v} \in \{0, \dots, N\}^l$, we define the constants

$$\kappa_{\bar{v}} = \bigoplus_{i=1}^{l} \bigoplus_{j=1}^{v_i} \kappa_i \qquad\qquad \lambda_{\bar{v}} = \bigodot_{i=1}^{l} \bigodot_{j=1}^{v_i} \kappa_i.$$

Again, the empty sum is defined as $\mathbb{0}$ and the empty product as $\mathbb{1}$. Then for the case $\varphi = \bigoplus\mathcal{X}.\varphi'$ with $\mathcal{X} \in \{x, X\}$, we define the formula

$$\psi = \bigoplus_{\bar{v}\in\{0,\dots,N\}^l} \kappa_{\bar{v}} \otimes \bigwedge_{i=1}^{l} \bar{\exists}^{v_i}\mathcal{X}.\beta_i.$$

By the definition of $\kappa_{\bar{v}}$ and the choice of N, we have $[\![\varphi]\!] = [\![\psi]\!]$ and ψ is almost Boolean. For the case $\varphi = \bigotimes\mathcal{X}.\varphi'$ with $\mathcal{X} \in \{x, X\}$, we define

$$\psi = \bigoplus_{\bar{v}\in\{0,\dots,N\}^l} \lambda_{\bar{v}} \otimes \bigwedge_{i=1}^{l} \bar{\exists}^{v_i}\mathcal{X}.\beta_i.$$

Again, we have $[\![\varphi]\!] = [\![\psi]\!]$ and ψ is almost Boolean. □

We let τ, $\mathcal{W}$, and $\mathcal{Z}$ be as in Section 3.3. Then we have the following theorems.

Theorem 3.37. *Let K be a weakly biaperiodic commutative semiring. Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T\in\mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wMSO}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^{2} \in \mathrm{wMSO}(\sigma, K)^n$ with $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^{2}) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$ such that the following holds. For all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, all $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})).$$

Theorem 3.38. *Let K be a weakly biaperiodic commutative semiring. Let $\Phi = (\phi_{\mathcal{U}}, (\phi_T)_{T\in\mathrm{Rel}_\tau})$ be a σ-τ-translation scheme over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathrm{wFO}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar{\varphi}^{\mathbb{1}}, \bar{\varphi}^2 \in \mathrm{wFO}(\sigma, K)^n$ with $\mathrm{Free}(\bar{\varphi}^{\mathbb{1}}) \cup \mathrm{Free}(\bar{\varphi}^2) \subseteq \mathrm{Free}(\varphi) \cup \mathrm{Free}(\Phi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$ such that the following holds. For all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, all $(\mathcal{W}, \mathfrak{A} \times \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Phi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar{\varphi}^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho)\restriction_{\mathfrak{A}}), [\![\bar{\varphi}^2]\!](\mathfrak{B}, (\varsigma \cup \rho)\restriction_{\mathfrak{B}})).$$

Proof. This can be shown using the exact same methods as in the proofs of Lemma 3.30 and Theorems 3.18 and 3.19. Note that, since every formula $\varphi \in \mathrm{wMSO}(\sigma, K)$ is equivalent to some almost Boolean formula, we do not need any assumptions on the finiteness of our structures. □

Courcelle's transductions

Like translation schemes, *transductions* provide a tool to translate structures over one signature into structures over another signature. Transductions extend our notion of translation scheme by allowing multiple copies of the given universe. More precisely, a σ-τ-translation scheme is a 1-σ-τ transduction in the sense defined below. In the following, we show that, with some adjustments, our weighted Feferman-Vaught Theorems can also be applied to transductions. For a survey on transductions, see [23].

Definition 3.39 ([23])**.** Let $k > 0$ be a natural number, $[k] = \{1, \ldots, k\}$, and

$$\tau * k = \{(T, \bar{\imath}) \mid T \in \mathrm{Rel}_\tau \text{ and } \bar{\imath} \in [k]^{\mathrm{ar}_\tau(T)}\}.$$

A *k-σ-τ-transduction Ψ over $\mathcal{W}$ and $\mathcal{Z}$* is a tuple $(\psi^1_{\mathcal{U}}, \ldots, \psi^k_{\mathcal{U}}, (\psi_w)_{w\in\tau*k})$ where $\psi^i_{\mathcal{U}}, \psi_w \in \mathrm{MSO}(\sigma)$ are formulas with variables from $\mathcal{W} \cup \mathcal{Z}$. The variables from $\mathcal{Z}$ may not be used for quantification, i.e., all variables from $\mathcal{Z}$ must be free.

Intuitively, the formulas $\psi^1_{\mathcal{U}}, \ldots, \psi^k_{\mathcal{U}}$ are filters for the copies of the universe, i.e., for an element a from the universe of the σ-structure, there will be one copy of a in the universe of the new τ-structure for each $\psi^i_{\mathcal{U}}$ which is satisfied when the free variable z is mapped to a. Likewise, the formulas $\psi_{(T,\bar{\imath})}$ are used to define the interpretation of T for the new τ-structure, where $\bar{\imath}$ determines from which copy of the universe each entry of the new tuple has to be.

For a σ-structure $\mathfrak{A} = (A, \mathcal{I}_{\mathfrak{A}})$ and a $(\mathcal{W}, \mathfrak{A})$-assignment ς, the *Ψ-induced τ-structure of $\mathfrak{A}$ and ς*, denoted by $\Psi^\star(\mathfrak{A}, \varsigma)$, is defined as a τ-structure with universe $\mathcal{U}_{\mathfrak{C}}$ and interpretation $\mathcal{I}_{\mathfrak{C}}$ as follows. For $i \in \{1, \ldots, k\}$ we define

$$A_i = \{a \in A \mid (\mathfrak{A}, \varsigma[z \to a]) \models \psi^i_{\mathcal{U}}\}$$

and let $\iota_i \colon A_i \to A_1 \sqcup \ldots \sqcup A_k$ be the inclusions. Then we let

$$\begin{aligned} \mathcal{U}_{\mathfrak{C}} &= A_1 \sqcup \ldots \sqcup A_k \\ \mathcal{I}_{\mathfrak{C}}(T) &= \bigcup_{\bar{\imath}\in[k]^{\mathrm{ar}_\tau(T)}} \{(\iota_{i_1}(a_1), \ldots, \iota_{i_{\mathrm{ar}_\tau(T)}}(a_{\mathrm{ar}_\tau(T)})) \mid (a_1, \ldots, a_{\mathrm{ar}_\tau(T)}) \in \\ &\qquad A_{i_1} \times \ldots \times A_{i_{\mathrm{ar}_\tau(T)}} \text{ and } (\mathfrak{A}, \varsigma[\bar{z} \to \bar{a}]) \models \psi_{(T,\bar{\imath})}\} \end{aligned}$$

where $\bar{\imath} = (i_1, \dots, i_{\mathrm{ar}_\tau(T)})$ and $\bar{a} = (a_1, \dots, a_{\mathrm{ar}_\tau(T)})$.

We refrain from restricting the domain of the transduction, as it does not make any difference for our purpose.

We can prove an analogue of Lemma 3.30 for transductions. Therefore, Theorems 3.18 and 3.19 are true for transductions as well. However, we have to make two small concessions. First, for the Boolean fragment of our first order logic, we need a new atomic formula $\mathtt{def}(x)$, where x is a first order variable. This formula is satisfied if the variable x is defined, and otherwise it is not satisfied. For our second order logic, we use $\mathtt{def}(x)$ as an abbreviation for the formula $\exists X.(x \in X)$. We denote by $\mathtt{def}\text{-wFO}(\sigma, K)$ the first order logic where $\mathtt{def}(x)$ is allowed as an atomic formula. Second, the variables of the formula we want to "translate" do usually not suffice for the translated formula. In particular, the translated formula potentially has more free variables than the formula to translate.

For a set of first and second order variables $\mathcal{V}$ and $k > 0$, we let $\mathcal{V}^{\sqcup k} = \{\mathcal{X}^i \mid \mathcal{X} \in \mathcal{V}, i \in \{1, \dots, k\}\}$ be the set of variables containing k copies of every variable from $\mathcal{V}$. Then, with the above notation, we define for a $(\mathcal{V}, \Psi^\star(\mathfrak{A}, \varsigma))$-assignment ρ the $(\mathcal{V}^{\sqcup k}, \mathfrak{A})$-assignment $\rho^\#$ by

$$\rho^\#(\mathcal{X}^i) = \begin{cases} \iota_i^{-1}(\rho(\mathcal{X}) \cap \iota_i(A)) & \text{if } \mathcal{X} \text{ is a second order variable} \\ \iota_i^{-1}(\rho(\mathcal{X})) & \text{if } \mathcal{X} \text{ is a first order variable and } \rho(\mathcal{X}) \in \iota_i(A) \\ \text{undefined} & \text{if } \mathcal{X} \text{ is a first order variable and } \rho(\mathcal{X}) \notin \iota_i(A). \end{cases}$$

Then we have the following lemma.

Lemma 3.40. *Let K be a commutative semiring. Let $\Psi = (\psi^1_{\mathcal{U}}, \dots, \psi^k_{\mathcal{U}}, (\psi_w)_{w \in \tau * k})$ be a k-σ-τ-transduction over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \mathtt{def}\text{-wFO}(\tau, K)$ or $\varphi \in \mathrm{wMSO}(\tau, K)$ with variables from $\mathcal{V}$. Then there exists a formula $\psi \in \mathtt{def}\text{-wFO}(\sigma, K)$ or $\psi \in \mathrm{wMSO}(\sigma, K)$, respectively, with $\mathrm{Free}(\psi) \subseteq \mathrm{Free}(\varphi)^{\sqcup k} \cup \mathrm{Free}(\Psi)$ such that the following holds. For all structures $\mathfrak{A} \in \mathrm{Str}(\sigma)$, or, for all structures $\mathfrak{A} \in \mathrm{Str}(\sigma)$ if K is bicomplete, all $(\mathcal{W}, \mathfrak{A})$-assignments ς, and all $(\mathcal{V}, \Psi^\star(\mathfrak{A}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Psi^\star(\mathfrak{A}, \varsigma), \rho) = [\![\psi]\!](\mathfrak{A}, \varsigma \cup \rho^\#).$$

If φ is from $\mathrm{wMSO}^{\otimes\text{-res}}(\tau, K)$ or $\mathtt{def}\text{-wFO}^{\otimes\text{-free}}(\tau, K)$, then ψ can also be chosen as a formula from $\mathrm{wMSO}^{\otimes\text{-res}}(\sigma, K)$ or $\mathtt{def}\text{-wFO}^{\otimes\text{-free}}(\sigma, K)$, respectively. Furthermore, if φ does not contain free variables, ψ can be chosen to not contain any subformula of the form $\mathtt{def}(x)$.

Proof. We proceed by induction and first cover the Boolean case.
If $\varphi = T(x_1, \dots, x_n)$ for some $T \in \mathrm{Rel}_\tau$, we let

$$\psi = \bigvee_{\bar{\imath} \in [k]^n} \Big(\psi_{(T,\bar{\imath})}(x_1^{i_1}, \dots, x_n^{i_n}) \wedge \bigwedge_{j=1}^{n} \mathtt{def}(x_j^{i_j}) \Big).$$

If $\varphi = (x \in X)$, we let $\psi = \bigvee_{i=1}^k x^i \in X^i$. If $\varphi = \mathtt{def}(x)$ we let $\psi = \bigvee_{i=1}^k \mathtt{def}(x^i)$.

Now we assume that by induction, the theorem holds for the formulas φ_1, φ_2, and φ' with the formulas ψ_1, ψ_2, and ψ'. If $\varphi = \varphi_1 \vee \varphi_2$, we let $\psi = \psi_1 \vee \psi_2$, and if $\varphi = \neg\varphi'$, we let $\psi = \neg\psi'$.

If $\varphi = \exists x.\varphi'$, we define for $i \in \{1, \dots, k\}$ the formula ψ'^{+i} as the formula obtained by replacing all atomic subformulas in ψ' that contain a variable x^j with $j \neq i$ by $\texttt{false}$. Then we let $\psi = \bigvee_{i=1}^{k} \exists x^i.\, (\psi^i_{\mathcal{U}}(x^i) \wedge \psi'^{+i})$.

If $\varphi = \exists X.\varphi'$, we let $\psi = \exists X^1 \dots \exists X^k.\, \Big(\psi' \wedge \bigwedge_{i=1}^{k} \forall x.(x \in X^i \to \psi^i_{\mathcal{U}}(x))\Big)$, where x is a new first order variable.

We now turn to the weighted case.

If $\varphi = \kappa \in K$, we let $\psi = \kappa$.

If $\varphi = \varphi_1 \oplus \varphi_2$ or $\varphi = \varphi_1 \otimes \varphi_2$, we let $\psi = \psi_1 \oplus \psi_2$ or $\psi = \psi_1 \otimes \psi_2$, respectively.

If $\varphi = \bigoplus x.\varphi'$, we let $\psi = \bigoplus_{i=1}^{k} \bigoplus x^i.\, (\psi^i_{\mathcal{U}}(x^i) \otimes \psi'^{+i})$.

If $\varphi = \bigoplus X.\varphi'$, we let $\psi = \bigoplus X^1 \dots \bigoplus X^k.\, \Big(\psi' \otimes \bigwedge_{i=1}^{k} \forall x.(x \in X^i \to \psi^i_{\mathcal{U}}(x))\Big)$, where x is a new first order variable.

If $\varphi = \bigotimes x.\varphi'$, we let $\psi = \bigotimes_{i=1}^{k} \bigotimes x^i.((\psi'^{+i} \otimes \psi^i_{\mathcal{U}}(x^i)) \oplus \neg\psi^i_{\mathcal{U}}(x^i))$.

If $\varphi = \bigotimes X.\varphi'$ we define $\beta = \bigwedge_{i=1}^{k} \forall x.(x \in X^i \to \psi^i_{\mathcal{U}}(x))$, where x is a new first order variable, and let $\psi = \bigotimes X^1 \dots \bigotimes X^k.((\psi' \otimes \beta) \oplus \neg\beta)$.

To see that all atomic subformulas $\texttt{def}(x)$ in ψ can be removed if φ does not contain free variables, note that every subformula $\texttt{def}(x)$ can be replaced by $\texttt{true}$ without changing the semantics of ψ if x is a bound variable. □

With this, we have the following versions of Theorems 3.18 and 3.19 for transductions.

Theorem 3.41. *Let K be a commutative semiring. Let $\Psi = (\psi^1_{\mathcal{U}}, \dots, \psi^k_{\mathcal{U}}, (\psi_w)_{w \in \tau * k})$ be a k-σ-τ-transduction over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \texttt{def}\text{-wMSO}^{\bigotimes\text{-res}}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar\varphi^{\mathbb{1}}, \bar\varphi^{2} \in \texttt{def}\text{-wMSO}^{\bigotimes\text{-res}}(\sigma, K)^n$ with $\mathrm{Free}(\bar\varphi^{\mathbb{1}}) \cup \mathrm{Free}(\bar\varphi^{2}) \subseteq \mathrm{Free}(\varphi)^{\sqcup k} \cup \mathrm{Free}(\Psi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$ such that the following holds. For all finite structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, or, for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ if K is bicomplete, all $(\mathcal{W}, \mathfrak{A} \sqcup \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Psi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Psi^\star(\mathfrak{A} \sqcup \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar\varphi^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho^{\#})\restriction_{\mathfrak{A}}), [\![\bar\varphi^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho^{\#})\restriction_{\mathfrak{B}})).$$

Theorem 3.42. *Let K be a commutative semiring. Let $\Psi = (\psi^1_{\mathcal{U}}, \dots, \psi^k_{\mathcal{U}}, (\psi_w)_{w \in \tau * k})$ be a k-σ-τ-transduction over $\mathcal{W}$ and $\mathcal{Z}$, $\mathcal{V}$ be a set of first and second order variables such that $\mathcal{V}$, $\mathcal{W}$, and $\mathcal{Z}$ are pairwise disjoint, and $\varphi \in \texttt{def}\text{-wFO}^{\bigotimes\text{-free}}(\tau, K)$ with variables from $\mathcal{V}$. Then there exist $n \geq 1$, tuples of formulas $\bar\varphi^{\mathbb{1}}, \bar\varphi^{2} \in \texttt{def}\text{-wFO}^{\bigotimes\text{-free}}(\sigma, K)^n$ with $\mathrm{Free}(\bar\varphi^{\mathbb{1}}) \cup \mathrm{Free}(\bar\varphi^{2}) \subseteq \mathrm{Free}(\varphi)^{\sqcup k} \cup \mathrm{Free}(\Psi)$, and an expression $E_\varphi \in \mathrm{Exp}_n(K)$ such that the following holds. For all finite structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$, or, for all structures $\mathfrak{A}, \mathfrak{B} \in \mathrm{Str}(\sigma)$ if K is bicomplete, all $(\mathcal{W}, \mathfrak{A} \times \mathfrak{B})$-assignments ς, and all $(\mathcal{V}, \Psi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma))$-assignments ρ we have*

$$[\![\varphi]\!](\Psi^\star(\mathfrak{A} \times \mathfrak{B}, \varsigma), \rho) = \langle\!\langle E_\varphi \rangle\!\rangle([\![\bar\varphi^{\mathbb{1}}]\!](\mathfrak{A}, (\varsigma \cup \rho^{\#})\restriction_{\mathfrak{A}}), [\![\bar\varphi^{2}]\!](\mathfrak{B}, (\varsigma \cup \rho^{\#})\restriction_{\mathfrak{B}})).$$

Proof. Theorems 3.41 and 3.42 are immediate by first applying Lemma 3.40 and then Theorem 3.18 or Theorem 3.19, respectively, while treating `def` as a relation symbol. □

4

Decidable Properties of Max-Plus Tree Automata

Homer	Kids, there's three ways to do things.
	The right way, the wrong way, and the Max Power way!
Bart	Isn't that the wrong way?
Homer	Yeah, but faster!

"Homer to the Max", *The Simpsons*

In this chapter, we extend four decidability results from max-plus word automata to max-plus tree automata. A max-plus word automaton is a finite automaton which assigns real numbers to words over a given alphabet. The transitions of a max-plus automaton each carry a weight from the real numbers. To every run of the automaton, a weight is associated by summing over the weights of the transitions which constitute the run. The weight of a word is given by the maximum over the weights of all runs on this word. More generally, max-plus word automata and their min-plus counterparts are weighted automata [98, 97, 68, 8, 29] over the max-plus or min-plus semiring. Min-plus automata were originally introduced by Imre Simon as a means to show the decidability of the *finite power property* [102, 103]. Since their introduction, max-plus and min-plus automata have enjoyed a continuing interest [66, 55, 63, 10, 24, 46] and they have been employed in many different contexts. To only name some examples, they can be used to determine the star height of a

language [54], to prove the termination of some string rewriting systems [106], and to model certain discrete event systems [64]. Additionally, they appear in the context of natural language processing [75], where for reasons of numerical stability, probabilities are often computed in the min-plus semiring as negative log-likelihoods.

For practical applications, the decidable properties of an automaton model are usually of great interest. Typical decidability problems considered include the *emptiness*, *universality*, *inclusion*, *equivalence*, *unambiguity*, and *sequentiality* problems. We consider the last three of these problems for *finitely ambiguous* automata and the lesser known *finite sequentiality* problem for *unambiguous* automata. Here, we call a max-plus word automaton unambiguous if there exists at most one accepting run on every word. We call it *finitely ambiguous* if the number of runs on each word is bounded by a global constant. Moreover, if on every word the number of accepting runs is bounded polynomially in the length of the word, we call the automaton *polynomially ambiguous*. As a special type of unambiguity, we consider *determinism* or *sequentiality*. We call a max-plus word automaton *deterministic* or *sequential* if at most one of its states is initial and for each pair of a state and an input symbol, there is at most one valid transition into a next state. Note that the ambiguity of a max-plus automaton is a decidable property, as it is easily reduced to deciding the ambiguity of a finite automaton. Deciding the sequentiality of a finite automaton is trivial, polynomial time algorithms for deciding the unambiguity, the finite ambiguity, and the polynomial ambiguity of a finite automaton can be found in [11, 108, 100]. Furthermore, the classes of functions definable by deterministic, unambiguous, and finitely ambiguous max-plus automata form a strictly ascending hierarchy [63], and the classes of functions definable by finitely ambiguous, polynomially ambiguous, and arbitrary min-plus automata form a strictly ascending hierarchy [58, 74]. The methods of [58] and [74] can be used to show that the expressive hierarchy of finitely ambiguous, polynomially ambiguous, and arbitrary max-plus automata is strict as well. In the following, we quickly recall the considered decidability problems and the related results.

The *equivalence problem* asks whether two given max-plus automata coincide on the weights they assign to each word. In general, the equivalence problem is undecidable for max-plus automata [66], but for finitely ambiguous max-plus word automata it becomes decidable [107, 55]. The *sequentiality problem* asks whether for a given max-plus automaton, there exists an equivalent deterministic automaton. This problem was shown to be decidable by Mohri [75] for unambiguous max-plus word automata. The *unambiguity problem* asks whether for a given max-plus automaton, there exists an equivalent unambiguous automaton. This problem is known to be decidable for finitely ambiguous [63] and even polynomially ambiguous max-plus word automata [61]. In conjunction with Mohri's results, it follows that the sequentiality problem is decidable for these classes of automata as well. Finally, the *finite sequentiality problem* asks whether a given max-plus automaton can be represented as a pointwise maximum of finitely many deterministic max-plus automata. In [55], it was left as an open question to determine the decidability of the finite sequentiality problem for finitely ambiguous max-plus automata. It was shown only recently that for the classes of unambiguous as well as finitely ambiguous automata, the finite sequentiality problem is decidable [5, 4]. The class of functions which allow

a finitely sequential representation by max-plus automata lies strictly between the classes of functions definable by deterministic and by finitely ambiguous max-plus automata, and it is incomparable to the class of functions definable by unambiguous max-plus automata [63].

In this chapter, we investigate all four of these problems for max-plus tree automata. In the same way that finite automata have been generalized to finite tree automata [49, 48], weighted automata have been generalized to weighted tree automata [1, 7, 39, 47]. Max-plus tree automata are weighted tree automata over the max-plus semiring and are thus in particular a generalization of max-plus word automata. Applications for max-plus tree automata include proving the termination of certain term rewriting systems [65], and they are also commonly employed in natural language processing [87] in the form of *probabilistic context-free grammars*. We will show that the equivalence, unambiguity, and sequentiality problems are decidable for finitely ambiguous max-plus tree automata, and that the finite sequentiality problem is decidable for unambiguous max-plus tree automata.

Our approach to the decidability of the equivalence problem employs ideas from [55]. We reduce the equivalence problem to the same decidable problem as [55], namely the decidability of the existence of an integer solution for a system of linear inequalities [80]. However, instead of the *cycle decompositions* which were used both in [55] and [84], we employ Parikh's theorem [81, Theorem 2]. This idea was suggested by Mikołaj Bojańczyk in a discussion following the presentation of the proof from [84]. The proof presented here is a revised version of the one from [84]. We note that our solution of the equivalence problem can be applied to weighted logics. In [83], a fragment of a weighted logic is shown to have the same expressive power as finitely ambiguous weighted tree automata. Over the max-plus semiring, equivalence is decidable for formulas of this fragment due to our results.

The decidability of the unambiguity problem employs ideas from [63]. Here, we show how the *dominance property* can be generalized to max-plus tree automata. To show the decidability of the sequentiality problem for finitely ambiguous max-plus tree automata, we first combine results from [18] and [75] to show the decidability of this problem for unambiguous max-plus tree automata, and then combine this result with the decidability of the unambiguity problem. For the finite sequentiality problem, we employ ideas from [5]. We show how the *fork property* can be generalized to max-plus tree automata and that for unambiguous max-plus tree automata, this generalization is a criterion for deciding finite sequentiality.

Except for the equivalence problem, we show all decidability results for max-plus automata with weights in the real numbers. Therefore, we want to point out how we understand the word "decidable" here. Clearly, not every real number can be finitely represented, thus our results can only hold for max-plus automata with weights in the computable real numbers. Admittedly, however, in the most general sense our results do not even hold for the computable reals for the following reason. For our decision procedures to be effective, we need to be able to check arbitrary finite sums of transition weights for zero. However, although having algorithms to compute two real numbers to an arbitrary precision allows us to compute their sum to an arbitrary precision, we are not necessarily able to decide whether their sum equals zero. We therefore assume the following "meta-restriction" on the weights of our max-plus

automata. By considering the real numbers as a vector space over the field of rational numbers, we obtain the notion of linearly independent sets of real numbers. We will assume that for each max-plus automaton, we are given a finite linearly independent set of computable real numbers such that each weight of the automaton is a finite linear combination with rational coefficients of numbers from this set. Under this assumption, all our algorithms are effective.

An extended abstract of our results on the equivalence, the unambiguity, and the sequentiality problems appeared at the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS) in 2017 [84]. Our results on the finite sequentiality problem have appeared as an extended abstract at the 36th International Symposium on Theoretical Aspects of Computer Science (STACS) in 2019 [86].

4.1 Max-Plus Automata

In the following, we introduce max-plus automata on words and trees and the notion of ambiguity of these automata models.

Trees

We recall that $\mathbb{N}^*$ denotes the set of all finite words over $\mathbb{N}$. By $\leq_p$, we denote the prefix-relation on $\mathbb{N}^*$, and by $\leq_l$, we denote the lexicographic order on $\mathbb{N}^*$, i.e., the relations

$$\leq_p = \{(u, uv) \mid u, v \in \mathbb{N}^*\}$$
$$\leq_l = \{(uiv_1, ujv_2) \mid u, v_1, v_2 \in \mathbb{N}^*, i, j \in \mathbb{N}, i < j\} \cup \leq_p.$$

Note that $\leq_p$ is a partial order and $\leq_l$ is a total order on $\mathbb{N}^*$. Two words from $\mathbb{N}^*$ are called *prefix-dependent* if they are in prefix relation, and otherwise they are called *prefix-independent*. We call a set $X \subseteq \mathbb{N}^*$ *prefix-closed* if $uv \in X$ implies $u \in X$ for every two words $u, v \in \mathbb{N}^*$.

A *ranked alphabet* is a pair $(\Gamma, \mathrm{rk}_\Gamma)$, often abbreviated by Γ, where Γ is a finite set and $\mathrm{rk}_\Gamma \colon \Gamma \to \mathbb{N}$ a mapping which assigns a *rank* to every symbol. For every $m \geq 0$, we define $\Gamma^{(m)} = \mathrm{rk}_\Gamma^{-1}(m)$ as the set of all symbols of rank m. The rank of Γ is defined as $\mathrm{rk}(\Gamma) = \max\{\mathrm{rk}_\Gamma(a) \mid a \in \Gamma\}$.

The set of (*finite, labeled, and ordered*) Γ-*trees*, denoted by T_Γ, is the set of all pairs $t = (\mathrm{pos}(t), \mathrm{label}_t)$, where $\mathrm{pos}(t) \subset \mathbb{N}_+^*$ is a finite non-empty prefix-closed set of *positions*, $\mathrm{label}_t \colon \mathrm{pos}(t) \to \Gamma$ is a mapping, and for every $w \in \mathrm{pos}(t)$ we have $wi \in \mathrm{pos}(t)$ iff $1 \leq i \leq \mathrm{rk}_\Gamma(\mathrm{label}_t(w))$. We write $t(w)$ for $\mathrm{label}_t(w)$ and $|t|$ for $|\mathrm{pos}(t)|$. We also refer to the elements of $\mathrm{pos}(t)$ as *nodes*, to ε as the *root* of t, and to prefix-maximal nodes as *leaves*. The *height* of t is defined as $\mathrm{height}(t) = \max_{w \in \mathrm{pos}(t)} |w|$. For a leaf $w \in \mathrm{pos}(t)$, the set $\{v \in \mathrm{pos}(t) \mid v \leq_p w\}$ is called a *branch* of t.

Now let $s, t \in T_\Gamma$ and $w \in \mathrm{pos}(t)$. The *subtree of t at w*, denoted by $t{\restriction_w}$, is a Γ-tree defined as follows. We let $\mathrm{pos}(t{\restriction_w}) = \{v \in \mathbb{N}^* \mid wv \in \mathrm{pos}(t)\}$ and for $v \in \mathrm{pos}(t{\restriction_w})$, we let $\mathrm{label}_{t{\restriction_w}}(v) = t(wv)$.

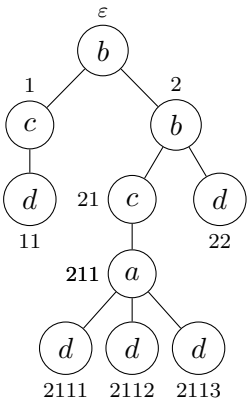
The *substitution of s into w of t*, denoted by $t\langle s \to w\rangle$, is a Γ-tree defined as follows. We let $\mathrm{pos}(t\langle s \to w\rangle) = (\mathrm{pos}(t) \setminus \{v \in \mathrm{pos}(t) \mid w \leq_p v\}) \cup \{wv \mid v \in \mathrm{pos}(s)\}$. For $v \in \mathrm{pos}(t\langle s \to w\rangle)$, we let $\mathrm{label}_{t\langle s\to w\rangle}(v) = s(u)$ if $v = wu$ for some $u \in \mathrm{pos}(s)$, and otherwise $\mathrm{label}_{t\langle s\to w\rangle}(v) = t(v)$.

For $a \in \Gamma^{(m)}$ and trees $t_1, \ldots, t_m \in T_\Gamma$, we also write $a(t_1, \ldots, t_m)$ to denote the tree t with $\mathrm{pos}(t) = \{\varepsilon\} \cup \{iw \mid i \in \{1, \ldots, m\}, w \in \mathrm{pos}(t_i)\}$, $\mathrm{label}_t(\varepsilon) = a$, and $\mathrm{label}_t(iw) = t_i(w)$. For $a \in \Gamma^{(0)}$, the tree $a()$ is abbreviated by a.

For a ranked alphabet Γ, a tree over the alphabet $\Gamma_\diamond = (\Gamma \cup \{\diamond\}, \mathrm{rk}_\Gamma \cup \{\diamond \mapsto 0\})$ is called a Γ-*context*. Let $t \in T_{\Gamma_\diamond}$ be a Γ-context and let $w_1, \ldots, w_n \in \mathrm{pos}(t)$ be a lexicographically ordered enumeration of all leaves of t labeled $\diamond$. Then we call t an n-Γ-*context* and define $\Diamond_i(t) = w_i$ for $i \in \{1, \ldots, n\}$. For an n-Γ-context t and contexts $t_1, \ldots, t_n \in T_{\Gamma_\diamond}$, we define $t(t_1, \ldots, t_n) = t\langle t_1 \to \Diamond_1(t)\rangle \ldots \langle t_n \to \Diamond_n(t)\rangle$ by substitution of $t_1, \ldots, t_n$ into the $\diamond$-leaves of t. A 1-Γ-context is also called a Γ-*word*. For a Γ-word s, we define $s^0 = \diamond$ and $s^{n+1} = s(s^n)$ for $n \geq 0$.

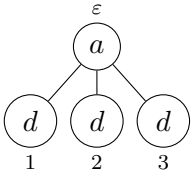
Example 4.1. Let $\Gamma = \{a, b, c, d\}$ with $\mathrm{rk}_\Gamma(a) = 3$, $\mathrm{rk}_\Gamma(b) = 2$, $\mathrm{rk}_\Gamma(c) = 1$ and $\mathrm{rk}_\Gamma(d) = 0$. Then an example tree is

$$t = b(c(d), b(c(a(d, d, d)), d))$$
$$\text{with } \mathrm{pos}(t) = \{\varepsilon, 1, 11, 2, 21, 211, 2111, 2112, 2113, 22\}.$$



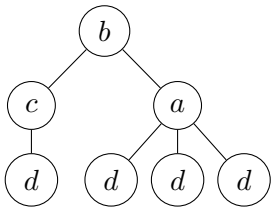
The subtree $t|_{211}$ of t at position 211 is the tree

$$t|_{211} = a(d, d, d) \text{ with } \mathrm{pos}(t|_{211}) = \{\varepsilon, 1, 2, 3\}.$$



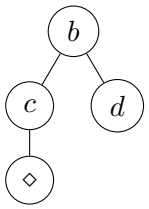
To illustrate substitution, we substitute the subtree $t|_{211}$ into position 2 of t.

$$t\langle t\restriction_{211} \to 2\rangle = b(c(d), a(d, d, d))$$



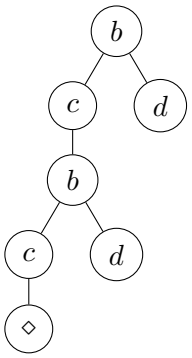
By substituting the tree $\diamond$ into position 211 of t and taking the subtree at 2 of the resulting tree, we obtain a Γ-word $s = t\langle \diamond \to 211\rangle\restriction_2$ as follows.

$$t\langle \diamond \to 211\rangle\restriction_2 = b(c(\diamond), d)$$



Then the second power s^2 of s is the Γ-word

$$s^2 = b(c(b(c(\diamond), d)), d).$$



Weighted Automata on Words and Trees

Although we will not employ or derive results for weighted automata on words, we will often compare our results for tree automata to the corresponding results for word automata. Therefore, we want to briefly define what we understand under a weighted automaton on words.

Let $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ be a commutative semiring and Σ an alphabet. A *weighted finite automaton (short: WA)* over K and Σ is a tuple $\mathcal{A} = (Q, \Sigma, \lambda, \mu, \nu)$ where Q is a finite set (of states), Σ the input alphabet, $\lambda\colon Q \to K$ (the function of initial weights), $\mu\colon Q \times \Sigma \times Q \to K$ (the function of transition weights), and $\nu\colon Q \to K$

(the function of final weights). A tuple $(p, a, q) \in Q \times \Sigma \times Q$ is called a *transition* and (p, a, q) is called *valid* if $\mu(p, a, q) \neq \mathbb{0}$. A state $q \in Q$ is called *initial* if $\lambda(q) \neq \mathbb{0}$ and it is called *final* if $\nu(q) \neq \mathbb{0}$. We call a WA over the max-plus semiring a max-plus-WA and a WA over the Boolean semiring a *nondeterministic finite automaton* (NFA). An NFA $\mathcal{A} = (Q, \Gamma, \lambda, \mu, \nu)$ is also written as a tuple $\mathcal{A}' = (Q, \Gamma, I, \delta, F)$ where $I = \{q \in Q \mid \lambda(q) = 1\}$, $\delta = \{d \in Q \times \Sigma \times Q \mid \mu(d) = 1\}$, and $F = \{q \in Q \mid \nu(q) = 1\}$. For an overview of finite automata theory, see also [57].

For a word $w = a_1 a_2 \dots a_n \in \Sigma^*$, a *(valid) run of $\mathcal{A}$ on w* is a finite sequence of valid transitions $r = (q_0, a_1, q_1) \dots (q_{n-1}, a_n, q_n)$ from $Q \times \Sigma \times Q$. We call r *accepting* if q_0 is initial and q_n is final. We let $\lambda(r) = \lambda(q_0)$ and $\nu(r) = \nu(q_n)$. By $\mathrm{Run}_{\mathcal{A}}(w)$ and $\mathrm{Acc}_{\mathcal{A}}(w)$, we denote the sets of all runs and all accepting runs of $\mathcal{A}$ on w, respectively. The *weight of r* is defined by

$$\mathrm{wt}_{\mathcal{A}}(r) = \bigodot_{i=1}^{n} \mu(q_{i-1}, a_i, q_i).$$

The *behavior of $\mathcal{A}$*, denoted by $[\![\mathcal{A}]\!]$, is the mapping defined for every $w \in \Sigma^*$ by

$$[\![\mathcal{A}]\!](w) = \bigoplus_{r \in \mathrm{Acc}_{\mathcal{A}}(w)} (\lambda(r) \odot \mathrm{wt}_{\mathcal{A}}(r) \odot \nu(r)),$$

where the sum over the empty set is $\mathbb{0}$ by convention. The *support* of $\mathcal{A}$ is the set $\mathrm{supp}(\mathcal{A}) = \{w \in \Sigma^* \mid [\![\mathcal{A}]\!](w) \neq \mathbb{0}\}$. The support of an NFA $\mathcal{A}$ is also called the *language accepted by $\mathcal{A}$* and is denoted by $\mathcal{L}(\mathcal{A})$. A subset $L \subseteq \Sigma^*$ is called *recognizable* if there exists an NFA $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.

Next, we recall weighted tree automata. Let $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ be a commutative semiring and Γ a ranked alphabet. A *weighted bottom-up finite state tree automaton (short: WTA) over K and Γ* is a tuple $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ where Q is a finite set (of states), Γ is a ranked alphabet (of input symbols), $\mu \colon \bigcup_{m=0}^{\mathrm{rk}(\Gamma)} Q^m \times \Gamma^{(m)} \times Q \to K$ (the function of transition weights), and $\nu \colon Q \to K$ (the function of final weights). We define $\Delta_{\mathcal{A}} = \mathrm{dom}(\mu)$. A tuple $(\bar{p}, a, q) \in \Delta_{\mathcal{A}}$ is called a *transition* and $(\bar{p}, a, q)$ is called *valid* if $\mu(\bar{p}, a, q) \neq \mathbb{0}$. A state $q \in Q$ is called *final* if $\nu(q) \neq \mathbb{0}$.

We call a WTA over the max-plus semiring a max-plus-WTA and a WTA over the Boolean semiring a *finite tree automaton* (FTA). An FTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ is also written as a tuple $\mathcal{A}' = (Q, \Gamma, \delta, F)$ where $\delta = \{d \in \Delta_{\mathcal{A}} \mid \mu(d) = 1\}$ and $F = \{q \in Q \mid \nu(q) = 1\}$. For an overview of the theory of finite tree automata, see also [49].

For a tree $t \in T_\Gamma$, a mapping $r \colon \mathrm{pos}(t) \to Q$ is called a *quasi-run of $\mathcal{A}$ on t*. For a quasi-run r on t and a position $w \in \mathrm{pos}(t)$ with $t(w) = a \in \Gamma^{(m)}$, the tuple

$$\mathbb{t}(t, r, w) = (r(w1), \dots, r(wm), a, r(w))$$

is called the *transition at w*. The quasi-run r is called a *(valid) run* if for every $w \in \mathrm{pos}(t)$ the transition $\mathbb{t}(t, r, w)$ is valid with respect to $\mathcal{A}$. We call a run r *accepting* if $r(\varepsilon)$ is final. By $\mathrm{Run}_{\mathcal{A}}(t)$ and $\mathrm{Acc}_{\mathcal{A}}(t)$, we denote the sets of all runs and all accepting runs of $\mathcal{A}$ on t, respectively. For a state $q \in Q$, we denote by $\mathrm{Run}_{\mathcal{A}}(t, q)$ the set of all runs $r \in \mathrm{Run}_{\mathcal{A}}(t)$ such that $r(\varepsilon) = q$.

For a run $r \in \mathrm{Run}_{\mathcal{A}}(t)$, the *weight of* r is defined by

$$\mathrm{wt}_{\mathcal{A}}(t,r) = \bigodot_{w \in \mathrm{pos}(t)} \mu(\mathbb{t}(t,r,w)).$$

The *behavior of* $\mathcal{A}$, denoted by $[\![\mathcal{A}]\!]$, is the mapping defined for every $t \in T_\Gamma$ by

$$[\![\mathcal{A}]\!](t) = \bigoplus_{r \in \mathrm{Acc}_{\mathcal{A}}(t)} (\mathrm{wt}_{\mathcal{A}}(t,r) \odot \nu(r(\varepsilon))),$$

where again the sum over the empty set is $\mathbb{0}$ by convention. The *support* of $\mathcal{A}$ is the set $\mathrm{supp}(\mathcal{A}) = \{t \in T_\Gamma \mid [\![\mathcal{A}]\!](t) \neq \mathbb{0}\}$. The support of an FTA $\mathcal{A}$ is also called the *(tree) language accepted by* $\mathcal{A}$ and is denoted by $\mathcal{L}(\mathcal{A})$. A subset $L \subseteq T_\Gamma$ is called *recognizable* if there exists an FTA $\mathcal{A}$ with $L = \mathcal{L}(\mathcal{A})$.

For a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$, a run of $\mathcal{A}$ on a Γ-context t is a run of the WTA $\mathcal{A}' = (Q, \Gamma_\diamond, \mu', \nu)$ on t, where $\mu'(\diamond, q) = \mathbb{1}$ for all $q \in Q$ and $\mu'(d) = \mu(d)$ for all $d \in \Delta_{\mathcal{A}}$. We denote $\mathrm{Run}^\diamond_{\mathcal{A}}(t) = \mathrm{Run}_{\mathcal{A}'}(t)$ and for $r \in \mathrm{Run}^\diamond_{\mathcal{A}}(t)$ write $\mathrm{wt}^\diamond_{\mathcal{A}}(t,r) = \mathrm{wt}_{\mathcal{A}'}(t,r)$. For an n-Γ-context $t \in T_{\Gamma_\diamond}$ and states $q_0, \ldots, q_n$, we denote by $\mathrm{Run}^\diamond_{\mathcal{A}}(q_1, \ldots, q_n, t, q_0)$ the set of all runs $r \in \mathrm{Run}^\diamond_{\mathcal{A}}(t)$ such that $r(\varepsilon) = q_0$ and $r(\Diamond_i(t)) = q_i$ for every $i \in \{1, \ldots, n\}$. For a Γ-word s, we write $p \xrightarrow{s|x} q$ if there exists a run $r \in \mathrm{Run}^\diamond_{\mathcal{A}}(p, s, q)$ with $\mathrm{wt}^\diamond_{\mathcal{A}}(s,r) = x$. In this case, r is said to *realize* $p \xrightarrow{s|x} q$. Note here that if $K = \mathbb{R}_{\max}$, then $r \in \mathrm{Run}^\diamond_{\mathcal{A}}(s)$ implies $x \neq -\infty$.

Similar to trees, we define restrictions, substitutions, and powers of runs as follows. Let $t, s \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{A}}(t)$, $w \in \mathrm{pos}(t)$, and $r_s \in \mathrm{Run}_{\mathcal{A}}(s)$ with $r_s(\varepsilon) = r(w)$. Then we define $r\restriction_w \in \mathrm{Run}_{\mathcal{A}}(t\restriction_w)$ by $r\restriction_w(v) = r(wv)$ for every $v \in \mathrm{pos}(t\restriction_w)$. We define $r\langle r_s \to w\rangle \in \mathrm{Run}_{\mathcal{A}}(t\langle s \to w\rangle)$ by $r\langle r_s \to w\rangle(v) = r_s(u)$ if $v = wu$ for some $u \in \mathrm{pos}(s)$, and $r\langle r_s \to w\rangle(v) = r(v)$ otherwise. For a Γ-word s and a run $r \in \mathrm{Run}^\diamond_{\mathcal{A}}(s)$ with $r(\varepsilon) = r(\Diamond_1(s))$, we let $v = \Diamond_1(s)$ and define $r^{0\langle v\rangle} = \{\varepsilon \mapsto r(\varepsilon)\}$ and $r^{n+1\langle v\rangle} = r\langle r^{n\langle v\rangle} \to v\rangle \in \mathrm{Run}^\diamond_{\mathcal{A}}(s^{n+1})$ for $n \geq 0$.

For a WTA $\mathcal{A}$, we define a relation $\preceq$ on Q by $p \preceq q$ iff there exists a Γ-word $s \in T_{\Gamma_\diamond}$ such that $\mathrm{Run}^\diamond_{\mathcal{A}}(q, s, p) \neq \emptyset$. We write $p \approx q$ if $p \preceq q$ and $q \preceq p$. By $[p]$ we denote the set of all $q \in Q$ with $p \approx q$.

A WTA $\mathcal{A}$ is called *trim* if for every $p \in Q$, there exist $t \in T_\Gamma$, $r \in \mathrm{Acc}_{\mathcal{A}}(t)$, and $w \in \mathrm{pos}(t)$ such that $r(w) = p$. The *trim part of* $\mathcal{A}$ is the automaton obtained from $\mathcal{A}$ by removing all states $p \in Q$ for which no such t, r, and w exist. This process obviously has no influence on $[\![\mathcal{A}]\!]$.

Ambiguity of Automata

A WA $\mathcal{A} = (Q, \Sigma, \lambda, \mu, \nu)$ over a semiring $(K, \oplus, \odot, \mathbb{0}, \mathbb{1})$ is called *deterministic* or *sequential* if (1) there is at most one initial state, i.e., at most one state $p \in Q$ with $\lambda(p) \neq \mathbb{0}$ and (2) for every $a \in \Sigma$ and $p \in Q$, there exists at most one state $q \in Q$ with $\mu(p, a, q) \neq \mathbb{0}$. If there exists an integer $M \geq 1$ such that $|\mathrm{Acc}_{\mathcal{A}}(w)| \leq M$ for every word $w \in \Sigma^*$, we say that $\mathcal{A}$ is *M-ambiguous*. We call $\mathcal{A}$ *finitely ambiguous* if it is M-ambiguous for some $M \geq 1$. A 1-ambiguous WA is also called *unambiguous*. If there exists a polynomial P such that $|\mathrm{Acc}_{\mathcal{A}}(w)| \leq P(|w|)$ for every $w \in \Sigma^*$, we call $\mathcal{A}$ *polynomially ambiguous*. The behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ is called *finitely sequential*

if there exist finitely many deterministic WA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over K and Σ such that $[\![\mathcal{A}]\!] = \bigoplus_{i=1}^{n} [\![\mathcal{A}_i]\!]$, where the sum is taken pointwise.

Likewise, a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ over K is called *deterministic* or *sequential* if for every $m \geq 0$, $a \in \Gamma^{(m)}$, and $\bar{p} \in Q^m$, there exists at most one $q \in Q$ with $\mu(\bar{p}, a, q) \neq \mathbb{0}$. If there exists $M \geq 1$ such that $|\mathrm{Acc}_{\mathcal{A}}(t)| \leq M$ for every every tree $t \in T_\Gamma$, we say that $\mathcal{A}$ is *M-ambiguous*. We call $\mathcal{A}$ *finitely ambiguous* if it is M-ambiguous for some $M \geq 1$ and we call a 1-ambiguous WTA *unambiguous*. We call $\mathcal{A}$ *polynomially ambiguous* if there exists a polynomial P such that $|\mathrm{Acc}_{\mathcal{A}}(t)| \leq P(|t|)$ for every $t \in T_\Gamma$. Finally, we call the behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ *finitely sequential* if there exist finitely many deterministic WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over K and Γ such that $[\![\mathcal{A}]\!] = \bigoplus_{i=1}^{n} [\![\mathcal{A}_i]\!]$, where the sum is taken pointwise.

Remark 4.2. A trim WA $\mathcal{A}$ is deterministic if and only if $|\mathrm{Run}_{\mathcal{A}}(w)| \leq 1$ for every $w \in \Sigma^*$, and a trim WTA $\mathcal{A}$ is deterministic if and only if $|\mathrm{Run}_{\mathcal{A}}(t)| \leq 1$ for every $t \in T_\Gamma$.

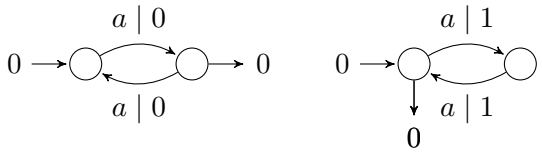
We can decide whether a weighted word or tree automaton is deterministic, unambiguous, finitely ambiguous, or polynomially ambiguous as follows. If $\mathcal{A} = (Q, \Sigma, \lambda, \mu, \nu)$ is a WA over K, we consider the NFA $\mathcal{A}' = (Q, \Sigma, I, \delta, F)$ where $I = \{q \in Q \mid q \text{ is initial in } \mathcal{A}\}$, $\delta = \{d \in \Delta_{\mathcal{A}} \mid d \text{ is valid in } \mathcal{A}\}$, and $F = \{q \in Q \mid q \text{ is final in } \mathcal{A}\}$. Then $\mathcal{A}$ is deterministic, unambiguous, finitely ambiguous, respectively polynomially ambiguous if and only if the same applies to $\mathcal{A}'$ since for every word $w \in \Sigma^*$, we have $\mathrm{Run}_{\mathcal{A}}(w) = \mathrm{Run}_{\mathcal{A}'}(w)$ and $\mathrm{Acc}_{\mathcal{A}}(w) = \mathrm{Acc}_{\mathcal{A}'}(w)$. Likewise, deciding the sequentiality, unambiguity, finite ambiguity, or polynomial ambiguity of a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ can be reduced to deciding the same for the FTA $\mathcal{A}' = (Q, \Gamma, \delta, F)$ with $\delta = \{d \in \Delta_{\mathcal{A}} \mid d \text{ is valid in } \mathcal{A}\}$ and $F = \{q \in Q \mid q \text{ is final in } \mathcal{A}\}$. For NFAs and FTAs, deciding sequentiality is trivial. Polynomial time algorithms to check NFAs and FTAs for unambiguity, finite ambiguity, and polynomial ambiguity can be found in [11, 108, 100].

Remark 4.3. Note that for max-plus automata, the above reduction to finite automata shows in particular that the support of every max-plus word or tree automaton is a recognizable language. For both weighted word and weighted tree automata, this is in fact true for every *zero-sum free* commutative semiring, i.e., every commutative semiring K where $\kappa + \lambda = \mathbb{0}$ implies $\kappa = \lambda = \mathbb{0}$ for every two elements $\kappa, \lambda \in K$ [59, 28].

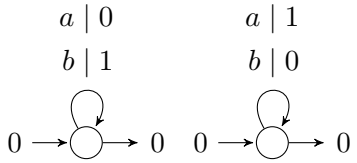
By applying the classical powerset construction [88, 104], we can construct for every NFA $\mathcal{A}$ a deterministic NFA $\mathcal{A}'$ with the same behavior as $\mathcal{A}$. In particular, for every recognizable language L of words there exists a deterministic NFA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = L$. Likewise, every recognizable language of trees is recognizable by a deterministic FTA. In this sense, recognizable languages of words or trees do not carry any inherent degree of ambiguity. The same is in general not true for weighted automata over semirings other than the Boolean semiring. For max-plus automata, the classes of behaviors which can be described by deterministic, unambiguous, finitely ambiguous, polynomially ambiguous, and arbitrary automata form a strictly ascending hierarchy as summarized in the following diagram [63].

$$\mathrm{Seq} \subsetneq \mathrm{Unamb} \cap \mathrm{FinSeq} \begin{array}{c} \subsetneq \\ \subsetneq \end{array} \begin{array}{c} \mathrm{FinSeq} \\ \mathrm{Unamb} \end{array} \begin{array}{c} \subsetneq \\ \subsetneq \end{array} \mathrm{FinAmb} \subsetneq \mathrm{PolyAmb} \subsetneq \mathrm{Rec}$$

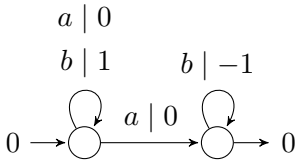
Note that the class of finitely sequential behaviors lies strictly between the classes of behaviors definable by deterministic and by finitely ambiguous max-plus automata, and it is incomparable to the class of behaviors definable by unambiguous max-plus automata. The class Rec (for *recognizable*) denotes all behaviors which can be described by arbitrary max-plus-WA. See also Figure 4.1 for example automata separating the classes above; the complement of a class like Seq is denoted by $\overline{\mathrm{Seq}}$.



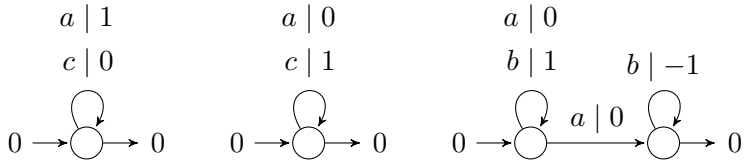
(a) An unambiguous max-plus-WA whose behavior is in $\overline{\text{Seq}} \cap \text{Unamb} \cap \text{FinSeq}$ [63].



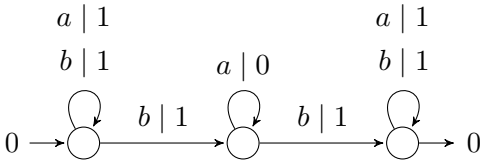
(b) A max-plus-WA whose behavior is in $\overline{\text{Unamb}} \cap \text{FinSeq}$ [63].



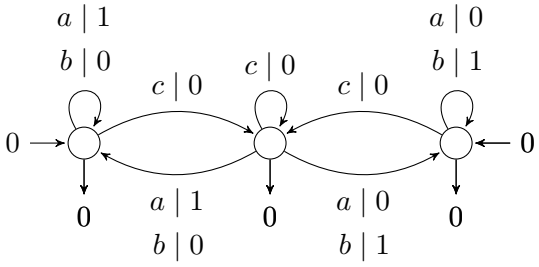
(c) An unambiguous max-plus-WA whose behavior is in $\text{Unamb} \cap \overline{\text{FinSeq}}$ [63].



(d) A finitely ambiguous max-plus-WA whose behavior is in $\overline{\text{Unamb}} \cap \overline{\text{FinSeq}} \cap \text{FinAmb}$ [63].



(e) A polynomially ambiguous max-plus-WA whose behavior is in $\overline{\text{FinAmb}} \cap \text{PolyAmb}$ [58].



(f) A max-plus-WA whose behavior is in $\overline{\text{PolyAmb}} \cap \text{Rec}$ [74].

Figure 4.1: Max-plus word automata which illustrate the ascending hierarchy of behaviors describable by max-plus-WA of a certain degree of ambiguity.

4.2 Decomposing Finitely Ambiguous Max-Plus Tree Automata

For use in Section 4.4, we want to show in this section that every finitely ambiguous max-plus-WTA can be decomposed into a finite pointwise maximum of unambiguous max-plus-WTA such that the supports of all these unambiguous automata coincide. In fact, it is already known that every finitely ambiguous WTA can be decomposed into a finite sum of unambiguous WTA [83, 82]. For the max-plus semiring, this result can be extended quite easily to ensure that all the unambiguous automata have the same support, which for completeness we do in Lemma 4.4. The main objective of this section, however, is to provide an alternative proof for the statement from [83, 82]. The reason to do this is twofold. First, our new proof shows a very tight bound on the number of unambiguous automata needed for the decomposition: we show that if the finitely ambiguous automaton is M-ambiguous, then it can be decomposed into a sum of M unambiguous automata. This bound does not follow from [83, 82]. Second, the new proof relies almost entirely on logics, while the approach in [83, 82] is purely automata-theoretic. We note that the idea for the new proof was suggested by one of the reviewers of the paper [83]. Before we present the new proof, we show that for the max-plus semiring the result from [83, 82] can be extended such that all unambiguous automata have the same support. The idea for this proof is to simply take the automata obtained by [83, 82], unite their supports, and add to each automaton a run of "small" weight for every tree from this union which is not already in the support of the respective automaton.

Lemma 4.4. *Let $\mathcal{A}$ be a finitely ambiguous max-plus-WTA over Γ, then there exist finitely many unambiguous max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_M$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{M} [\![\mathcal{A}_i]\!]$ and $\operatorname{supp}(\mathcal{A}_1) = \ldots = \operatorname{supp}(\mathcal{A}_M)$.*

Proof. By [83, Theorem 1] (or [82, Lemma 7.2]) we can find finitely many unambiguous max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_M$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{M} [\![\mathcal{A}_i]\!]$. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$. We let $L = \bigcup_{i=1}^{M} \operatorname{supp}(\mathcal{A}_i)$ and let κ be the smallest weight used in the automata $\mathcal{A}_1, \dots, \mathcal{A}_M$, i.e., for $R = \bigcup_{i=1}^{M} (\mu_i(\Delta_{\mathcal{A}_i}) \cup \nu_i(Q_i))$ we let $\kappa = \min(R \setminus \{-\infty\})$.

The language L is recognizable, therefore for $i \in \{1, \dots, M\}$, the language $L_i = L \setminus \operatorname{supp}(\mathcal{A}_i)$ is also recognizable and there exists a deterministic FTA $\mathcal{A}'_i = (Q'_i, \Gamma, \delta'_i, F'_i)$ with $\mathcal{L}(\mathcal{A}'_i) = L_i$. We define the max-plus-WTA $\mathcal{A}''_i = (Q'_i, \Gamma, \mu''_i, \nu'')$ by

$$\mu''_i(d) = \begin{cases} \kappa & \text{if } d \in \delta'_i \\ -\infty & \text{otherwise} \end{cases} \qquad \text{and} \qquad \nu''_i(q) = \begin{cases} \kappa & \text{if } q \in F'_i \\ -\infty & \text{otherwise.} \end{cases}$$

We assume without loss of generality that $Q_i \cap Q'_i = \emptyset$ and define $\mathcal{A}'''_i = (Q_i \cup Q'_i, \Gamma, \mu'''_i, \nu_i \cup \nu''_i)$ with

$$\mu'''_i(d) = \begin{cases} \mu_i(d) & \text{if } d \in \Delta_{\mathcal{A}_i} \\ \mu''_i(d) & \text{if } d \in \Delta_{\mathcal{A}''_i} \\ -\infty & \text{otherwise} \end{cases}$$

as the union of $\mathcal{A}_i$ and $\mathcal{A}_i''$. Then $\mathcal{A}_i'''$ is unambiguous since $\mathcal{A}_i$ is unambiguous, $\mathcal{A}_i''$ is deterministic, and $\operatorname{supp}(\mathcal{A}_i) \cap \operatorname{supp}(\mathcal{A}_i'') = \emptyset$. Furthermore, for $t \in \operatorname{supp}(\mathcal{A}_i)$ we have $[\![\mathcal{A}_i''']\!](t) = [\![\mathcal{A}_i]\!](t)$.

For every $t \in \operatorname{supp}(\mathcal{A}_i'')$, there exists some $j \in \{1, \ldots, M\}$ with $t \in \operatorname{supp}(\mathcal{A}_j)$ and due to the choice of κ we have $[\![\mathcal{A}_j]\!](t) \geq [\![\mathcal{A}_i'']\!](t)$. In conclusion, for all $i \in \{1, \ldots, M\}$ we have that $\mathcal{A}_i'''$ is unambiguous, $\operatorname{supp}(\mathcal{A}_i''') = L$, and $\max_{i=1}^{M} [\![\mathcal{A}_i''']\!] = \max_{i=1}^{M} [\![\mathcal{A}_i]\!] = [\![\mathcal{A}]\!]$. □

In the following, we present our new proof for the statement from [83, 82].

Lemma 4.5. *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be an M-ambiguous WTA over a commutative semiring K. Then there exist M unambiguous WTA $\mathcal{A}_1, \ldots, \mathcal{A}_M$ over K and Γ such that $[\![\mathcal{A}]\!] = [\![\mathcal{A}_1]\!] \oplus \ldots \oplus [\![\mathcal{A}_M]\!]$. Over the max-plus semiring, we can choose $\mathcal{A}_1, \ldots, \mathcal{A}_M$ such that $\operatorname{supp}(\mathcal{A}_1) = \ldots = \operatorname{supp}(\mathcal{A}_M)$.*

Proof. We consider the signature $\sigma = (\mathrm{Rel}_\sigma, \mathrm{ar}_\sigma)$ where $\mathrm{Rel}_\sigma = \{\mathrm{label}_a \mid a \in \Gamma\} \cup \{\mathrm{edge}_i \mid i \in \{1, \ldots, \mathrm{rk}(\Gamma)\}\}$, all relations label_a are unary, and all relations edge_i are binary. Every tree $t \in T_\Gamma$ can be translated into a σ-structure $\mathfrak{t}$ with universe $\mathrm{pos}(t)$ as follows. The interpretations for the labels are given by $w \in \mathrm{label}_a^{\mathfrak{t}}$ iff $t(w) = a$, and the interpretations for the edge relations are given by $(v, w) \in \mathrm{edge}_i^{\mathfrak{t}}$ iff $w = vi$. In the following, we will identify t and $\mathfrak{t}$.

We consider the logic $\mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)$ (see Definition 3.17). This particular logic was introduced in [34] and shown to be expressively equivalent to WTA. In [83, 82], various fragments of this logic were investigated and shown to be expressively equivalent to WTA of a certain degree of ambiguity. Most importantly for us, we obtain from [83, Theorem 16] (or [82, Theorem 6.1]) that for every almost Boolean formula $\psi \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$, the formula $\bigotimes x.\psi$ can be translated into an unambiguous WTA over K and Γ. To prove our lemma, we will therefore define M almost Boolean formulas $\psi_1, \ldots, \psi_M$ such that $[\![\mathcal{A}]\!] = [\![\bigotimes x.\psi_1]\!] \oplus \ldots \oplus [\![\bigotimes x.\psi_M]\!]$. By translating each formula $\bigotimes x.\psi_i$ into an unambiguous WTA, we obtain a decomposition of $\mathcal{A}$ into M unambiguous automata as desired.

The idea for the construction of the formulas ψ_i is the following. We can fix an arbitrary linear ordering on the valid transitions of $\mathcal{A}$. This enables us to sort the runs on a tree t lexicographically: for two runs r_1, r_2 on t, we define $r_1 < r_2$ iff for the lexicographically smallest position w at which r_1 and r_2 differ, we have $\mathfrak{t}(t, r_1, w) < \mathfrak{t}(t, r_2, w)$, i.e., the transition of r_1 at w is smaller than the transition of r_2 at w with respect to this arbitrary linear ordering. We can thereby define the formulas ψ_i such that ψ_i "selects" the i-th run, according to the lexicographic ordering, and returns the weight of the transition at position x of this run. In addition, if x is the root, ψ_i returns the weight of the transition at the root multiplied by the final weight of the state at the root.

Formally, we proceed as follows. Let $D = \{d \in \Delta_\mathcal{A} \mid \mu(d) \neq \mathbb{0}\}$ be the set of all valid transitions and $D_F = \{(p_1, \ldots, p_m, a, p) \in D \mid \nu(p) \neq \mathbb{0}\}$ be the set of all valid final transitions. Let $n = |D|$ and let $v \colon D \to \{1, \ldots, n\}$ be a bijection. Then v induces a linear ordering on D by $d_1 < d_2$ iff $v(d_1) < v(d_2)$. For second order variables $X_1, \ldots, X_n$ and a transition $d \in D$, we write X_d for $X_{v(d)}$ and $\bar{X}$ for $(X_1, \ldots, X_n)$.

We begin constructing the formulas ψ_i by defining four abbreviations, namely $x \leq_p y$, $x \leq_l y$, root(x), and partition$(\bar{X})$. The first formula checks whether x is predecessor of y with regard to the prefix ordering, the second formula does the same for the lexicographic ordering, the third formula checks whether x is the root of the tree, and the last formula checks whether $\{X_1, \ldots, X_n\}$ forms a partition of the universe. We define these formulas as follows.

$$(x \leq_p y) = \forall X. \Bigg(\Big(y \in X \wedge \forall z. \Big(\big(\bigvee_{i=1}^{\mathrm{rk}(\Gamma)} (\exists z'. (\mathrm{edge}_i(z, z') \wedge z' \in X)) \big) \rightarrow z \in X \Big) \Big) \rightarrow x \in X \Bigg)$$

$$(x \leq_l y) = (x \leq_p y) \vee \exists z. \exists z_1. \exists z_2. \Big(\bigvee_{1 \leq i < j \leq \mathrm{rk}(\Gamma)} \mathrm{edge}_i(z, z_1) \wedge \mathrm{edge}_j(z, z_2) \wedge z_1 \leq_p x \wedge z_2 \leq_p y \Big)$$

$$\mathrm{root}(x) = \forall y. \left(\bigwedge_{i=1}^{\mathrm{rk}(\Gamma)} \neg \mathrm{edge}_i(y, x) \right)$$

$$\mathrm{partition}(\bar{X}) = \forall x. \bigvee_{i=1}^{n} \left(x \in X_i \wedge \bigwedge_{j \neq i} \neg (x \in X_j) \right).$$

All these formulas are clearly MSO(σ, K)-formulas. We interpret a partition $\bar{X}$ of the nodes of a tree as a run of $\mathcal{A}$ in the sense that "$w \in X_d$" means that at position w, we have transition d. We therefore define the MSO(σ, K)-formula matched$(\bar{X})$ to check whether a guessed partition is a well matched run as follows.

$$\begin{aligned} &\mathrm{matched}(\bar{X}) \\ &= \bigwedge_{(\bar{p}, a, p) \in D} \forall x. \Big((x \in X_{(\bar{p}, a, p)}) \rightarrow \mathrm{label}_a(x) \Big) \wedge && (4.2.1) \\ &\bigwedge_{\substack{(p_1, \ldots, p_m, a, p) \in D \\ m \geq 1}} \forall x. \Bigg((x \in X_{(p_1, \ldots, p_m, a, p)}) \rightarrow \exists y_1 \ldots \exists y_m. \Bigg(\Big(\bigwedge_{i=1}^{m} \mathrm{edge}_i(x, y_i) \Big) \wedge && (4.2.2) \\ &\qquad \bigvee_{\substack{(\bar{q}_1, a_1, p_1) \in D \\ \ldots \\ (\bar{q}_m, a_m, p_m) \in D}} \Big(\bigwedge_{i=1}^{m} (y_i \in X_{(\bar{q}_i, a_i, p_i)}) \Big) \Bigg) \Bigg) \end{aligned}$$

Part (4.2.1) verifies that the labeling of the run is consistent with the letters in the tree and Part (4.2.2) verifies that the transitions used are well matched. With this in hand, we define the MSO(σ, K)-formula accept$_{\mathcal{A}}(\bar{X})$ which checks whether $\bar{X}$ encodes an accepting run of $\mathcal{A}$.

$$\mathrm{accept}_{\mathcal{A}}(\bar{X}) = \mathrm{partition}(\bar{X}) \wedge \mathrm{matched}(\bar{X}) \wedge \exists x. \left(\mathrm{root}(x) \wedge \bigvee_{d \in D_F} (x \in X_d) \right)$$

That is, we verify that $\bar{X}$ is a partition, that this partition is well matched, and that the state at the root is a final state. Next, we define abbreviations to compare runs according to the lexicographic ordering we described earlier.

$$(\bar{X} = \bar{Y}) = \forall x. \bigwedge_{1 \leq i \leq n} (x \in X_i \leftrightarrow x \in Y_i)$$

$$(\bar{X} < \bar{Y}) = \exists x. \Big(\Big(\bigvee_{1 \leq i < j \leq n} x \in X_i \wedge x \in Y_j \Big) \wedge \forall y. \Big((y \leq_l x \wedge \neg(x \leq_l y)) \rightarrow \bigwedge_{1 \leq i \leq n} y \in X_i \leftrightarrow y \in Y_i \Big) \Big)$$

Clearly, both formulas are in $\mathrm{MSO}(\sigma, K)$. We use these two formulas to define for every $k \in \{1, \ldots, M\}$ an $\mathrm{MSO}(\sigma, K)$-formula $\mathrm{accept}^k_{\mathcal{A}}(\bar{X})$ which checks whether $\bar{X}$ is an accepting run and, out of all accepting runs on the given tree, it is the k-th run according to the lexicographic ordering.

$$\mathrm{accept}^k_{\mathcal{A}}(\bar{X}) = \exists \bar{Y}_1 \ldots \exists \bar{Y}_k. \Big((\bar{Y}_k = \bar{X}) \wedge \big(\bigwedge_{1 \leq i \leq k} \mathrm{accept}_{\mathcal{A}}(\bar{Y}_i) \big) \wedge \big(\bigwedge_{1 \leq i < k} \bar{Y}_i < \bar{Y}_{i+1} \big) \wedge \forall \bar{Z}. \big((\mathrm{accept}_{\mathcal{A}}(\bar{Z}) \wedge \bigwedge_{1 \leq i \leq k} \neg(\bar{Y}_i = \bar{Z})) \rightarrow \bar{Y}_k < \bar{Z} \big) \Big)$$

We can now define the formulas ψ_i with the idea we described above. For simplicity, we extend the mapping ν to D by defining $\nu(p_1, \ldots, p_m, a, p) = \nu(p)$. Then for $i \in \{1, \ldots, M\}$, we define ψ_i by

$$\begin{aligned} \psi_i(x) = \quad & \bigoplus\nolimits_{d \in D} \Big(\mu(d) \otimes \nu(d) \otimes \exists \bar{X}. (\mathrm{accept}^i_{\mathcal{A}}(\bar{X}) \wedge x \in X_d \wedge \mathrm{root}(x)) \Big) \\ & \oplus \bigoplus\nolimits_{d \in D} \Big(\mu(d) \otimes \exists \bar{X}. (\mathrm{accept}^i_{\mathcal{A}}(\bar{X}) \wedge x \in X_d \wedge \neg \mathrm{root}(x)) \Big) \end{aligned}$$

Clearly, $\psi_i \in \mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$ for all i, and if for a tree t the run r is the i-th run of $\mathcal{A}$ on t according to the lexicographic ordering, we have $[\![\bigotimes x.\psi_i]\!](t) = \mathrm{wt}_{\mathcal{A}}(t, r) \odot \nu(r(\varepsilon))$.

If K is the max-plus semiring, we can ensure that $\mathrm{supp}(\mathcal{A}_1) = \ldots = \mathrm{supp}(\mathcal{A}_M)$ as follows. To each $\psi_i(x)$, we add a formula which returns $\psi_1(x)$ whenever no i-th run exists. For $i \in \{1, \ldots, M\}$, we define

$$\psi'_i(x) = \psi_i(x) \oplus \Big(\psi_1(x) \otimes \neg \exists \bar{X}. \mathrm{accept}^i_{\mathcal{A}}(\bar{X}) \Big).$$

Then for a tree $t \in T_\Gamma$ with less then i runs, we have $[\![\bigotimes x.\psi'_i]\!](t) = [\![\bigotimes x.\psi_1]\!](t)$ and therefore $\mathrm{supp}(\mathcal{A}_i) = \mathrm{supp}(\mathcal{A}_1) = \mathrm{supp}(\mathcal{A})$. Also, we clearly have $\max_{i=1}^{M} [\![\bigotimes x.\psi'_i]\!] = \max_{i=1}^{M} [\![\bigotimes x.\psi_i]\!] = [\![\mathcal{A}]\!]$. □

4.3 The Equivalence Problem

The equivalence problem for max-plus (tree) automata asks whether for two given max-plus (tree) automata $\mathcal{A}_1$ and $\mathcal{A}_2$, it holds that $[\![\mathcal{A}_1]\!] = [\![\mathcal{A}_2]\!]$. In this case, we say that $\mathcal{A}_1$ and $\mathcal{A}_2$ are *equivalent*. For words, this problem was shown to be undecidable in general [66], but it is decidable if both automata are finitely ambiguous [55]. In this section, we prove that the equivalence problem is decidable for finitely ambiguous max-plus-WTA. Like in [55], we reduce the equivalence problem to the decidability of the existence of an integer solution for a system of linear inequalities [80]. This latter problem is decidable only for systems over the rationals, which is why for the equivalence problem, we consider only max-plus-WTA over the max-plus semiring $\mathbb{Q}_{\max} = (\mathbb{Q} \cup \{-\infty\}, \max, +, -\infty, 0)$ restricted to the rationals. The proof presented here is a revised version of the one from [84]. It is largely based on ideas from [55], but employs Parikh's theorem [81, Theorem 2] instead of the *cycle decompositions* which were used both in [55] and [84]. This idea was suggested by Mikołaj Bojańczyk in a discussion following the presentation of the proof from [84]. We formulate the main result of this section as follows.

Theorem 4.6. *The equivalence problem for finitely ambiguous max-plus tree automata with transition and final weights from $\mathbb{Q} \cup \{-\infty\}$ is decidable.*

In fact, we will show that if $\mathcal{A}_1$ is a finitely ambiguous max-plus-WTA and $\mathcal{A}_2$ any max-plus-WTA, then it is decidable whether for all trees $t \in T_\Gamma$ it holds that $[\![\mathcal{A}_1]\!](t) \geq [\![\mathcal{A}_2]\!](t)$. If this is the case, we also write $[\![\mathcal{A}_1]\!] \geq [\![\mathcal{A}_2]\!]$ and say that $\mathcal{A}_1$ *dominates* $\mathcal{A}_2$.

Theorem 4.7. *Let $\mathcal{A}_1$ be a finitely ambiguous max-plus-WTA and $\mathcal{A}_2$ any max-plus-WTA, both with transition and final weights from $\mathbb{Q} \cup \{-\infty\}$. It is decidable whether or not $[\![\mathcal{A}_1]\!] \geq [\![\mathcal{A}_2]\!]$.*

If both automata in Theorem 4.7 are finitely ambiguous, we can reverse their roles. Consequently, Theorem 4.6 is a corollary of Theorem 4.7. The remainder of this section is dedicated to the proof of Theorem 4.7. As part of the proof, we will employ the following concepts.

Let $\Sigma = \{a_1, \ldots, a_n\}$ be an alphabet. The *Parikh vector* $\mathbb{p}(w) \in \mathbb{N}^n$ of a word $w \in \Sigma^*$ is the vector $\mathbb{p}(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_n})$. For a language $L \subseteq \Sigma^*$, the *Parikh image of L* is the set $\mathbb{p}(L) = \{\mathbb{p}(w) \mid w \in L\}$.

A set of vectors $J \subseteq \mathbb{N}^n$ is called *linear* if there exist $k \geq 0$ and vectors $\alpha, \beta_1, \ldots, \beta_k \in \mathbb{N}^n$ such that

$$J = \{\alpha + \sum_{i=1}^{k} n_i \cdot \beta_i \mid n_1, \ldots, n_k \in \mathbb{N}\}.$$

The set J is called *semilinear* if it is the union of finitely many linear subsets of $\mathbb{N}^n$.

A *context-free grammar* (short: CFG) [49] is a tuple (N, Σ, P, S) where (1) N is a finite set of *nonterminal symbols*, (2) Σ is a finite set of *terminal symbols* with

$N \cap \Sigma = \emptyset$, (3) $P \subseteq N \times (N \cup \Sigma)^*$ is a finite set of *productions* or *rules*, and (4) $S \in N$ is the *initial symbol*. We usually denote a rule $(A, w) \in P$ by $A \to w$.

Let $G = (N, \Sigma, P, S)$ be a context-free grammar. For $u, v \in (N \cup \Sigma)^*$ we write $u \Rightarrow_G v$ if there exists $u', u'' \in (N \cup \Sigma)^*$ and a production $A \to w \in P$ such that $u = u'Au''$ and $v = u'wu''$. The *language generated* by G is the language

$$\mathcal{L}(G) = \{w \in \Sigma^* \mid \exists n \geq 0 \exists u_1, \ldots, u_n \in (N \cup \Sigma)^* \colon S \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n \Rightarrow_G w\}.$$

A language $L \subseteq \Sigma^*$ is called *context-free* if there exists a context-free grammar G with $L = \mathcal{L}(G)$.

As a first step, we show in the following lemma that every finitely ambiguous max-plus-WTA $\mathcal{A}$ can be normalized such that all trees, on which there exists at least one accepting run of $\mathcal{A}$, have the same number of accepting runs. The idea here is that we can simply add dummy runs with low weight for every tree which does not already have a sufficient number of runs.

Lemma 4.8. *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be an M-ambiguous max-plus-WTA. Then there exists a finitely ambiguous max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ and $|\mathrm{Acc}_{\mathcal{A}'}(t)| \in \{0, M\}$ for all $t \in T_\Gamma$.*

Proof. First, we show that for every $n \in \{1, \ldots, M\}$, the set $L_n = \{t \in T_\Gamma \mid |\mathrm{Acc}_{\mathcal{A}}(t)| \geq n\}$ is recognizable. For this, we construct an automaton which simulates n runs of $\mathcal{A}$ in parallel, keeps track of which runs are pairwise distinct, and accepts only when all simulated runs are pairwise distinct. Let $\mathcal{A}_n = (Q^n \times \mathcal{P}(\{1, \ldots, n\}^2), \Gamma, \delta_n, F_n)$ be the FTA defined as follows. For $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$, $\bar{p}_0, \ldots, \bar{p}_m \in Q^n$ with $\bar{p}_i = (p_{i1}, \ldots, p_{in})$, and $R_0, \ldots, R_m \subseteq \{1, \ldots, n\}^2$, we let $((\bar{p}_1, R_1), \ldots, (\bar{p}_m, R_m), a, (\bar{p}_0, R_0)) \in \delta_n$ iff for all $i \in \{1, \ldots, n\}$ we have $\mu(p_{1i}, \ldots, p_{mi}, a, p_{0i}) \neq -\infty$ and $R_0 = \{(k, l) \in \{1, \ldots, n\}^2 \mid p_{0k} \neq p_{0l}\} \cup \bigcup_{i=1}^m R_i$. Furthermore, $(\bar{p}_0, R_0) \in F_n$ iff for all $i \in \{1, \ldots, n\}$ we have $\nu(p_{0i}) \neq -\infty$ and $R_0 = \{(k, l) \in \{1, \ldots, n\}^2 \mid k \neq l\}$.

It is easy to see that there is an accepting run of $\mathcal{A}_n$ on $t \in T_\Gamma$ if and only if there are at least n pairwise distinct accepting runs of $\mathcal{A}$ on t. Therefore, $\mathcal{L}(\mathcal{A}_n) = L_n$. Since recognizable tree languages are closed under complement and intersection, for $n \in \{1, \ldots, M-1\}$ the languages $L'_n = L_n \setminus L_{n+1} = \{t \in T_\Gamma \mid |\mathrm{Acc}_{\mathcal{A}}(t)| = n\}$ are also recognizable and we can find deterministic FTA $\mathcal{A}'_n = (Q'_n, \Gamma, \delta'_n, F'_n)$ with $\mathcal{L}(\mathcal{A}'_n) = L'_n$.

Now let κ be the smallest weight used in $\mathcal{A}$, i.e., with $R = \mu(\Delta_\mathcal{A}) \cup \nu(Q)$ we let $\kappa = \min(R \setminus \{-\infty\})$. For $n \in \{1, \ldots, M-1\}$, we define the max-plus-WTA $\mathcal{A}''_n = (Q'_n, \Gamma, \mu''_n, \nu''_n)$ by

$$\mu''_n(d) = \begin{cases} \kappa & \text{if } d \in \delta'_n \\ -\infty & \text{otherwise} \end{cases} \qquad \text{and} \qquad \nu''_n(q) = \begin{cases} \kappa & \text{if } q \in F'_n \\ -\infty & \text{otherwise.} \end{cases}$$

Finally, we construct $\mathcal{A}'$ as follows. For each $n \in \{1, \ldots, M-1\}$, we take $M - n$ copies of $\mathcal{A}''_n$ and unite them with $\mathcal{A}$, where we assume that all sets of states are pairwise disjoint. By choice of κ, this does not influence the behavior of $\mathcal{A}$. By choice

of the languages L''_n, every tree which had at least one accepting run in $\mathcal{A}$ now has exactly M accepting runs in $\mathcal{A}'$ and all other trees still have no accepting run in $\mathcal{A}'$. □

Next, we show that every max-plus-WTA $\mathcal{A}$ can be normalized such that all final weights are equal either to $-\infty$ or to 0. The idea is that the final weight can be included in the transition weight of the transition at the root, see also [15].

Lemma 4.9 ([15])**.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Then there exists a max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$, $\nu'(Q') \subseteq \{-\infty, 0\}$, and $|\mathrm{Acc}_{\mathcal{A}}(t)| = |\mathrm{Acc}_{\mathcal{A}'}(t)|$ for every $t \in T_\Gamma$.*

Proof. We define a max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ as follows. We let $Q' = Q \times \{0, 1\}$ and define $\nu'(q, 0) = -\infty$ and $\nu'(q, 1) = 0$ for all $q \in Q$. For every $d = (p_1, \ldots, p_m, a, p_0) \in \Delta_{\mathcal{A}}$, we let $\mu'((p_1, 0), \ldots, (p_m, 0), a, (p_0, 0)) = \mu(d)$ and $\mu'((p_1, 0), \ldots, (p_m, 0), a, (p_0, 1)) = \mu(d) + \nu(p_0)$. On all remaining transitions we define μ' as $-\infty$.

It is easy to see that for every tree $t \in T_\Gamma$, we have a bijection $f \colon \mathrm{Acc}_{\mathcal{A}}(t) \to \mathrm{Acc}_{\mathcal{A}'}(t)$ given by $(f(r))(\varepsilon) = (r(\varepsilon), 1)$ and $(f(r))(w) = (r(w), 0)$ for $w \in \mathrm{pos}(t) \setminus \{\varepsilon\}$, and for this bijection it holds that $\mathrm{wt}_{\mathcal{A}}(t, r) + \nu(r(\varepsilon)) = \mathrm{wt}_{\mathcal{A}'}(t, f(r))$. □

For the rest of this section, we fix an M-ambiguous max-plus-WTA $\mathcal{A}_1$ and a max-plus-WTA $\mathcal{A}_2$, both with transition and final weights from $\mathbb{Q} \cup \{-\infty\}$. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i = 1, 2$. By Lemma 4.8, we can assume that for all $t \in T_\Gamma$ we have $|\mathrm{Acc}_{\mathcal{A}_1}(t)| \in \{0, M\}$. By Lemma 4.9, we may furthermore assume that $\nu_1(Q_1) \subseteq \{-\infty, 0\}$ and $\nu_2(Q_2) \subseteq \{-\infty, 0\}$. Note that $[\![\mathcal{A}_1]\!] \geq [\![\mathcal{A}_2]\!]$ can only hold if $\mathrm{supp}(\mathcal{A}_2) \subseteq \mathrm{supp}(\mathcal{A}_1)$, which is decidable since the supports of $\mathcal{A}_1$ and $\mathcal{A}_2$ are recognizable tree languages. Therefore, in the forthcoming considerations we will assume that $\mathrm{supp}(\mathcal{A}_2) \subseteq \mathrm{supp}(\mathcal{A}_1)$ holds.

We call a tuple $\bar{v} \in \mathbb{Q}^{M+1}$ an *outcome vector* if there exists a tree $t \in T_\Gamma$, runs $r_1, \ldots, r_M \in \mathrm{Acc}_{\mathcal{A}_1}(t)$, and a run $r_{M+1} \in \mathrm{Acc}_{\mathcal{A}_2}(t)$ with $\mathrm{Acc}_{\mathcal{A}_1}(t) = \{r_1, \ldots, r_M\}$ and $\bar{v} = (\mathrm{wt}_{\mathcal{A}_1}(t, r_1), \ldots, \mathrm{wt}_{\mathcal{A}_1}(t, r_M), \mathrm{wt}_{\mathcal{A}_2}(t, r_{M+1}))$. We denote the set of all outcome vectors by $\mathbb{O}$. We can make the following observation.

Proposition 4.10. *$\mathcal{A}_1$ does not dominate $\mathcal{A}_2$ iff there exists a vector $(v_1, \ldots, v_{M+1}) \in \mathbb{O}$ such that for all $i \in \{1, \ldots, M\}$ we have $v_i < v_{M+1}$.*

We give an overview of the rest of the proof. We first construct a weighted tree automaton $\mathcal{A}$ over the product semiring $(\mathbb{Q}_{\max})^{M+1}$ such that the weights realized by the runs of $\mathcal{A}$ are exactly the vectors from $\mathbb{O}$. We then define Parikh vectors of runs by counting the transitions occurring in a run, just like the Parikh vector of a word counts the number of occurrences of the letters in a word. By arranging the weight vectors of the transitions of $\mathcal{A}$ as columns into a matrix Ω, we see that the weight of a run of $\mathcal{A}$ is simply the result of multiplying the matrix Ω with the Parikh vector of the run.

We proceed to show that the set of Parikh vectors of the accepting runs of $\mathcal{A}$ can also be expressed as the Parikh image of a context free language over the alphabet of transitions from $\Delta_{\mathcal{A}}$. By Parikh's theorem, the Parikh image of a context-free

language is semilinear, and thus so is the set of Parikh vectors of the accepting runs of $\mathcal{A}$.

It follows that the set $\mathbb{O}$ can be represented as the image of a semilinear set, namely the set of Parikh vectors of the accepting runs of $\mathcal{A}$, under a matrix with rational entries, namely the matrix Ω. We then use Proposition 4.10 to reduce the dominance problem to the satisfiability problem of systems of linear inequalities over the rationals with an integer solution. The latter problem is decidable [13, Theorem 3.4]. We begin by constructing $\mathcal{A}$.

Lemma 4.11. *There exists a weighted tree automaton $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ over the product semiring $(\mathbb{Q}_{\max})^{M+1}$ such that $\mathbb{O} = \{\mathrm{wt}_{\mathcal{A}}(t, r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_{\mathcal{A}}(t)\}$. The automaton $\mathcal{A}$ can be effectively constructed from $\mathcal{A}_1$ and $\mathcal{A}_2$.*

Proof. We let $Q = Q_1^M \times Q_2 \times \mathcal{P}(\{1, \dots, M\}^2)$. The first $M + 1$ entries of the states from Q are used to simulate the M runs of $\mathcal{A}_1$ and one run of $\mathcal{A}_2$, and the last entry is used to keep a record of which runs from $\mathcal{A}_1$ are distinct in order to ensure that accepting runs of $\mathcal{A}$ simulate all accepting runs of $\mathcal{A}_1$ in the respective entries. For $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and $\mathbf{p}_0, \dots, \mathbf{p}_m \in Q$ with $\mathbf{p}_i = (p_{i1}, \dots, p_{iM}, p_{iM+1}, R_i)$, we define weights as follows. For $i \in \{1, \dots, M\}$, we let $x_i = \mu_1(p_{1i}, \dots, p_{mi}, a, p_{0i})$ and $y_i = \nu_1(p_{0i})$, and we let $x_{M+1} = \mu_2(p_{1M+1}, \dots, p_{mM+1}, a, p_{0M+1})$ and $y_{M+1} = \nu_2(p_{0M+1})$. Furthermore, we let $R = \{(k, l) \in \{1, \dots, M\}^2 \mid p_{0k} \neq p_{0l}\} \cup \bigcup_{i=1}^{M} R_i$. Then we define μ and ν by

$$\mu(\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}_0) = \begin{cases} (x_1, \dots, x_{M+1}) & \text{if } (x_1, \dots, x_{M+1}) \in \mathbb{Q}^{M+1} \text{ and } R_0 = R \\ (-\infty, \dots, -\infty) & \text{otherwise} \end{cases}$$

$$\nu(\mathbf{p}_0) = \begin{cases} (y_1, \dots, y_{M+1}) & \text{if } (y_1, \dots, y_{M+1}) \in \mathbb{Q}^{M+1} \text{ and} \\ & R_0 = \{(k, l) \in \{1, \dots, M\}^2 \mid k \neq l\} \\ (-\infty, \dots, -\infty) & \text{otherwise.} \end{cases}$$

It is easy to see that for an accepting run of $\mathcal{A}$ on a tree t, projecting on each of the first $M + 1$ entries yields M distinct accepting runs of $\mathcal{A}_1$ and one accepting run of $\mathcal{A}_2$ on t, and that the transition weights are preserved by this projection.

Furthermore, for M pairwise distinct accepting runs $r_1, \dots, r_M$ of $\mathcal{A}_1$ and one accepting run r_{M+1} of $\mathcal{A}_2$ on a tree t, we can construct a mapping $R\colon \mathrm{pos}(t) \to \mathcal{P}(\{1, \dots, M\}^2)$ such that $(r_1, \dots, r_{M+1}, R)$ is an accepting run of $\mathcal{A}$ on t with $\mathrm{wt}_{\mathcal{A}}(t, (r_1, \dots, r_{M+1}, R)) = (\mathrm{wt}_{\mathcal{A}_1}(t, r_1), \dots, \mathrm{wt}_{\mathcal{A}_1}(t, r_M), \mathrm{wt}_{\mathcal{A}_2}(t, r_{M+1}))$. □

Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be the automaton from Lemma 4.11 and let $d_1, \dots, d_D$ be an enumeration of $\Delta_{\mathcal{A}}$. We define a matrix $\Omega \in \mathbb{Q}^{(M+1) \times D}$ by $\Omega = (\mu(d_1), \dots, \mu(d_D))$ where every vector $\mu(d_i)$ is considered to be a column vector. Furthermore, for a run r of $\mathcal{A}$ on a tree t, we define the *transition Parikh vector* of r by

$$\mathbb{p}(t, r) = (|\{w \in \mathrm{pos}(t) \mid \mathbb{t}(t, r, w) = d_1\}|, \dots, |\{w \in \mathrm{pos}(t) \mid \mathbb{t}(t, r, w) = d_D\}|).$$

In the following Lemma, we show that multiplying Ω with every possible transition Parikh vector of $\mathcal{A}$ yields precisely $\mathbb{O}$.

Lemma 4.12. *We have* $\mathbb{O} = \{\Omega \cdot \mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_\mathcal{A}(t)\}$.

Proof. Let $\bar{v} \in \mathbb{O}$, then by assumption on $\mathcal{A}$, there exists a tree $t \in T_\Gamma$ and a run $r \in \mathrm{Acc}_\mathcal{A}(t)$ with $\bar{v} = \mathrm{wt}_\mathcal{A}(t,r)$. By definition of $\mathrm{wt}_\mathcal{A}$ and the commutativity of "+", it follows that $\mathrm{wt}_\mathcal{A}(t,r) = \Omega \cdot \mathbb{p}(t,r)$.

On the other hand, let $t \in T_\Gamma$ and $r \in \mathrm{Acc}_\mathcal{A}(t)$. Then with the same arguments and our assumption on $\mathcal{A}$, we have $\Omega \cdot \mathbb{p}(t,r) = \mathrm{wt}_\mathcal{A}(t,r) \in \mathbb{O}$. □

Next, we construct a context-free language whose Parikh image coincides with the set of possible transition Parikh vectors of $\mathcal{A}$.

Lemma 4.13. *There exists a context-free language L over the alphabet $\Delta_\mathcal{A}$ such that $\mathbb{p}(L) = \{\mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_\mathcal{A}(t)\}$. A context-free grammar G generating L can be found effectively from $\mathcal{A}$.*

Proof. We define the context-free grammar $G = (Q \cup \{S\}, \Delta_\mathcal{A}, P, S)$, where S is a new symbol, by

$$\begin{aligned} P = \quad & \{S \to \mathbf{p} \mid \nu(\mathbf{p}) \in \mathbb{Q}^{M+1}\} \\ & \cup \{\mathbf{p} \to (\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p})\mathbf{p}_1 \dots \mathbf{p}_m \mid \mu(\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}) \in \mathbb{Q}^{M+1}\}. \end{aligned}$$

Then $L = \mathcal{L}(G)$ is context-free and we see as follows that $\mathbb{p}(L) = \{\mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_\mathcal{A}(t)\}$.

"$\subseteq$": Let $w \in L$. We construct a tree $t \in T_\Gamma$ and a run $r \in \mathrm{Acc}_\mathcal{A}(t)$ such that $\mathbb{p}(w) = \mathbb{p}(t,r)$. Since $w \in L$, we find words $u_1, \dots, u_n \in (Q \cup \Delta_\mathcal{A})^*$ such that $u_n = w$ and $S \Rightarrow_G u_1 \Rightarrow_G \dots \Rightarrow_G u_n$. We construct by induction for every $i \in \{1, \dots, n\}$ a Γ-context $t_i \in T_{\Gamma_\diamond}$ and a run $r_i \in \mathrm{Run}^\diamond_\mathcal{A}(t_i)$ such that $\nu(r_i(\varepsilon)) \in \mathbb{Q}^{M+1}$ and for every $\mathbf{p} \in Q$ and $d \in \Delta_\mathcal{A}$ we have

$$\begin{aligned} |u_i|_\mathbf{p} &= |\{v \in \mathrm{pos}(t) \mid t_i(v) = \diamond \text{ and } r_i(v) = \mathbf{p}\}| \\ |u_i|_d &= |\{v \in \mathrm{pos}(t) \mid \mathbb{t}(t,r,w) = d\}|. \end{aligned}$$

For $i = 1$, we know by the definition of G that $u_1 = \mathbf{p}$ with $\nu(\mathbf{p}) \in \mathbb{Q}^{M+1}$, so we let $t_1 = \diamond$ and $r_1(\varepsilon) = \mathbf{p}$. Now assume we have constructed t_i and r_i with the properties above. We have $u_i \Rightarrow_G u_{i+1}$, so by definition of G, there exists a transition $d = (\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}) \in \Delta_\mathcal{A}$ with $\mu(d) \in \mathbb{Q}^{M+1}$ and words $u', u'' \in (Q \cup \Delta_\mathcal{A})^*$ such that $u_i = u'\mathbf{p}u''$ and $u_{i+1} = u'd\mathbf{p}_1 \dots \mathbf{p}_m u''$. Thus $|u_i|_\mathbf{p} \geq 1$, so by induction we find $v \in \mathrm{pos}(t_i)$ with $t_i(v) = \diamond$ and $r_i(v) = \mathbf{p}$. We let $t_{i+1} = t_i\langle a(\diamond, \dots, \diamond) \to v\rangle$ and define r_{i+1} by $r_{i+1}(v') = r_i(v')$ for $v' \in \mathrm{pos}(t_i)$ and $r_{i+1}(vj) = \mathbf{p}_j$ for $j \in \{1, \dots, m\}$. It is easy to check that t_{i+1} and r_{i+1} satisfy all of the above properties.

Since $u_n = w \in \Delta^*_\mathcal{A}$, the Γ-context t_n is actually a Γ-tree, the run $r_n \in \mathrm{Run}^\diamond_\mathcal{A}(t_n)$ is an accepting run of $\mathcal{A}$ on t_n, and we have $\mathbb{p}(w) = \mathbb{p}(u_n) = \mathbb{p}(t_n, r_n)$. Thus, we have $\mathbb{p}(L) \subseteq \{\mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_\mathcal{A}(t)\}$.

"$\supseteq$": Now let $t \in T_\Gamma$ and $r \in \mathrm{Acc}_\mathcal{A}(t)$. We construct a word $w \in L$ with $\mathbb{p}(w) = \mathbb{p}(t,r)$. For this, we construct by induction for every $v \in \mathrm{pos}(t)$ words $u_1, \dots, u_n$ such that $r(v) \Rightarrow_G u_1 \Rightarrow_G \dots \Rightarrow_G u_n$, $u_n \in \Delta^*_\mathcal{A}$, and $\mathbb{p}(u_n) = \mathbb{p}(t|_v, r|_v)$. We proceed by a reverse induction on the length of v. For $|v| = \mathrm{height}(t)$, we let $n = 1$ and $u_1 = \mathbb{t}(t,r,v)$, then we have $r(v) \Rightarrow_G u_1$, $u_n \in \Delta^*_\mathcal{A}$, and $\mathbb{p}(u_n) = \mathbb{p}(t|_v, r|_v)$.

For $|v| < \mathrm{height}(t)$, we assume that $\mathbb{t}(t,r,v) = d = (\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p})$ and that for every $i \in \{1,\ldots,m\}$ we have words $u_1^{(i)}, \ldots, u_{n_i}^{(i)}$ with $\mathbf{p}_i \Rightarrow_G u_1^{(i)} \Rightarrow_G \ldots \Rightarrow_G u_{n_i}^{(i)}$, $u_{n_i}^{(i)} \in \Delta_{\mathcal{A}}^*$, and $\mathbb{p}(u_{n_i}^{(i)}) = \mathbb{p}(t|_{vi}, r|_{vi})$. Since $r \in \mathrm{Acc}_{\mathcal{A}}(t)$, we have $\mu(d) \in \mathbb{Q}^{M+1}$, so by the definition of G, we have $\mathbf{p} \Rightarrow_G d\mathbf{p}_1 \ldots \mathbf{p}_m$. Thus, we see that

$$\begin{aligned}
\mathbf{p} &\Rightarrow_G d\mathbf{p}_1 \ldots \mathbf{p}_m \\
&\Rightarrow_G du_1^{(1)}\mathbf{p}_2 \ldots \mathbf{p}_m \Rightarrow_G \ldots \Rightarrow_G du_{n_1}^{(1)}\mathbf{p}_2 \ldots \mathbf{p}_m \\
&\Rightarrow_G du_{n_1}^{(1)}u_1^{(2)}\mathbf{p}_3 \ldots \mathbf{p}_m \Rightarrow_G \ldots \Rightarrow_G du_{n_1}^{(1)}u_{n_2}^{(2)}\mathbf{p}_3 \ldots \mathbf{p}_m \\
&\;\;\vdots \\
&\Rightarrow_G du_{n_1}^{(1)} \ldots u_{m-1}^{(m-1)}u_1^{(m)} \Rightarrow_G \ldots \Rightarrow_G du_{n_1}^{(1)} \ldots u_{n_m}^{(m)}.
\end{aligned}$$

From this, we obtain words $u_1, \ldots, u_n \in (Q \cup \Delta_{\mathcal{A}})^*$ with $\mathbf{p} \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n$ such that $u_n = du_{n_1}^{(1)} \ldots u_{n_m}^{(m)} \in \Delta_{\mathcal{A}}^*$, and therefore $\mathbb{p}(u_n) = \mathbb{p}(d) + \sum_{i=1}^m \mathbb{p}(u_{n_i}^{(i)}) = \mathbb{p}(d) + \sum_{i=1}^m \mathbb{p}(t|_{vi}, r|_{vi}) = \mathbb{p}(t|_v, r|_v)$.

For $v = \varepsilon$, we thus obtain words $u_1, \ldots, u_n$ such that $r(\varepsilon) \Rightarrow_G u_1 \Rightarrow_G \ldots \Rightarrow_G u_n$, $u_n \in \Delta_{\mathcal{A}}^*$, and $\mathbb{p}(u_n) = \mathbb{p}(t,r)$. Due to $r \in \mathrm{Acc}_{\mathcal{A}}(t)$ we have $r(\varepsilon) \in \mathbb{Q}^{M+1}$, which means that $S \Rightarrow_G r(\varepsilon)$. Therefore $u_n \in L$, which shows that $\mathbb{p}(L) \supseteq \{\mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_{\mathcal{A}}(t)\}$. □

Finally, we recall Parikh's theorem, after which we are ready to conclude the proof of Theorem 4.7.

Theorem 4.14 ([81, Theorem 2],[43])**.** *For every context-free language L, the set $\mathbb{p}(L)$ is semilinear. Furthermore, indices $k, k_1, \ldots, k_k$ and vectors $\alpha^{(i)}, \beta_j^{(i)} \in \mathbb{N}^D$ $(i \in \{1,\ldots,k\}, j \in \{1,\ldots,k_i\})$ with*

$$\mathbb{p}(L) = \bigcup_{i=1}^{k} \{\alpha^{(i)} + \sum_{j=1}^{k_i} n_j \cdot \beta_j^{(i)} \mid n_1, \ldots, n_{k_i} \in \mathbb{N}\}$$

can be effectively found from every context-free grammar generating L.

Proof of Theorem 4.7. Let L be as in Lemma 4.13. By Lemma 4.12 and Lemma 4.13, we then have $\mathbb{O} = \{\Omega \cdot \mathbb{p}(t,r) \mid t \in T_\Gamma, r \in \mathrm{Acc}_{\mathcal{A}}(t)\} = \{\Omega \cdot \bar{v} \mid \bar{v} \in \mathbb{p}(L)\}$.

For L, let $k, k_1, \ldots, k_k, \alpha^{(i)}, \beta_j^{(i)} \in \mathbb{N}^D$ $(i \in \{1,\ldots,k\}, j \in \{1,\ldots,k_i\})$ be as in Theorem 4.14. Then

$$\mathbb{O} = \bigcup_{i=1}^{k} \{\Omega \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} n_j \cdot \Omega \cdot \beta_j^{(i)} \mid n_1, \ldots, n_{k_i} \in \mathbb{N}\}.$$

Let $\bar{\omega}_1, \ldots, \bar{\omega}_{M+1}$ be the rows of Ω. Then by Proposition 4.10, $\mathcal{A}_1$ does not dominate $\mathcal{A}_2$ iff there exist $i \in \{1,\ldots,k\}$ and $n_1, \ldots, n_{k_i} \in \mathbb{N}$ such that for every $l \in \{1,\ldots,M\}$ we have

$$\bar{\omega}_l \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_l \cdot \beta_j^{(i)}) \cdot n_j < \bar{\omega}_{M+1} \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_{M+1} \cdot \beta_j^{(i)}) \cdot n_j.$$

In other words, for every $i \in \{1, \ldots, k\}$ we have a system of linear inequalities

$$\bar{\omega}_l \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_l \cdot \beta_j^{(i)}) \cdot X_j < \bar{\omega}_{M+1} \cdot \alpha^{(i)} + \sum_{j=1}^{k_i} (\bar{\omega}_{M+1} \cdot \beta_j^{(i)}) \cdot X_j \quad (l = 1, \ldots, M)$$

$$0 \leq X_j \quad (j = 1, \ldots, k_i)$$

and $\mathcal{A}_1$ does not dominate $\mathcal{A}_2$ iff one of these systems possesses an integer solution. The first M inequalities of each system form a system of the form $A'\bar{X} < 0$ for a matrix A'. The satisfiability of this system with a non-negative integer solution is equivalent to that of the system $A'\bar{X} \leq -1$ since every non-negative integer solution $\bar{X}$ of the first can be inflated by a sufficiently large integer C to a solution $C \cdot \bar{X}$ of the latter system. Thus, we effectively need to check the satisfiability of systems of the form $A\bar{X} \leq \bar{b}$ for a matrix A and a vector $\bar{b}$, both with entries from $\mathbb{Q}$, with an integer solution. By [13, Theorem 3.4], the satisfiability of such systems with an integer solution is decidable, so we can decide whether $\mathcal{A}_1$ dominates $\mathcal{A}_2$ or not. □

4.4 The Unambiguity Problem

> A man with a watch knows what time it is.
> A man with two watches is never sure.
>
> *Segal's Law*

The unambiguity problem asks whether for a given max-plus-WTA $\mathcal{A}$, there exists an unambiguous max-plus-WTA $\mathcal{A}'$ such that $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. In this section, we show that the unambiguity problem is decidable for finitely ambiguous max-plus-WTA. We follow ideas from [63, Section 5], where the decidability of this problem was shown for finitely ambiguous max-plus word automata. We first generalize the *dominance property* from max-plus word automata to max-plus tree automata and show that it is decidable whether a max-plus tree automaton satisfies the dominance property. Then we show that for a finitely ambiguous max-plus tree automaton $\mathcal{A}$, there exists an equivalent unambiguous max-plus tree automaton $\mathcal{A}'$ if and only if $\mathcal{A}$ satisfies the dominance property. We note that for max-plus word automata, the unambiguity problem is known to be decidable even for polynomially ambiguous automata [61]. We leave the question open as to whether the same holds true for polynomially ambiguous max-plus-WTA. The main result of this section is the following theorem.

Theorem 4.15. *For a finitely ambiguous max-plus-WTA $\mathcal{A}$, it is decidable whether there exists an unambiguous max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. If $\mathcal{A}'$ exists, it can be effectively constructed.*

The rest of this section is dedicated to the proof of Theorem 4.15. For the proof, we will employ the concept of an $\mathcal{A}$-*circuit* of a WTA $\mathcal{A}$ defined as follows. For a WTA $\mathcal{A} = (Q, \Gamma, \mu, \nu)$, a Γ-word $s \in T_{\Gamma_\diamond}$, and a run $r \in \mathrm{Run}^{\diamond}_{\mathcal{A}}(s)$ with $r(\varepsilon) = r(\Diamond_1(s))$, the pair (s, r) is called an $\mathcal{A}$-*circuit*. We call (s, r) *small* if $\mathrm{height}(s) \leq 2|Q|$.

For the rest of this section, we fix a finitely ambiguous max-plus-WTA $\mathcal{A}$. We apply Lemma 4.4 (or equivalently Lemma 4.5) and obtain unambiguous max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_M$ such that $\mathrm{supp}(\mathcal{A}_1) = \ldots = \mathrm{supp}(\mathcal{A}_M)$ and $[\![\mathcal{A}]\!] = \max_{i=1}^{M} [\![\mathcal{A}_i]\!]$. The *product automaton* $\mathcal{B} = (Q, \Gamma, \mu, \nu)$ *of* $\mathcal{A}_1, \ldots, \mathcal{A}_M$ is a weighted tree automaton over the product semiring $(\mathbb{R}_{\max})^M$ which, intuitively, executes all of the automata $\mathcal{A}_1, \ldots, \mathcal{A}_M$ in parallel. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i \in \{1, \ldots, M\}$ and define $\mathcal{B}$ as the trim part of the automaton $\mathcal{B}' = (Q', \Gamma, \mu', \nu')$ defined as follows. We let $Q' = Q_1 \times \ldots \times Q_M$ and for $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and $\mathbf{p}_0, \ldots, \mathbf{p}_m \in Q'$ with $\mathbf{p}_i = (p_{i1}, \ldots, p_{iM})$ we define, with $x_i = \mu_i(p_{1i}, \ldots, p_{mi}, a, p_{0i})$ and $y_i = \nu_i(p_{0i})$,

$$\mu'(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}_0) = \begin{cases} (x_1, \ldots, x_M) & \text{if } (x_1, \ldots, x_M) \in \mathbb{R}^M \\ (-\infty, \ldots, -\infty) & \text{otherwise} \end{cases}$$

$$\nu'(\mathbf{p}_0) = \begin{cases} (y_1, \ldots, y_M) & \text{if } (y_1, \ldots, y_M) \in \mathbb{R}^M \\ (-\infty, \ldots, -\infty) & \text{otherwise.} \end{cases}$$

Then $\mathcal{B}$ is unambiguous and for $t \in T_\Gamma$ we have $[\![\mathcal{B}]\!](t) = ([\![\mathcal{A}_1]\!](t), \ldots, [\![\mathcal{A}_M]\!](t))$.

Definition 4.16 (Victorious coordinate)**.** Let $s \in T_{\Gamma_\diamond}$ be a Γ-context, $r \in \mathrm{Run}^\diamond_{\mathcal{B}}(s)$, and write $\mathrm{wt}^\diamond_{\mathcal{B}}(s,r) = (\kappa_1, \ldots, \kappa_M)$. We define $\mathrm{wt}_i(s,r) = \kappa_i$ and $\mathrm{wt}(s,r) = \max_{i=1}^M \mathrm{wt}_i(s,r)$.

A coordinate $i \in \{1, \ldots, M\}$ is called *victorious* if $\mathrm{wt}_i(s,r) = \mathrm{wt}(s,r)$. The set of all victorious coordinates of (s,r) is denoted by $\mathrm{Vict}(s,r)$. For $\mathbf{q} \in Q$ we define

$$\mathrm{Vict}([\mathbf{q}]) = \bigcap_{\substack{(s,r) \text{ small } \mathcal{B}\text{-circuit} \\ r(\varepsilon) \in [\mathbf{q}]}} \mathrm{Vict}(s,r)$$

where the empty intersection is defined as $\{1, \ldots, M\}$. For $P \subseteq Q$, we let $\mathrm{Vict}(P) = \bigcap_{\mathbf{p} \in P} \mathrm{Vict}([\mathbf{p}])$. We have the following lemma which relates victorious coordinates to the decidability of the unambiguity problem.

Lemma 4.17. *There exists an unambiguous max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ if and only if for all $t \in T_\Gamma$ and all $r \in \mathrm{Acc}_{\mathcal{B}}(t)$ we have $\mathrm{Vict}(r(\mathrm{pos}(t))) \neq \emptyset$. The latter property is called the* dominance property *and is denoted by* (**P**)*. The dominance property is decidable, and therefore so is the unambiguity problem.*

Proof. Here, we only show that (**P**) is decidable. We defer the proof that (**P**) is a necessary condition to Lemma 4.18. The proof for the sufficiency of (**P**) takes some more preparation and is split into several lemmata.

(**P**) is decidable as follows. We can consider Q as an (unranked) alphabet and construct an FTA which accepts exactly the accepting runs of $\mathcal{B}$, i.e., all pairs $(\mathrm{pos}(t), r)$ for some $t \in T_\Gamma$ and $r \in \mathrm{Acc}_{\mathcal{B}}(t)$. Also, for every subset $P \subseteq Q$ we can construct an FTA which accepts all trees in T_Q in which every $p \in P$ occurs at least once as a label. By taking the intersection of these two automata and checking for emptiness, we can decide for every $P \subseteq Q$ whether there exists $t \in T_\Gamma$ and $r \in \mathrm{Acc}_{\mathcal{B}}(t)$ with $P \subseteq r(\mathrm{pos}(t))$. Checking whether all P for which this is true satisfy $\mathrm{Vict}(P) \neq \emptyset$ is equivalent to checking (**P**). Note that $\mathrm{Vict}(P)$ can be effectively computed since there are only finitely many small $\mathcal{B}$-circuits. □

First, we prove that (**P**) is a necessary condition, i.e., that from the existence of an unambiguous automaton $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ it follows that $\mathcal{B}$ satisfies (**P**).

Lemma 4.18. *If there exists an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ then $\mathcal{B}$ satisfies* (**P**)*.*

Proof. We proceed by contradiction and assume that $\mathcal{A}'$ as above exists and that (**P**) is not satisfied. Then there exists a tree $t \in T_\Gamma$ and a run $r \in \mathrm{Acc}_{\mathcal{B}}(t)$ with $\mathrm{Vict}(r(\mathrm{pos}(t)) = \emptyset$. We let $\mathcal{C}$ be the set of all small circuits which are relevant to show this, i.e., $\mathcal{C} = \{(s, r_s) \text{ small } \mathcal{B}\text{-circuit} \mid [r_s(\varepsilon)] \cap r(\mathrm{pos}(t)) \neq \emptyset\}$.

Let $(s, r_s) \in \mathcal{C}$ and $\mathbf{q} = r_s(\varepsilon)$. We may assume that $\mathbf{q} \in r(\mathrm{pos}(t))$ due to the following argument. If $\mathbf{q} \in r(\mathrm{pos}(t))$ does not hold, there exists some $\mathbf{p} \in r(\mathrm{pos}(t))$ with $\mathbf{p} \approx \mathbf{q}$. Then there exist Γ-words $s^{\mathbf{p}}_{\mathbf{q}}, s^{\mathbf{q}}_{\mathbf{p}} \in T_{\Gamma_\diamond}$ and runs $r^{\mathbf{p}}_{\mathbf{q}} \in \mathrm{Run}^\diamond_{\mathcal{B}}(\mathbf{q}, s^{\mathbf{p}}_{\mathbf{q}}, \mathbf{p})$ and $r^{\mathbf{q}}_{\mathbf{p}} \in \mathrm{Run}^\diamond_{\mathcal{B}}(\mathbf{p}, s^{\mathbf{q}}_{\mathbf{p}}, \mathbf{q})$. Thus, with $s' = s^{\mathbf{p}}_{\mathbf{q}}(s^{\mathbf{q}}_{\mathbf{p}})$ and $r_{s'} = r^{\mathbf{p}}_{\mathbf{q}}\langle r^{\mathbf{q}}_{\mathbf{p}} \to \Diamond_1(s^{\mathbf{p}}_{\mathbf{q}})\rangle$, we obtain a circuit $(s', r_{s'})$ with $r_{s'}(\varepsilon) = \mathbf{p}$ and $r_{s'}(\Diamond_1(s^{\mathbf{p}}_{\mathbf{q}})) = \mathbf{q}$. We can insert $(s', r_{s'})$ into t and r to obtain a tree t' and a run $r' \in \mathrm{Acc}_{\mathcal{B}}(t')$ with $\mathbf{q} \in r'(\mathrm{pos}(t'))$.

Now let $c_1, \dots, c_n$ be an enumeration of $\mathcal{C}$. We write $c_i = (s_i, r_i)$ and let $\mathbf{q}_i = r_i(\varepsilon)$, $w_{\mathbf{q}_i} \in \mathrm{pos}(t)$ with $r(w_{\mathbf{q}_i}) = \mathbf{q}_i$, and $w_i = \Diamond_1(s_i)$. We may assume that $c_1, \dots, c_n$ are ordered such that $w_{\mathbf{q}_1} \leq_l \dots \leq_l w_{\mathbf{q}_n}$. Then for every $i \in \{1, \dots, n\}$, we can insert the circuit $(s_i^{|Q'|}, r_i^{|Q'|\langle w_i \rangle})$ at $w_{\mathbf{q}_i}$ to obtain a tree $t' = t\langle s_n^{|Q'|} \to w_{\mathbf{q}_n}\rangle \cdots \langle s_1^{|Q'|} \to w_{\mathbf{q}_1}\rangle \in T_\Gamma$ together with a run $r'_{\mathcal{B}} = r\langle r_n^{|Q'|\langle w \rangle} \to w_{\mathbf{q}_n}\rangle \cdots \langle r_1^{|Q'|\langle w \rangle} \to w_{\mathbf{q}_1}\rangle \in \mathrm{Acc}_{\mathcal{B}}(t')$. For simplicity, we assume that the root of each circuit $(s_i^{|Q'|}, r_i^{|Q'|\langle w_i \rangle})$ is still at position $w_{\mathbf{q}_i}$ in t'.

Since $\operatorname{supp} \mathcal{B} = \operatorname{supp} \mathcal{A}'$, we find a run $r'_{\mathcal{A}'} \in \mathrm{Acc}_{\mathcal{A}'}(t')$. By pigeon hole principle, we find $0 \leq m_i < n_i \leq |Q'|$ for each $i \in \{1, \dots, n\}$ such that $r'_{\mathcal{A}'}(w_{\mathbf{q}_i} w_i^{m_i}) = r'_{\mathcal{A}'}(w_{\mathbf{q}_i} w_i^{n_i})$. We thus obtain runs $r_i^{\mathcal{A}} \in \mathrm{Run}^{\Diamond}_{\mathcal{A}'}(s^{n_i - m_i})$ through $r_i^{\mathcal{A}}(w) = r'_{\mathcal{A}'}(w_{\mathbf{q}_i} w)$ such that each $(s_i^{n_i - m_i}, r_i^{\mathcal{A}})$ is an $\mathcal{A}'$-circuit. We let $\tilde{s}_i = s_i^{n_i - m_i}$ and $r_i^{\mathcal{B}} = r_i^{n_i - m_i \langle w_i \rangle}$. Then $(\tilde{s}_i, r_i^{\mathcal{B}})$ are $\mathcal{B}$-circuits with $\mathrm{Vict}(c_i) = \mathrm{Vict}(\tilde{s}_i, r_i^{\mathcal{B}})$ for all i. For $\bar{v} = (v_1, \dots, v_n) \in \mathbb{N}^n$, we denote by $t_{\bar{v}}$ the tree obtained by adding v_i copies of $\tilde{s}_i$ to t' for each $i \in \{1, \dots, n\}$, i.e., the tree $t_{\bar{v}} = t'\langle \tilde{s}_n^{v_n} \to w_{\mathbf{q}_n}\rangle \cdots \langle \tilde{s}_1^{v_1} \to w_{\mathbf{q}_1}\rangle$. Then we see that the runs

$$r'_{\mathcal{B}}\langle (r_n^{\mathcal{B}})^{v_n \langle w_n^{n_n - m_n} \rangle} \to w_{\mathbf{q}_n}\rangle \cdots \langle (r_1^{\mathcal{B}})^{v_1 \langle w_1^{n_1 - m_1} \rangle} \to w_{\mathbf{q}_1}\rangle \in \mathrm{Acc}_{\mathcal{B}}(t_{\bar{v}})$$

$$r'_{\mathcal{A}'}\langle (r_n^{\mathcal{A}})^{v_n \langle w_n^{n_n - m_n} \rangle} \to w_{\mathbf{q}_n} w_n^{m_n}\rangle \cdots \langle (r_1^{\mathcal{A}})^{v_1 \langle w_1^{n_1 - m_1} \rangle} \to w_{\mathbf{q}_1} w_1^{m_1}\rangle \in \mathrm{Acc}_{\mathcal{A}'}(t_{\bar{v}})$$

are accepting on $t_{\bar{v}}$.

By assumption, there exists $I \in \{1, \dots, n\}$ such that $\bigcap_{i=1}^{I} \mathrm{Vict}(c_i) \neq \emptyset$ and $\bigcap_{i=1}^{I+1} \mathrm{Vict}(c_i) = \emptyset$. In the following, we show that $\bigcap_{i=1}^{I+1} \mathrm{Vict}(c_i) \neq \emptyset$, which yields the desired contradiction. For $k, l \in \mathbb{N}$, let $t_{k,l} = t_{(k,\dots,k,l,0,\dots,0)}$, where l is at index $I+1$. Since $\mathcal{A}'$ is unambiguous, we see that with $x = [\![\mathcal{A}']\!](t_{0,0})$, $\kappa = \mathrm{wt}_{\mathcal{A}'}(\tilde{s}_1, r_1^{\mathcal{A}}) + \dots + \mathrm{wt}_{\mathcal{A}'}(\tilde{s}_I, r_I^{\mathcal{A}})$, and $\lambda = \mathrm{wt}_{\mathcal{A}'}(\tilde{s}_{I+1}, r_{I+1}^{\mathcal{A}})$, we have

$$[\![\mathcal{A}']\!](t_{k,l}) = x + k\kappa + l\lambda$$

for all $k, l \in \mathbb{N}$.

Due to the definition of victorious coordinates, we can find a number $N \in \mathbb{N}$ such that for all $l \geq N$, the tuple $[\![\mathcal{B}]\!](t_{0,l})$ has its maximum in some victorious coordinate from $\mathrm{Vict}(c_{I+1})$; this is because with every repetition of a circuit, non-victorious coordinates fall behind victorious coordinates in terms of weight by a small fixed margin. Then for every $l' \geq 0$, we have

$$[\![\mathcal{A}]\!](t_{0,N+l'}) = [\![\mathcal{A}]\!](t_{0,N}) + l' \cdot \mathrm{wt}(\tilde{s}_{I+1}, r_{I+1}^{\mathcal{B}}).$$

Since $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$, it follows that

$$\mathrm{wt}(\tilde{s}_{I+1}, r_{I+1}^{\mathcal{B}}) = [\![\mathcal{A}]\!](t_{0,N+1}) - [\![\mathcal{A}]\!](t_{0,N}) = x + (N+1)\lambda - (x + N\lambda) = \lambda.$$

Similarly, due to the assumption that $\bigcap_{i=1}^{I} \mathrm{Vict}(c_i) \neq \emptyset$, we can find for every $l \in \mathbb{N}$ a number $M_l \in \mathbb{N}$ such that for all $k \geq M_l$, the tuple $[\![\mathcal{B}]\!](t_{k,l})$ has its maximum in some victorious coordinate $j_l \in \bigcap_{i=1}^{I} \mathrm{Vict}(c_i)$. We let $\tilde{M} = \max_{l=0}^{M} M_l$. By pigeon hole principle, there exist $l_1, l_2 \in \{0, \dots, M\}$ with $l_1 < l_2$ and $j_{l_1} = j_{l_2}$. Then we see that, again due to $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$, we have

$$(l_2 - l_1)\mathrm{wt}_{j_{l_1}}(\tilde{s}_{I+1}, r_{I+1}^{\mathcal{B}}) = [\![\mathcal{A}]\!](t_{\tilde{M}, l_2}) - [\![\mathcal{A}]\!](t_{\tilde{M}, l_1}) = (l_2 - l_1)\lambda.$$

It follows that

$$\mathrm{wt}(\tilde{s}_{I+1}, r^{\mathcal{B}}_{I+1}) = \lambda = \mathrm{wt}_{j_{l_1}}(\tilde{s}_{I+1}, r^{\mathcal{B}}_{I+1}),$$

which means that $j_{l_1} \in \mathrm{Vict}(c_{I+1})$. Since $j_{l_1} \in \bigcap_{i=1}^{I} \mathrm{Vict}(c_i)$ also holds, we have $\bigcap_{i=1}^{I+1} \mathrm{Vict}(c_i) \neq \emptyset$, which is a contradiction to the choice of I. In conclusion, t and r as chosen do not exist and therefore $\mathcal{B}$ satisfies (**P**). □

Next, we address the sufficiency of (**P**). In the following, we assume that $\mathcal{B}$ satisfies (**P**) and construct an unambiguous max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}']\!] = [\![\mathcal{A}]\!]$.

The idea behind $\mathcal{A}'$ is as follows. The states of $\mathcal{A}'$ will be taken from $\mathbb{R}_{\max}^{M} \times Q$. From a bottom-up perspective, $\mathcal{A}'$ remembers in each $\mathbb{R}_{\max}$-coordinate the weight which $\mathcal{B}$ would have assigned to the run in this coordinate "so far". Since this can become unbounded, we normalize the smallest coordinate to 0 in each transition, make this coordinate's weight the transition weight, and remember only the difference to this weight in the remaining coordinates. Still, these differences can become unbounded. Therefore, once the difference exceeds a certain bound $(2N+1)C$, the coordinates with small weights are discarded by being set to $-\infty$ and only the large weights are remembered. Here, N is the maximum possible number of nodes of a tree over Γ of height at most $2|Q|$. The constant C is the largest difference between all weights occurring in the automata $\mathcal{A}_1, \ldots, \mathcal{A}_M$.

We can show that the coordinate l which in $\mathcal{B}$ eventually yields the largest weight will not be discarded as follows. First, we can show that the weight of a victorious coordinate of a run will never be smaller than the largest weight (over all coordinates) minus NC. Second, we can show that if k is victorious, then the weight of coordinate l will never be smaller than the weight of k minus $NC + C$. Our assumption is that (**P**) holds, so there exists some victorious coordinate in every accepting run. Therefore, the weight of l will never be smaller than the largest weight minus $(2N+1)C$ and is never discarded.

Formally, we define $\mathcal{A}'$ as follows.

Construction 4.19. We let

$$N = \sum_{i=0}^{2|Q|} \mathrm{rk}(\Gamma)^i = \max\{|\mathrm{pos}(t)| \mid t \in T_\Gamma, \mathrm{height}(t) \leq 2|Q|\}$$

$$R = \bigcup_{i=1}^{M} (\mu_i(\Delta_{\mathcal{A}_i}) \cup \nu_i(Q_i))$$

$$C = \max R - \min(R \setminus \{-\infty\}).$$

For $\mathbf{x} = (x_1, \ldots, x_M) \in \mathbb{R}_{\max}^{M}$, we denote the smallest weight of $\mathbf{x}$ by

$$\check{\mathbf{x}} = \min\{x_i \mid 1 \leq i \leq M, x_i \neq -\infty\},$$

and define the normalization of $\mathbf{x}$ by

$$\underline{\mathbf{x}} = \mathbf{x} - (\check{\mathbf{x}}, \ldots, \check{\mathbf{x}}).$$

We construct an unambiguous max-plus-WTA $\mathcal{A}' = (Q', \Gamma, \mu', \nu')$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ and $Q' \subset \mathbb{R}_{\max}^{M} \times Q$ as follows.

Rule 1 For $(a, \mathbf{q}) \in \Delta_{\mathcal{B}} \cap (\Gamma \times Q)$ with $\mathbf{x} = \mu(a, \mathbf{q}) \in \mathbb{R}^M$, we let $(\underline{\mathbf{x}}, \mathbf{q}) \in Q'$ and $\mu'(a, (\underline{\mathbf{x}}, \mathbf{q})) = \check{\mathbf{x}}$.

Rule 2 Assume we have $d = (\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}_0) \in \Delta_{\mathcal{B}}$ with $\mathbf{x} = \mu(d) \in \mathbb{R}^M$ and $(\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m) \in Q'$ for some $\mathbf{z}_1, \dots, \mathbf{z}_m \in \mathbb{R}^M_{\max}$. We let $\mathbf{t} = \mathbf{x} + \sum_{i=1}^m \mathbf{z}_i$ and define $\mathbf{y} \in \mathbb{R}^M_{\max}$ through

$$y_i = \begin{cases} -\infty & \text{if } t_i < \max\{t_j \mid 1 \le j \le M\} - (2N+1)C \\ t_i & \text{otherwise.} \end{cases}$$

We let $(\underline{\mathbf{y}}, \mathbf{p}_0) \in Q'$ and $\mu'((\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m), a, (\underline{\mathbf{y}}, \mathbf{p}_0)) = \check{\mathbf{y}}$.

Rule 3 Assume $(\mathbf{z}, \mathbf{p}) \in Q'$ and $\mathbf{x} = \nu(\mathbf{p}) \in \mathbb{R}^M$. Then we let $\nu'(\mathbf{z}, \mathbf{p}) = \max_{i=1}^M (z_i + x_i)$.

Note that from the above definition, it is not obvious that Q' is finite, which is what we will show later on. The following is clear from the construction.

Proposition 4.20. *The projection $\pi \colon Q' \to Q, (\mathbf{z}, \mathbf{p}) \mapsto \mathbf{p}$ induces a bijection between the accepting runs of $\mathcal{B}$ and $\mathcal{A}'$. In particular, $\mathcal{A}'$ is unambiguous.*

Using a simple induction we can show the following relationship between the runs of $\mathcal{A}'$ and $\mathcal{B}$.

Lemma 4.21. *Let $t \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{B}}(t)$, and $r' = \pi^{-1}(r) \in \mathrm{Run}_{\mathcal{A}'}(t)$. We write $r'(\varepsilon) = (\mathbf{z}, \mathbf{p})$, then for every $l \in \{1, \dots, M\}$ we have*

(i) if $z_l \neq -\infty$ then $\mathrm{wt}_l(t, r) = \mathrm{wt}_{\mathcal{A}'}(t, r') + z_l$

(ii) if $z_l = -\infty$ then for some $w \in \mathrm{pos}(t)$ we have $\mathrm{wt}_l(t{\restriction_w}, r{\restriction_w}) < \mathrm{wt}(t{\restriction_w}, r{\restriction_w}) - (2N+1)C$.

Proof. (i) We proceed by induction on the height of t. If $\mathrm{height}(t) = 0$, the statement follows from Rule 1. Otherwise, we let $a = t(\varepsilon)$, $m = \mathrm{rk}(a)$, and $r'(i) = (\mathbf{z}_i, \mathbf{p}_i)$ for $i \in \{1, \dots, m\}$. Since $z_l \neq -\infty$, we know by Rule 2 that $z_{il} \neq -\infty$ holds for all $i \in \{1, \dots, m\}$. Thus, by induction we have $\mathrm{wt}_l(t{\restriction_i}, r{\restriction_i}) = \mathrm{wt}_{\mathcal{A}'}(t{\restriction_i}, r'{\restriction_i}) + z_{il}$ for all $i \in \{1, \dots, m\}$. It follows by Rule 2 that with $\mathbf{x} = \mu(\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p})$ and $y = \mu'((\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m), a, (\mathbf{z}, \mathbf{p}))$ we have

$$\begin{aligned} \mathrm{wt}_l(t, r) &= x_l + \sum_{i=1}^m \mathrm{wt}_l(t{\restriction_i}, r{\restriction_i}) \\ &= x_l + \sum_{i=1}^m (\mathrm{wt}_{\mathcal{A}'}(t{\restriction_i}, r'{\restriction_i}) + z_{il}) \\ &= z_l + y + \sum_{i=1}^m \mathrm{wt}_{\mathcal{A}'}(t{\restriction_i}, r'{\restriction_i}) \\ &= \mathrm{wt}_{\mathcal{A}'}(t, r') + z_l. \end{aligned}$$

(ii) Assume $z = -\infty$ and let w be a prefix-maximal position with the property that for $(\mathbf{z}', \mathbf{p}') = r'(w)$ we have $z'_l = -\infty$. By Rule 1, w cannot be a leaf. We let $a = t(w)$, $m = \mathrm{rk}(a)$, $r'(w) = (\mathbf{z}_0, \mathbf{p}_0)$, and $r'(wi) = (\mathbf{z}_i, \mathbf{p}_i)$ for $i \in \{1, \dots, m\}$. By choice of w, we have $z_{il} \neq -\infty$ for all $i \in \{1, \dots, m\}$, so by (i) we have $\mathrm{wt}_l(t|_{wi}, r|_{wi}) = \mathrm{wt}_{\mathcal{A}'}(t|_{wi}, r'|_{wi}) + z_{il}$ for all $i \in \{1, \dots, m\}$. Let $\mathbf{x} = \mu(\mathbf{p}_1, \dots, \mathbf{p}_m, a, \mathbf{p}_0)$ and $y = \mu'((\mathbf{z}_1, \mathbf{p}_1), \dots, (\mathbf{z}_m, \mathbf{p}_m), a, (\mathbf{z}_0, \mathbf{p}_0))$. Then since $z_{0l} = -\infty$, there exists by Rule 2 some $j \in \{1, \dots, M\}$ such that $z_{0j} \neq -\infty$ and $x_l + \sum_{i=1}^m z_{il} < z_{0j} + y - (2N+1)C$. Thereby, we have

$$\begin{aligned}
\mathrm{wt}_l(t|_w, r|_w) &= x_l + \sum_{i=1}^m \mathrm{wt}_l(t|_{wi}, r|_{wi}) \\
&= x_l + \sum_{i=1}^m (\mathrm{wt}_{\mathcal{A}'}(t|_{wi}, r'|_{wi}) + z_{il}) \\
&< z_{0j} + y + \sum_{i=1}^m \mathrm{wt}_{\mathcal{A}'}(t|_{wi}, r'|_{wi}) - (2N+1)C \\
&= z_{0j} + \mathrm{wt}_{\mathcal{A}'}(t|_w, r|_w) - (2N+1)C \\
&= \mathrm{wt}_j(t|_w, r|_w) - (2N+1)C \\
&\leq \mathrm{wt}(t|_w, r|_w) - (2N+1)C.
\end{aligned}$$

□

The dominance property (**P**) is defined only through small circuits. Thus, in order to use (**P**), we describe in the following how to decompose pairs (t, r) of a Γ-tree or a Γ-word t and a run r of $\mathcal{B}$ on t into small circuits. Intuitively, we cut circuits from the bottom of the tree using the pigeon hole principle.

Construction 4.22. Let $t \in T_{\Gamma_\diamond}$ be a Γ-tree or a Γ-word and $r \in \mathrm{Run}^\diamond_{\mathcal{B}}(t)$. A *circuit decomposition* of t and r is a *stub* (t_0, r_0), where $t_0 \in T_{\Gamma_\diamond}$ with $\mathrm{height}(t_0) \leq 2|Q|$ and $r_0 \in \mathrm{Run}^\diamond_{\mathcal{B}}(t_0)$, together with a finite sequence of small $\mathcal{B}$-circuits $(s_1, r_1), \dots, (s_n, r_n)$ defined as follows. If $\mathrm{height}(t) \leq 2|Q|$, then we let $t_0 = t$ and $r_0 = r$ and conclude the decomposition. Otherwise, we cut a small circuit from t and r.

If t is a Γ-tree, we proceed as follows. We choose $uv \in \mathrm{pos}(t)$ with $|uv| = \mathrm{height}(t)$ and $|v| = |Q|$. By pigeon hole principle, we find $u \leq_p w_1 <_p w_2 \leq_p uv$ with $r(w_1) = r(w_2)$. We let $s = (t\langle \diamond \to w_2 \rangle)|_{w_1}$, then for $w \in \mathrm{pos}(s)$ we see that

$$\mathrm{height}(t) \geq |w_1 w| = |w_1| + |w| \geq |u| + |w| = \mathrm{height}(t) - |Q| + |w|$$

from which $|w| \leq |Q|$ and therefore $\mathrm{height}(s) \leq |Q|$ follows. Thus from r we obtain a small circuit (s, r'') through $r''(w) = r(w_1 w)$ for $w \in \mathrm{pos}(s)$. With $t' = t\langle t|_{w_2} \to w_1 \rangle$ we obtain from r a run $r' \in \mathrm{Run}_{\mathcal{B}}(t')$ through $r' = r\langle r|_{w_2} \to w_1 \rangle$. We continue the decomposition with t' and r'. This procedure ends after finitely many steps.

If t is a Γ-word, we proceed in the following way in order to ensure that the process above never creates a 2-Γ-context when cutting a circuit. If there exists a position $v' \in \mathrm{pos}(t)$ which is prefix-independent from $\Diamond_1(t)$ and for which $\mathrm{height}(t|_{v'}) \geq |Q|$, we let $uv \in \mathrm{pos}(t)$ with $v' \leq_p uv$, $|uv| = |v'| + \mathrm{height}(t|_{v'})$, and $|v| = |Q|$. By

pigeon hole principle, we find $u \leq_p w_1 <_p w_2 \leq_p uv$ with $r(w_1) = r(w_2)$. We let $s = (t\langle \diamond \to w_2 \rangle)\restriction_{w_1}$, then for $w \in \mathrm{pos}(s)$ we see that

$$|v'| + \mathrm{height}(t\restriction_{v'}) \geq |w_1 w| = |w_1| + |w| \geq |u| + |w| = |v'| + \mathrm{height}(t\restriction_{v'}) - |Q| + |w|$$

from which $|w| \leq |Q|$ and therefore $\mathrm{height}(s) \leq |Q|$ follows. Thus from r we obtain a small circuit (s, r'') through $r''(w) = r(w_1 w)$ for $w \in \mathrm{pos}(s)$. With $t' = t\langle t\restriction_{w_2} \to w_1 \rangle$ we obtain from r a run $r' \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(t')$ through $r' = r\langle r\restriction_{w_2} \to w_1 \rangle$. Note that both s and t' are Γ-words since v' is prefix-independent from $\Diamond_1(t)$. We continue the decomposition with t' and r'.

If t is a Γ-word but $\mathrm{height}(t\restriction_{v'}) < |Q|$ for all $v' \in \mathrm{pos}(t)$ which are prefix-independent from $\Diamond_1(t)$, we proceed as follows. First, we show that $|\Diamond_1(t)| \geq |Q|$. We know that $\mathrm{height}(t) > 2|Q|$, thus there exists a position $w \in \mathrm{pos}(t)$ with $|w| \geq 2|Q|$. If $w \leq_p \Diamond_1(t)$, it immediately follows that $|\Diamond_1(t)| \geq |Q|$. Otherwise, since $\Diamond_1(t)$ is a leaf, w and $\Diamond_1(t)$ are prefix-independent and we can write $w = viv_1$ and $\Diamond_1(t) = vjv_2$ for some $i, j \in \mathbb{N}_+$ with $i \neq j$. As vi is prefix-independent from $\Diamond_1(t)$, we see that $|v_1| \leq \mathrm{height}(t\restriction_{vi}) < |Q|$ and therefore $|vi| \geq 2|Q| - |Q| = |Q|$. In particular, we have $|\Diamond_1(t)| \geq |vj| \geq |Q|$.

Since $|\Diamond_1(t)| \geq |Q|$, we can write $\Diamond_1(t) = uv$ with $|v| = |Q|$. By pigeon hole principle, we find $u \leq_p w_1 <_p w_2 \leq_p uv$ with $r(w_1) = r(w_2)$. We let $s = (t\langle \diamond \to w_2 \rangle)\restriction_{w_1}$ and show that $\mathrm{height}(s) \leq 2|Q|$. Let $w \in \mathrm{pos}(s)$. If $uw \leq_p \Diamond_1(t)$, we have $|w| \leq |v| = |Q|$. Otherwise, if uw is prefix-independent from $\Diamond_1(t)$, we can write $uw = uv'iv_i$ and $\Diamond_1(t) = uv'jv_j$ for some $i, j \in \mathbb{N}_+$ with $i \neq j$. Then $uv'i$ is prefix-independent from $\Diamond_1(t)$ which means we have $|v_i| \leq \mathrm{height}(t\restriction_{uv'i}) < |Q|$. Due to $|v'jv_j| = |v| = |Q|$, we see that $|v'| \leq |Q|$ and therefore $|w| = |v'iv_i| < |Q| + 1 + |Q|$. Therefore, we have $\mathrm{height}(s) \leq 2|Q|$. Thus from r we obtain a small circuit (s, r'') through $r''(w) = r(w_1 w)$ for $w \in \mathrm{pos}(s)$. With $t' = t\langle t\restriction_{w_2} \to w_1 \rangle$ we obtain from r a run $r' \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(t')$ through $r' = r\langle r\restriction_{w_2} \to w_1 \rangle$. Note that both s and t' are Γ-words since $w_2 \leq_p \Diamond_1(t)$. We continue the decomposition with t' and r'.

In the following lemma, we show that the weights of victorious coordinates never become much smaller than the maximum weight over all coordinates.

Lemma 4.23. *Let $t \in T_{\Gamma_\diamond}$ be a Γ-tree or a Γ-word and $r \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(t)$. If $k \in \mathrm{Vict}(r(\mathrm{pos}(t)))$ then $\mathrm{wt}_k(t, r) \geq \mathrm{wt}(t, r) - NC$.*

Proof. Take a circuit decomposition of t and r as in Construction 4.22 with stub (t_0, r_0) and small circuits $(s_1, r_1), \ldots, (s_n, r_n)$. Since $|\mathrm{pos}(t_0)| \leq N$, we have for all $j \in \{1, \ldots, M\}$ that

$$\begin{aligned} \mathrm{wt}_k(t, r) &= \mathrm{wt}_k(t_0, r_0) + \sum_{i=1}^{n} \mathrm{wt}_k(s_i, r_i) \\ &\geq \mathrm{wt}_j(t_0, r_0) - NC + \sum_{i=1}^{n} \mathrm{wt}_j(s_i, r_i) \\ &= \mathrm{wt}_j(t, r) - NC. \end{aligned}$$

This is true in particular for j with $\mathrm{wt}(t, r) = \mathrm{wt}_j(t, r)$. □

We are now able to show that Q' is finite. We proceed by contradiction and show that if Q' was infinite, we would be able to find arbitrarily long successions $(\mathbf{z}_n, \mathbf{p}) \preceq \ldots \preceq (\mathbf{z}_1, \mathbf{p})$ in Q' with $\mathbf{z}_1, \ldots, \mathbf{z}_n$ pairwise distinct. Then we show that such successions can in fact not be arbitrarily long, as from every $\mathbf{z}_i$ to the next, the difference in weights of at least one non-victorious coordinate to the victorious coordinates grows by at least δ, where δ is a fixed constant. Thus, after some $\mathbf{z}_i$, these differences exceed $(2N+1)C$ for all all non-victorious coordinates, and all subsequent $\mathbf{z}_i$ remain constant.

Lemma 4.24. *Q' is a finite set.*

Proof. We show first that if Q' is infinite, then for at least one $\mathbf{p} \in Q$ we can find arbitrarily long successions $(\mathbf{z}_n, \mathbf{p}) \preceq \ldots \preceq (\mathbf{z}_1, \mathbf{p})$ with $\mathbf{z}_1, \ldots, \mathbf{z}_n$ pairwise distinct. Let $P_0 \subseteq Q'$ be the set of all states added to Q' by Rule 1. For $i \geq 0$, let $P_{i+1} \subseteq Q'$ be the set of all states added to Q' by Rule 2 using only states $(\mathbf{z}_1, \mathbf{p}_1), \ldots, (\mathbf{z}_m, \mathbf{p}_m) \in P_i$. Then for all $i \geq 0$ we have $P_i \subseteq P_{i+1}$, $P_{i+1} \setminus P_i \neq \emptyset$ since Q is infinite, and P_i is finite.

Let $i > 0$ and $(\mathbf{z}, \mathbf{p}) \in P_{i+1} \setminus P_i$. Then there are $(\mathbf{z}_1, \mathbf{p}_1), \ldots, (\mathbf{z}_m, \mathbf{p}_m) \in P_i$ with at least one $(\mathbf{z}_j, \mathbf{p}_j) \in P_i \setminus P_{i-1}$ such that $(\mathbf{z}, \mathbf{p})$ is added to Q' by Rule 2 using $(\mathbf{z}_1, \mathbf{p}_1), \ldots, (\mathbf{z}_m, \mathbf{p}_m) \in P_i$, and a valid transition $(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}) \in \Delta_{\mathcal{B}}$. In particular, we have $(\mathbf{z}, \mathbf{p}) \preceq (\mathbf{z}_j, \mathbf{p}_j)$.

Now let $H > 0$, $n > H|Q|$, and $p \in P_n \setminus P_{n-1}$. Then according to the argumentation we just did, we can find $(\mathbf{z}_n, \mathbf{p}_n) \preceq \ldots \preceq (\mathbf{z}_0, \mathbf{p}_0)$ with $(\mathbf{z}_0, \mathbf{p}_0) \in P_0$ and $(\mathbf{z}_i, \mathbf{p}_i) \in P_i \setminus P_{i-1}$ for $i > 0$. In particular, $(\mathbf{z}_0, \mathbf{p}_0), \ldots, (\mathbf{z}_n, \mathbf{p}_n)$ are pairwise distinct. By pigeon hole principle, at least one $\mathbf{p} \in Q$ occurs H or more times among $\mathbf{p}_0, \ldots, \mathbf{p}_n$. Hence, we find $i_1 < \ldots < i_H$ with $\mathbf{p}_{i_1} = \ldots = \mathbf{p}_{i_H} = \mathbf{p}$ and have $(\mathbf{z}_{i_H}, \mathbf{p}) \preceq \ldots \preceq (\mathbf{z}_{i_1}, \mathbf{p})$ with $\mathbf{z}_{i_1}, \ldots, \mathbf{z}_{i_H}$ pairwise distinct.

Now we show that there can be no arbitrarily long successions $(\mathbf{z}_n, \mathbf{p}) \preceq \ldots \preceq (\mathbf{z}_1, \mathbf{p})$ with $\mathbf{z}_1, \ldots, \mathbf{z}_n$ pairwise distinct in Q'. This shows in particular that Q' must be finite. We define the constant

$$\delta = \min_{\substack{(s,r) \text{ small } \mathcal{B}\text{-circuit} \\ \mathrm{wt}_i(s,r) < \mathrm{wt}(s,r) \text{ for some } i}} \mathrm{wt}(s,r) - \max\{\mathrm{wt}_i(s,r) \mid \mathrm{wt}_i(s,r) < \mathrm{wt}(s,r)\}$$

where the minimum over the empty set is defined as ∞. Assume we have $(\mathbf{x}, \mathbf{p}) \preceq (\mathbf{y}, \mathbf{p})$ with $\mathbf{x} \neq \mathbf{y}$. Then there exists a Γ-word $s \in T_{\Gamma_\diamond}$ with a run $r' \in \mathrm{Run}^{\diamond}_{\mathcal{A}'}((\mathbf{y}, \mathbf{p}), s, (\mathbf{x}, \mathbf{p}))$. By projecting r to Q, we obtain a run $r \in \mathrm{Run}^{\diamond}_{\mathcal{B}}(\mathbf{p}, s, \mathbf{p})$. Take a circuit decomposition of s and r as in Construction 4.22 with stub (s_0, r_0) and small circuits $(s_1, r_1), \ldots, (s_n, r_n)$. Note that now, (s_0, r_0) is also a small circuit. Since $\mathcal{B}$ satisfies $(\mathbf{P})$, there exists $k \in \mathrm{Vict}(r(\mathrm{pos}(t)))$. Due to Lemma 4.21(ii) and Lemma 4.23, we have $x_k, y_k \in \mathbb{R}$.

For all $i \in \{0, \ldots, n\}$ and $j \in \{1, \ldots, M\}$, we have either $\mathrm{wt}_j(s_i, r_i) = \mathrm{wt}_k(s_i, r_i)$ or $\mathrm{wt}_j(s_i, r_i) \leq \mathrm{wt}_k(s_i, r_i) - \delta$. Hence, for all $j \in \{1, \ldots, M\}$ we have either $x_k - x_j = y_k - y_j$ or $x_k - x_j \geq y_k - y_j + \delta$. Since $\mathbf{x} \neq \mathbf{y}$, we have $x_k - x_j \geq y_k - y_j + \delta$ for at least one $j \in \{1, \ldots, M\}$.

Now let $(\mathbf{z}_n, \mathbf{p}) \preceq \ldots \preceq (\mathbf{z}_1, \mathbf{p})$ be a succession as above with $\mathbf{z}_1, \ldots, \mathbf{z}_n$ pairwise distinct. Then in every step from $\mathbf{z}_i$ to $\mathbf{z}_{i+1}$, for at least one non-victorious coordinate

the difference the victorious coordinates grows by at least δ. If this differences exceeds $(2N+1)C$, the coordinate is set to $-\infty$. Thus, at some point all non-victorious coordinates are $-\infty$. It follows that n cannot be arbitrarily large. □

Next, we show that if a coordinate yields the maximum weight in $\mathcal{B}$, then during the whole computation of the weight of the run, the distance to the maximum weight does not exceed the bound $(2N+1)C$.

Lemma 4.25. *Let $t \in T_\Gamma$ and $r \in \mathrm{Acc}_\mathcal{B}(t)$. If for $l \in \{1,\dots,M\}$ we have $[\![\mathcal{A}]\!](t) = [\![\mathcal{A}_l]\!](t)$, then for all $w \in \mathrm{pos}(t)$ we have $\mathrm{wt}_l(t{\restriction_w}, r{\restriction_w}) \geq \mathrm{wt}(t{\restriction_w}, r{\restriction_w}) - (2N+1)C$.*

Proof. We let $w \in \mathrm{pos}(t)$, $t' = t\langle \diamond \to w \rangle$, and let r' be the run on t' we obtain from r through $r'(v) = r(v)$. We write $r(\varepsilon) = (q_1, \dots, q_M)$ and let $k \in \mathrm{Vict}(r(\mathrm{pos}(t)))$. By assumption, we have

$$\nu_l(q_l) + \mathrm{wt}_l(t', r') + \mathrm{wt}_l(t{\restriction_w}, r{\restriction_w}) \geq \nu_k(q_k) + \mathrm{wt}_k(t', r') + \mathrm{wt}_k(t{\restriction_w}, r{\restriction_w}).$$

Due to Lemma 4.23, we have

$$\mathrm{wt}_l(t', r') \leq \mathrm{wt}(t', r') \leq \mathrm{wt}_k(t', r') + NC.$$

Thus, applying Lemma 4.23 also to $\mathrm{wt}_k(t{\restriction_w}, r{\restriction_w})$ we can conclude

$$\begin{aligned}
\mathrm{wt}_l(t{\restriction_w}, r{\restriction_w}) &\geq \nu_k(q_k) - \nu_l(q_l) + \mathrm{wt}_k(t', r') - \mathrm{wt}_k(t', r') - NC + \mathrm{wt}_k(t{\restriction_w}, r{\restriction_w}) \\
&\geq -C - NC + \mathrm{wt}(t{\restriction_w}, r{\restriction_w}) - NC \\
&= \mathrm{wt}(t{\restriction_w}, r{\restriction_w}) - (2N+1)C.
\end{aligned}$$

□

We are now ready to show that the behaviors of $\mathcal{A}'$ and $\mathcal{A}$ coincide.

Lemma 4.26. *We have $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$.*

Proof. It is clear that $\mathrm{supp}\,\mathcal{A} = \mathrm{supp}\,\mathcal{B} = \mathrm{supp}\,\mathcal{A}'$. Let $t \in \mathrm{supp}\,\mathcal{A}$ and let $r \in \mathrm{Acc}_\mathcal{B}(t)$ and $r' \in \mathrm{Acc}_{\mathcal{A}'}(t)$ be the unique accepting runs on t. We write $r'(\varepsilon) = (\mathbf{z}, \mathbf{q})$ and let $l \in \{1,\dots,M\}$ with $[\![\mathcal{A}_l]\!](t) = \max_{i=1}^M [\![\mathcal{A}_i]\!](t)$. Combining Lemma 4.21(ii) and Lemma 4.25, we see that we have $z_l \neq -\infty$. Thus, by Lemma 4.21(i) we have for all $i \in \{1,\dots,M\}$ that

$$\begin{aligned}
\nu_i(q_i) + z_i &\leq \nu_i(q_i) + \mathrm{wt}_i(t, r) - \mathrm{wt}_{\mathcal{A}'}(t, r') \\
&= [\![\mathcal{A}_i]\!](t) - \mathrm{wt}_{\mathcal{A}'}(t, r') \\
&\leq [\![\mathcal{A}_l]\!](t) - \mathrm{wt}_{\mathcal{A}'}(t, r') \\
&= \nu_l(q_l) + \mathrm{wt}_l(t, r) - \mathrm{wt}_{\mathcal{A}'}(t, r') \\
&= \nu_l(q_l) + z_l.
\end{aligned}$$

Therefore, again by Lemma 4.21(i) we have

$$\begin{aligned}
[\![\mathcal{A}]\!](t) &= \max_{i=1}^{M} [\![\mathcal{A}_i]\!](t) \\
&= [\![\mathcal{A}_l]\!](t) \\
&= \nu_l(q_l) + \mathrm{wt}_l(t, r) \\
&= \nu_l(q_l) + \mathrm{wt}_{\mathcal{A}'}(t, r') + z_l \\
&= \mathrm{wt}_{\mathcal{A}'}(t, r') + \max_{i=1}^{M}(\nu_i(q_i) + z_i) \\
&= \mathrm{wt}_{\mathcal{A}'}(t, r') + \nu'(\mathbf{z}, \mathbf{q}) \\
&= [\![\mathcal{A}']\!](t).
\end{aligned}$$

□

In conclusion, $\mathcal{A}'$ is an unambiguous max-plus-WTA with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$.

4.5 The Sequentiality Problem

The sequentiality problem asks whether for a given max-plus-WTA $\mathcal{A}$, there exists a deterministic max-plus-WTA $\mathcal{A}'$ such that $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. In this section, we show that the sequentiality problem is decidable for finitely ambiguous max-plus-WTA. For words, the corresponding result is known due to [63]. Like in [63], we obtain the decidability of the sequentiality problem for finitely ambiguous automata by combining the decidability of the unambiguity problem for these automata with the decidability of the sequentiality problem for unambiguous automata. Although to our knowledge the decidability of the sequentiality problem for unambiguous max-plus-WTA has never been stated explicitly, it follows rather easily from some known results and ideas.

We begin by showing that the *Lipschitz property* of deterministic max-plus word automata [63, 75] also holds for deterministic max-plus tree automata. On words, this Lipschitz property can be formulated follows. Let $\mathcal{A}$ be a deterministic max-plus-WA and let L be the largest weight, in terms of absolute value, occurring in $\mathcal{A}$ (excluding $-\infty$). Then for two words $w_1 = uv_1$ and $w_2 = uv_2$ which have an accepting run in $\mathcal{A}$, the difference between $[\![\mathcal{A}]\!](w_1)$ and $[\![\mathcal{A}]\!](w_2)$ can be at most $|L|(|v_1| + |v_2| + 2)$. This is clear since the unique runs of $\mathcal{A}$ on w_1 and w_2 will be identical on the prefix u, and then with every remaining letter of each word the difference between both runs cannot grow more than $|L|$. For deterministic max-plus-WTA, we can show a similar statement as follows.

Lemma 4.27 (c.f. [63, end of Section 2.4][75, Section 3.2])**.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a deterministic max-plus-WTA, let $X = (\mu(\Delta_\mathcal{A}) \cup \nu(Q)) \setminus \{-\infty\}$, and let $L = \max_{x \in X} |x|$. Furthermore, let $t_1, t_2 \in \operatorname{supp}(\mathcal{A})$ be two trees and let $w_1 \in \operatorname{pos}(t_1)$ and $w_2 \in \operatorname{pos}(t_2)$ be two positions such that $t_1{\restriction}_{w_1} = t_2{\restriction}_{w_2}$. Then with $t = t_1{\restriction}_{w_1}$ we have*

$$|[\![\mathcal{A}]\!](t_1) - [\![\mathcal{A}]\!](t_2)| \leq L(|t_1| + |t_2| - 2|t| + 2).$$

Proof. Since $\mathcal{A}$ is deterministic, there exists exactly one run $r_1 \in \operatorname{Acc}_\mathcal{A}(t_1)$ and exactly one run $r_2 \in \operatorname{Acc}_\mathcal{A}(t_2)$. Likewise, there exists exactly one run $r \in \operatorname{Run}_\mathcal{A}(t)$. Due to $t_1{\restriction}_{w_1} = t_2{\restriction}_{w_2} = t$, we thus have $r_1{\restriction}_{w_1} = r_2{\restriction}_{w_2} = r$. It follows that

$$\begin{aligned}
&|[\![\mathcal{A}]\!](t_1) - [\![\mathcal{A}]\!](t_2)| \\
&= \Big| \sum_{w \in \operatorname{pos}(t_1)} \mu(\mathbb{t}(t_1, r_1, w)) + \nu(r_1(\varepsilon)) - \Big(\sum_{w \in \operatorname{pos}(t_2)} \mu(\mathbb{t}(t_2, r_2, w)) + \nu(r_2(\varepsilon)) \Big) \Big| \\
&= \Big| \sum_{\substack{w \in \operatorname{pos}(t_1) \\ \neg(w_1 \leq_p w)}} \mu(\mathbb{t}(t_1, r_1, w)) + \nu(r_1(\varepsilon)) - \Big(\sum_{\substack{w \in \operatorname{pos}(t_2) \\ \neg(w_2 \leq_p w)}} \mu(\mathbb{t}(t_2, r_2, w)) + \nu(r_2(\varepsilon)) \Big) \Big| \\
&\leq L(|t_1| - |t| + 1) + L(|t_2| - |t| + 1) \\
&= L(|t_1| + |t_2| - 2|t| + 2).
\end{aligned}$$

□

Next, we recall the twins property. Let Γ be a ranked alphabet. We begin by introducing the concepts of *siblings* and *twins*. Intuitively, two states are called *siblings* if they can be "reached" by the same tree. Two siblings are called *twins* if for

every Γ-word which can "loop" in both states, the maximal weight for the loop is the same in both states.

Definition 4.28. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Two states $p, q \in Q$ are called *siblings* if there exists a tree $u \in T_\Gamma$ such that $\mathrm{Run}_\mathcal{A}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(u, q) \neq \emptyset$. We recall that $\mathrm{Run}_\mathcal{A}(u, p)$ and $\mathrm{Run}_\mathcal{A}(u, q)$ contain only valid runs.

Two siblings p, q are called *twins* if for every Γ-word s and weights

$$x = \max_{r \in \mathrm{Run}^\diamond_\mathcal{A}(p,s,p)} \mathrm{wt}^\diamond_\mathcal{A}(s, r) \qquad y = \max_{r \in \mathrm{Run}^\diamond_\mathcal{A}(q,s,q)} \mathrm{wt}^\diamond_\mathcal{A}(s, r),$$

we have $x = y$ whenever $x \neq -\infty$ and $y \neq -\infty$ holds. Here, the maximum over the empty set is $-\infty$ by convention.

A max-plus-WTA is said to satisfy the *twins property* if all of its siblings are twins. For unambiguous max-plus-WTA, the twins property is a criterion to decide the sequentiality problem. An unambiguous max-plus-WTA possesses a deterministic equivalent if and only if it satisfies the twins property. For words, this result is due to [75, Theorem 12]. For trees, we cite the following theorem which states that the twins property is a sufficient condition for determinizability.

Theorem 4.29 ([18, Lemma 5.10])**.** *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA. If $\mathcal{A}$ satisfies the twins property, there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$ which can be effectively constructed.*

The converse, namely that the twins property is also a necessary condition for determinizability, follows from the Lipschitz property of deterministic max-plus automata. For max-plus word automata, consider the following. If an unambiguous max-plus word automaton $\mathcal{A}$ does not satisfy the twins property, we can find states p and q which are siblings and not twins. We assume that our witnesses for this are u and s as above. Then we consider words of the form $w_1 = us^N v_p$ and $w_2 = us^N v_q$, where v_p and v_q are two fixed words which lead from p and q, respectively, to some final state. For every fixed $L \in \mathbb{R}$, we can choose N sufficiently large to ensure that $|[\![\mathcal{A}]\!](w_1) - [\![\mathcal{A}]\!](w_2)| > |L|(|v_p| + |v_q| + 2)$. Due to the Lipschitz property of deterministic max-plus-WA, it is thus not possible to determinize $\mathcal{A}$ if it does not satisfy the twins property. For trees, we can proceed in the same way.

Lemma 4.30. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA. If there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$, then $\mathcal{A}$ satisfies the twins property.*

Proof. We follow the idea for the proof of [75, Theorem 9]. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA and let $p, q \in Q$ be siblings, i.e., there exists a tree $u \in T_\Gamma$ and runs $r^p \in \mathrm{Run}_\mathcal{A}(u, p)$ and $r^q \in \mathrm{Run}_\mathcal{A}(u, q)$. Let $s \in T_{\Gamma_\diamond}$ be a Γ-word such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ for weights $x, y \in \mathbb{R}$. Since $\mathcal{A}$ is trim, there exist Γ-words $\hat{u}_p, \hat{u}_q \in T_{\Gamma_\diamond}$ such that $p \xrightarrow{\hat{u}_p|z_p} p'$ and $q \xrightarrow{\hat{u}_q|z_q} q'$ for two final states $p', q' \in Q$ and weights $z_p, z_q \in \mathbb{R}$. We let $\kappa_p = \mathrm{wt}_\mathcal{A}(u, r^p) + z_p + \nu(p')$ and $\kappa_q = \mathrm{wt}_\mathcal{A}(u, r^q) + z_q + \nu(q')$ and for $n \geq 1$ define the trees $t_p^{(n)} = \hat{u}_p(s^n(u))$ and $t_q^{(n)} = \hat{u}_q(s^n(u))$. Due to the

unambiguity of $\mathcal{A}$, we see that for every $n \geq 1$ we have

$$[\![\mathcal{A}]\!](t_p^{(n)}) = \kappa_p + nx$$
$$[\![\mathcal{A}]\!](t_q^{(n)}) = \kappa_q + ny.$$

Assume that there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. Then by Lemma 4.27, there exists $L \in \mathbb{R}$ such that for all $n \geq 1$ we have

$$|[\![\mathcal{A}]\!](t_p^{(n)}) - [\![\mathcal{A}]\!](t_q^{(n)})| \leq |L|(|\hat{u}_p| + |\hat{u}_q| + 2).$$

From the equations above we thus obtain that for every $n \geq 1$ we have

$$|\kappa_p - \kappa_q + n(x - y)| \leq |L|(|\hat{u}_p| + |\hat{u}_q| + 2).$$

This can only hold if $x = y$. It follows that $\mathcal{A}$ satisfies the twins property. □

The twins property is decidable for both max-plus word automata [2, 6, 75, 76, 60] and max-plus tree automata [17, Section 3]. Deciding whether a max-plus word automaton satisfies the twins property is PSPACE-complete [60]. For max-plus tree automata, the problem is thus PSPACE-hard, but no upper complexity bound is stated in [17]. We cite the following theorem.

Theorem 4.31 ([18, Theorem 5.17][17, Section 3])**.** *For an unambiguous max-plus-WTA $\mathcal{A}$ it is decidable whether $\mathcal{A}$ satisfies the twins property.*

Note that in general, it is undecidable whether two given siblings are twins [60], but for unambiguous max-plus automata (and even polynomially ambiguous max-plus automata), it was shown to be decidable on both words [2, Section 4] and trees [18, Section 5.4]. As we will need this statement in Section 4.6, we provide a short direct proof.

Lemma 4.32 ([18, Section 5.4])**.** *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be an unambiguous max-plus-WTA. For every two states $p, q \in Q$ it is decidable whether p and q are siblings and it is decidable whether p and q are twins.*

Proof. Let $p, q \in Q$ be two states. First, to check whether p and q are siblings, we see as follows that it suffices to check whether they can both be reached by a tree u of height at most $|Q|^2$. Assume we have a tree $u \in T_\Gamma$ and two runs $r^p \in \mathrm{Run}_\mathcal{A}(t, p)$ and $r^q \in \mathrm{Run}_\mathcal{A}(t, q)$. If $\mathrm{height}(u) > |Q|^2$, then by pigeon hole principle, we can find a simultaneous loop in r^p and r^q; that is, we can find two positions $w_1 <_p w_2$ in u with $r^p(w_1) = r^p(w_2)$ and also $r^q(w_1) = r^q(w_2)$. By removing everything between w_1 and w_2 from u, we obtain the smaller tree $u\langle u|_{w_2} \to w_1\rangle$ which still reaches p and q.

If p and q are siblings, we see as follows that we only need to check Γ-words s of height at most $4|Q|^2$ to decide whether p and q are twins. Assume p and q are not twins and our witness for this is the Γ-word s with $\mathrm{height}(s) > 4|Q|^2$. Let $r_p \in \mathrm{Run}^\diamond_\mathcal{A}(p, s, p)$ be the run on s which loops in p with weight $x = \mathrm{wt}^\diamond_\mathcal{A}(s, r_p)$ and let $r_q \in \mathrm{Run}^\diamond_\mathcal{A}(q, s, q)$ be the run on s which loops in q with weight $y = \mathrm{wt}^\diamond_\mathcal{A}(s, r_q)$. Furthermore, let $w \in \mathrm{pos}(s)$ with $|w| = \mathrm{height}(s)$ and let $w' \in \mathrm{pos}(s)$ be the longest common prefix of w and $\Diamond_1(s)$. Then either $|w'| > 2|Q|^2$ or $|w| - |w'| > 2|Q|^2$, or both.

In the first case, there exist two disjoint simultaneous loops in r_p and r_q above $\Diamond_1(s)$. More precisely, by pigeon hole principle we can find positions $w_1 <_p w_2 \leq_p w_3 <_p w_4$ with $w_4 \leq_p w' \leq_p \Diamond_1(s)$ in s for which $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$ and $(r_p(w_3), r_q(w_3)) = (r_p(w_4), r_q(w_4))$. In the second case, there exist two disjoint simultaneous loops in r_p and r_q which are prefix-independent from $\Diamond_1(s)$. That is, there exist positions $w_1 <_p w_2 \leq_p w_3 <_p w_4$ with $w' <_p w_1$ and $w_4 \leq_p w$ in s for which $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$ and $(r_p(w_3), r_q(w_3)) = (r_p(w_4), r_q(w_4))$.

Let x_{12} and x_{34} be the weights of the loops in the run r_p, and let y_{12} and y_{34} be the weights of the loops in the run r_q. We obtain a smaller Γ-word s' and runs r'_p and r'_q of distinct weights which loop in p and q, respectively, by removing either one of the two loops or both loops as follows. If $x - x_{12} \neq y - y_{12}$, we remove the w_1-w_2 loop. Otherwise, if $x - x_{34} \neq y - y_{34}$, we remove the w_3-w_4 loop. If we have both $x - x_{12} = y - y_{12}$ and $x - x_{34} = y - y_{34}$, we obtain that $2x - x_{12} - x_{34} = 2y - y_{12} - y_{34}$. From $x \neq y$, it follows that $x - x_{12} - x_{34} \neq y - y_{12} - y_{34}$, so we remove both loops. From the unambiguity of $\mathcal{A}$, we see that these two runs are the only runs on the smaller Γ-word, so we have found a smaller witness. □

Finally, we present the main theorem of this section, namely the decidability of the sequentiality problem for finitely ambiguous max-plus-WTA.

Theorem 4.33. *For a finitely ambiguous max-plus-WTA $\mathcal{A}$ it is decidable whether there exists a deterministic max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. If such an automaton $\mathcal{A}'$ exists, it can be effectively constructed.*

Proof. Let $\mathcal{A}$ be a finitely ambiguous max-plus-WTA. Due to Theorem 4.15 we can decide whether there exists an equivalent unambiguous max-plus-WTA. If this is not the case, $\mathcal{A}$ can also not be determinizable. Otherwise, we can effectively construct an unambiguous max-plus-WTA $\mathcal{A}'$ with $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$. Due to Theorem 4.31, we can decide whether $\mathcal{A}'$ satisfies the twins property, which according to Theorem 4.29 and Lemma 4.30 is equivalent to deciding whether $\mathcal{A}$ is determinizable. □

4.6 The Finite Sequentiality Problem

The finite sequentiality problem asks whether the behavior of a given max-plus-WTA is finitely sequential. In other words, the finite sequentiality problem asks whether for a given max-plus-WTA $\mathcal{A}$, there exist finitely many deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ such that $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$, where the maximum is taken pointwise. In this section, we show that the finite sequentiality problem is decidable for unambiguous max-plus-WTA. Moreover, if the behavior of an unambiguous max-plus-WTA is finitely sequential, we will obtain that the deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ can be effectively constructed. For the proof, we follow ideas from [5], where the decidability of the finite sequentiality problem was proved for unambiguous max-plus word automata. In [5], the *fork property* is shown to be a decidable criterion to determine the existence of a finitely sequential equivalent. More precisely, unambiguous max-plus word automata are shown to possess a finitely sequential representation if and only if they do not satisfy the fork property. It is shown elementarily that an unambiguous automaton satisfying the fork property cannot possess a finitely sequential equivalent. The proof for the existence of a finitely sequential representation in case that the fork property is not satisfied, on the other hand, relies on the construction of finitely many unambiguous max-plus automata whose pointwise maximum is equivalent to the original automaton, and which all satisfy the twins property. Since every unambiguous max-plus automaton which satisfies the twins property is determinizable [75], a finitely sequential representation is found by determinizing the unambiguous automata.

The general outline of our proof is similar to that of [5] and presents itself as follows. First, we generalize the fork property to the *tree fork property* by adding a condition which accounts for the nonlinear structure of trees. We then prove that an unambiguous max-plus tree automaton possesses a finitely sequential representation if and only if it does not satisfy the tree fork property. Like in the word case, we can use elementary proof methods to show that $[\![\mathcal{A}]\!]$ is not finitely sequential if $\mathcal{A}$ satisfies the tree fork property. To show that $[\![\mathcal{A}]\!]$ is finitely sequential if $\mathcal{A}$ does not satisfy the tree fork property, we also construct finitely many unambiguous max-plus tree automata which satisfy the twins property and which thus possess a deterministic equivalent. However, in comparison to [5] we need to take a different approach in order to obtain these automata. In [5], a modified *Schützenberger covering* [99, 95, 96] is first constructed from the unambiguous max-plus automaton, from which in turn an automaton is constructed which monitors the occurrence of certain states of the modified Schützenberger covering. This latter automaton is then decomposed into the finitely many unambiguous automata. This approach, however, is not applicable to trees, as the monitoring of states requires all relevant states to occur linearly. This happens trivially for word automata due to the inherent linear structure of words, but for tree automata examples can be found where relevant states occur nonlinearly. The approach we use here relies on constructing a max-plus automaton which tracks certain pairs of states of the original automaton. When applied to word automata, this immediately yields an automaton which can be decomposed into the desired unambiguous automata. Unfortunately, for tree automata this tracking of pairs of states again fails due to states occurring nonlinearly. Surprisingly however,

our construction can be applied to the Schützenberger covering of the original tree automaton, as the states relevant for tracking all occur pairwise linearly in the Schützenberger covering. The most difficult part of our proof is to show that the Schützenberger covering indeed has the property we just indicated.

For now, we introduce the *tree fork property* and show that it is decidable whether an unambiguous max-plus-WTA $\mathcal{A}$ satisfies this property. Recall that an unambiguous max-plus-WTA possesses a deterministic equivalent if and only if it satisfies the twins property, i.e., if and only if all of its siblings are twins (see Theorem 4.29 and Lemma 4.30). There exist unambiguous max-plus automata which cannot be determinized, but whose behavior is finitely sequential [63, Section 3.1], see also Figure 4.2. Thus, for the finite sequentiality problem we inevitably have to deal

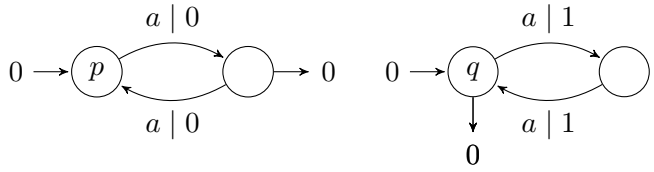


Figure 4.2: A max-plus word automaton $\mathcal{A}$ over the alphabet $\{a\}$ which is unambiguous, whose behavior is finitely sequential, but which does not satisfy the twins property as p and q are siblings but not twins. The behavior $[\![\mathcal{A}]\!]$ of $\mathcal{A}$ assigns 0 to all words of odd length and $|w|$ to all words w of even length.

with unambiguous automata in which not all siblings are twins. In the following, we will call two such states *rivals*. For unambiguous automata, which are the only type of max-plus-WTA we consider in this section, the following definition is equivalent to being siblings and not twins.

Definition 4.34. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. Two states $p, q \in Q$ are called *rivals* if there exists a tree $u \in T_\Gamma$ such that $\mathrm{Run}_\mathcal{A}(u, p) \neq \emptyset$ and $\mathrm{Run}_\mathcal{A}(u, q) \neq \emptyset$ and a Γ-word s such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ with $x \neq y$. In this case, u and s are also said to be *witnesses* for the fact that p and q are rivals.

We do not have to consider a maximum over runs here since $\mathcal{A}$ is unambiguous. Also note that by our definition of $\mathrm{Run}^\diamond_\mathcal{A}(s)$, we have $x \neq -\infty$ and $y \neq -\infty$ above.

Next, we introduce the tree fork property which, as we will show, is satisfied by an unambiguous max-plus-WTA if and only if its behavior is not finitely sequential. The property consists of two separate conditions. The first condition intuitively states that there exist two rivals p and q and a Γ-word t which can loop in p, and which can also lead from p to q. The second condition states that there exist two rivals which can occur at prefix-independent positions.

Definition 4.35. Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a max-plus-WTA. We say that $\mathcal{A}$ satisfies the *tree fork property* if at least one of the following two conditions is satisfied.

(i) There exist rivals $p, q \in Q$ and a Γ-word t with $p \xrightarrow{t|z_p} p$ and $p \xrightarrow{t|z_q} q$ for weights $z_p, z_q \in \mathbb{R}$. In this case, t is also called a *p-q-fork*.

(ii) There exist rivals $p, q \in Q$, a 2-Γ-context $t \in T_{\Gamma_\diamond}$, and a run $r \in \mathrm{Run}_{\mathcal{A}}^{\diamond}(t)$ with $r(\Diamond_1(t)) = p$ and $r(\Diamond_2(t)) = q$.

The tree fork property can be regarded as an extension of the *fork property* which was introduced in [5] and which for max-plus word automata plays the same role as the tree fork property does for max-plus tree automata. Condition (i) is essentially a tree version of the fork property. Casually put, if we take only condition (i) and replace "Γ-word" by "word", we obtain the fork property. The automaton depicted in Figure 4.3 is unambiguous and satisfies the fork property. Condition (ii) is new and possesses no counterpart in the fork property.

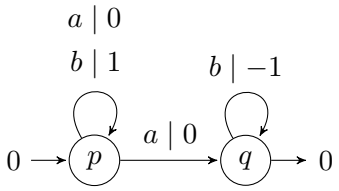


Figure 4.3: An unambiguous max-plus word automaton $\mathcal{A}$ over the alphabet $\{a, b\}$ which satisfies the fork property. With $u = a$ and $s = b$, we see that p and q are rivals, and a is a p-q-fork. All b's after the last a in a word are treated differently from the b's before the last a. A deterministic automaton cannot "guess" which a is the last in the word, and since there may be arbitrarily many a's in a word, even finitely many deterministic automata cannot compensate this inability to guess.

We have the following theorem which relates the tree fork property to the finite sequentiality problem.

Theorem 4.36. *Let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA over Γ. Then there exist deterministic max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$ if and only if $\mathcal{A}$ does not satisfy the tree fork property. If such automata $\mathcal{A}_1, \ldots, \mathcal{A}_n$ exist, they can be effectively constructed. In particular, the finite sequentiality problem is decidable for unambiguous max-plus-WTA.*

Proof. Here, we only show that it is decidable whether $\mathcal{A}$ satisfies the tree fork property. The rest of the proof is deferred to Sections 4.6.1 and 4.6.2, where we show that the behavior of $\mathcal{A}$ is finitely sequential if and only if $\mathcal{A}$ does not satisfy the tree fork property.

To decide whether $\mathcal{A}$ satisfies condition (i), we first show that if there exists a p-q-fork t for two rivals p and q, then there exists a p-q-fork t' of height at most $2|Q|^2$. The argumentation for this is similar to the proof of Lemma 4.32 that the property of being siblings is decidable. Assume that t is a p-q-fork with $\mathrm{height}(t) > 2|Q|^2$ and that r_p and r_q are runs that realize $p \xrightarrow{t|z_p} p$ and $p \xrightarrow{t|z_q} q$ for some weights $z_p, z_q \in \mathbb{R}$. We let $w \in \mathrm{pos}(t)$ be a position with $|w| = \mathrm{height}(t)$ and let w' be the longest common prefix of w and $\Diamond_1(t)$. Then either $|w'| > |Q|^2$ or $|w| - |w'| > |Q|^2$, or both. In the first case, there exist by pigeon hole principle two positions $w_1 <_p w_2$ in t with $w_2 \leq_p w' \leq_p \Diamond_1(t)$ and $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$. In the second case, there exist two positions

$w_1 <_p w_2$ in t with $w' <_p w_1$ and $(r_p(w_1), r_q(w_1)) = (r_p(w_2), r_q(w_2))$. By removing the part of t between w_1 and w_2, we obtain that $t' = t\langle t|_{w_2} \to w_1\rangle$ is a p-q-fork as well. Iterating this process, we obtain a p-q-fork of height at most $2|Q|^2$.

Next, we identify all pairs of rivals, which is possible since by Lemma 4.32, we can decide for every pair of states whether they are siblings and not twins. Then, for every pair of rivals p, q and all Γ-words t of height at most $2|Q|^2$, we check whether t is a p-q-fork. If this yields no p-q-fork, $\mathcal{A}$ does not satisfy condition (i).

In order to decide whether $\mathcal{A}$ satisfies condition (ii), we first compute the relation $\preceq$ on Q. This is possible since Q is a finite set and $\preceq$ is the smallest transitive and reflexive relation satisfying $\mu(q_1, \dots, q_m, a, q_0) \neq -\infty \to q_0 \preceq q_i$ for all transitions $(q_1, \dots, q_m, a, q_0) \in \Delta_{\mathcal{A}}$ and $i \in \{1, \dots, m\}$. Then, by the trimness of $\mathcal{A}$, condition (ii) is satisfied if and only if there exist two rivals p and q, a transition $(q_1, \dots, q_m, a, q_0) \in \Delta_{\mathcal{A}}$ with $\mu(q_1, \dots, q_m, a, q_0) \neq -\infty$, and indices $i, j \in \{1, \dots, m\}$ with $i \neq j$, $q_i \preceq p$, and $q_j \preceq q$. □

The following two sections are dedicated to completing the proof of Theorem 4.36.

4.6.1 Necessity

In this section, we show that if an unambiguous max-plus-WTA $\mathcal{A}$ satisfies either condition (i) or condition (ii) of the tree fork property, then $[\![\mathcal{A}]\!]$ is not finitely sequential. We begin with condition (i) and show that an unambiguous max-plus-WTA which satisfies condition (i) of the tree fork property does not possess a finitely sequential representation. The following theorem and its proof are an adaptation of [5, Theorem 2]. The proof relies on the Lipschitz property of deterministic max-plus automata (see Lemma 4.27) and its approach is similar to the proof that the twins property is a necessary condition for determinizability (see Lemma 4.30.

Theorem 4.37. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA over Γ. If $\mathcal{A}$ satisfies condition (i) of the tree fork property, then there do not exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$.*

Proof. For contradiction, assume that $\mathcal{A}$ satisfies condition (i) of the tree fork property and that there exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$. We write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ and let $N = \max_{i=1}^n |Q_i|$. Let p, q, t, z_p, z_q be as in condition (i) of the tree fork property and for the rivals p and q, let u, s, x, y be as in the definition of rivals. We let $r^p \in \mathrm{Run}_{\mathcal{A}}(u, p)$ and define $z_u = \mathrm{wt}_{\mathcal{A}}(u, r^p)$. Furthermore, by trimness there exists a Γ-word $\hat{u}$ with $q \xrightarrow{\hat{u}|z_{\hat{u}}} q_f$ for some weight $z_{\hat{u}} \in \mathbb{R}$ and some state $q_f \in Q$ with $\nu(q_f) \neq -\infty$.

We define the constant $L \in \mathbb{R}$ to be the largest weight, in terms of absolute value, which occurs in the automata $\mathcal{A}_1, \dots, \mathcal{A}_n$ as follows. We let $X = \bigcup_{i=1}^n \mu_i(\Delta_{\mathcal{A}_i}) \cup \nu_i(Q_i)$ and define $L = \max_{x \in X \setminus \{-\infty\}} |x|$. Furthermore, we define natural numbers $N_0, \dots, N_n$ inductively as follows. We let $N_n = 0$ and if $N_{l+1}, \dots, N_n$ are defined,

then we define N_l such that for all $k \in \{l+1, \dots, n\}$ we have

$$N_l|x-y| > L\Big((k-l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big) + (k-l)|z_p| + (\sum_{i=l+1}^{k-1} N_i|x|) + N_k|y|.$$

We define trees $t'_0, \dots, t'_n$ inductively by $t'_0 = s^{N_0}(t(u))$ and $t'_{k+1} = s^{N_{k+1}}(t(t'_k))$; for clarity, in the word case we would have $t'_k = uts^{N_0}ts^{N_1}\cdots ts^{N_k}$. Then for $k \in \{1, \dots, n\}$, we let $t_k = \hat{u}(t'_k)$. Due to the unambiguity of $\mathcal{A}$, we see that for every $k \in \{1, \dots, n\}$ we have

$$[\![\mathcal{A}]\!](t_k) = z_u + kz_p + (\sum_{i=0}^{k-1} N_i x) + z_q + N_k y + z_{\hat{u}} + \nu(q_f).$$

Thus, for $k > l$, we have

$$\begin{aligned}|[\![\mathcal{A}]\!](t_k) - [\![\mathcal{A}]\!](t_l)| &= |N_l(x-y) + (k-l)z_p + (\sum_{i=l+1}^{k-1} N_i x) + N_k y| \\ &\geq N_l|x-y| - (k-l)|z_p| - (\sum_{i=l+1}^{k-1} N_i|x|) - N_k|y| \\ &> L\Big((k-l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big).\end{aligned}$$

Note that the first inequality is an application of the reverse triangle inequality. The second inequality follows from the definition of N_l. Now let $j \in \{1, \dots, n\}$, then by choice of L and because $\mathcal{A}_j$ is deterministic, we have by Lemma 4.27 that

$$|[\![\mathcal{A}_j]\!](t_k) - [\![\mathcal{A}_j]\!](t_l)| \leq L\Big((k-l)|t| + (\sum_{i=l+1}^{k} N_i|s|) + 2|\hat{u}| + 2\Big).$$

In conclusion, we have $n+1$ trees t_i, and n automata $\mathcal{A}_i$, so by pigeonhole principle and the assumption that $[\![\mathcal{A}]\!] = \max_{i=1}^{n} [\![\mathcal{A}_i]\!]$, there must be $j \in \{1, \dots, n\}$ and $k, l \in \{0, \dots, n\}$ with $k > l$ such that $[\![\mathcal{A}]\!](t_k) = [\![\mathcal{A}_j]\!](t_k)$ and $[\![\mathcal{A}]\!](t_l) = [\![\mathcal{A}_j]\!](t_l)$. However, we have $|[\![\mathcal{A}]\!](t_k) - [\![\mathcal{A}]\!](t_l)| > |[\![\mathcal{A}_j]\!](t_k) - [\![\mathcal{A}_j]\!](t_l)|$, which is a contradiction. □

Next, we address condition (ii) of the tree fork property. On words, states cannot occur in prefix-independent positions. Thus, this condition is new for the tree case. Intuitively, the reason that the behavior of an unambiguous max-plus-WTA $\mathcal{A}$ cannot be finitely sequential if it satisfies condition (ii) is as follows. Assume we have a 2-Γ-context t and two rivals p and q as in condition (ii) and let u and s be as in the definition of rivals. Then we can construct trees of the form $t(s^n(u), s^n(u))$ such that, by increasing n, the difference between the weights on the two subtrees $s^n(u)$ is arbitrarily large. However, every deterministic automaton necessarily assigns the same weight to both subtrees.

Theorem 4.38. *Let $\mathcal{A}$ be a trim unambiguous max-plus-WTA over Γ. If $\mathcal{A}$ satisfies condition (ii) of the tree fork property, then there do not exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$.*

Proof. For contradiction, we assume that $\mathcal{A}$ satisfies condition (ii) of the tree fork property and that there exist deterministic max-plus-WTA $\mathcal{A}_1, \dots, \mathcal{A}_n$ over Γ with $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$. First, we construct a tree of the above mentioned form $t(s^n(u), s^n(u))$ and choose n large enough to ensure that in each of the deterministic automata, some sub-Γ-word s^m of s^n loops in some state. Then we show that every choice of a weight for such a loop leads to a contradiction.

Let p, q, t, r be as in condition (ii) of the tree fork property, $v_1 = \Diamond_1(t)$, and $v_2 = \Diamond_2(t)$. For the rivals p and q, let u and s be as in the definition of rivals and $v = \Diamond_1(s)$. We let $r_u^p \in \mathrm{Run}_{\mathcal{A}}(u, p)$, $r_u^q \in \mathrm{Run}_{\mathcal{A}}(u, q)$, $r_s^p \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(p, s, p)$, and $r_s^q \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(q, s, q)$. Furthermore, we write $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ and let $N = \max_{i=1}^n |Q_i|$.

By the following argument, we may assume that $\nu(r(\varepsilon)) \neq -\infty$. By trimness, there exists a Γ-word s'' and a run $r'' \in \mathrm{Run}_{\mathcal{A}}^{\Diamond}(s'')$ with $r''(\Diamond_1(s'')) = r(\varepsilon)$ and $\nu(r''(\varepsilon)) \neq -\infty$. Thus, if $\nu(r(\varepsilon)) = -\infty$, we can consider the 2-$\Gamma$-context $s''(t)$ with the run $r''\langle r \to \Diamond_1(s'')\rangle$ instead of t and r.

We now consider the tree $t' = t(s^N(u), s^N(u))$ together with the run

$$r' = r\langle (r_s^p)^{N\langle v\rangle} \langle r_u^p \to v^N\rangle \to v_1\rangle \langle (r_s^q)^{N\langle v\rangle} \langle r_u^q \to v^N\rangle \to v_2\rangle.$$

Since $r' \in \mathrm{Run}_{\mathcal{A}}(t')$ and $\nu(r'(\varepsilon)) \neq -\infty$, we have $[\![\mathcal{A}]\!](t') \neq -\infty$, so for some $j \in \{1, \dots, n\}$ we have $[\![\mathcal{A}_j]\!](t') = [\![\mathcal{A}]\!](t')$. By pigeonhole principle, since $N \geq |Q_j|$, we have $r'(v_1 v^{n_1}) = r'(v_1 v^{n_2})$ for some $n_1, n_2 \in \{0, \dots, N\}$ with $n_1 < n_2$. Since $\mathcal{A}_j$ is deterministic, we also obtain $r'(v_2 v^{n_1}) = r'(v_2 v^{n_2}) = r'(v_1 v^{n_1})$. Let $m = n_2 - n_1$ and let $x, y, z \in \mathbb{R}$ be the weights such that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$ in $\mathcal{A}$ and $r'(v_1 v^{n_1}) \xrightarrow{s^m|z} r'(v_1 v^{n_1})$ in $\mathcal{A}_j$. In particular, $x \neq y$. We may assume that $x < y$. We consider two cases.

First, if $z \geq \frac{m}{2}(x + y)$, then for the tree $t^+ = t(s^{N+m}(u), s^N(u))$ we obtain

$$\begin{aligned} \max_{i=1}^n [\![\mathcal{A}_i]\!](t^+) &\geq [\![\mathcal{A}_j]\!](t^+) = [\![\mathcal{A}_j]\!](t') + z \\ &\geq [\![\mathcal{A}_j]\!](t') + \frac{m}{2}(x + y) \\ &> [\![\mathcal{A}_j]\!](t') + mx = [\![\mathcal{A}]\!](t^+). \end{aligned}$$

For the other case, namely that $z \leq \frac{m}{2}(x + y)$, we see that for the tree $t^- = t(s^N(u), s^{N-m}(u))$ we obtain

$$\begin{aligned} \max_{i=1}^n [\![\mathcal{A}_i]\!](t^-) &\geq [\![\mathcal{A}_j]\!](t^-) = [\![\mathcal{A}_j]\!](t') - z \\ &\geq [\![\mathcal{A}_j]\!](t') - \frac{m}{2}(x + y) \\ &> [\![\mathcal{A}_j]\!](t') - my = [\![\mathcal{A}]\!](t^-). \end{aligned}$$

In both cases, we see that $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$ does not hold, which is a contradiction. □

Together, Theorems 4.37 and 4.38 show that if a trim unambiguous max-plus-WTA satisfies the tree fork property, then its behavior is not finitely sequential.

4.6.2 Sufficiency

In this section, we show that the behavior of an unambiguous max-plus-WTA $\mathcal{A}$ which does not satisfy the tree fork property is finitely sequential. For simplicity, we begin with a description of our method of proof on max-plus word automata and compare it to the proof method of Bala and Koniński [5].

Both proofs work by distributing the runs of $\mathcal{A}$ across a finite set of unambiguous max-plus word automata such that all of these automata satisfy the twins property. This distribution essentially has the aim of separating the rivals of $\mathcal{A}$. By Theorem 4.29, these unambiguous automata can then be determinized. The major difference between our approach and that of [5] lies in the way we obtain these unambiguous automata. To understand our approach, let p and q be two rivals of $\mathcal{A}$. Furthermore, let $u = u_1 \cdots u_n$ be a word for which there exist valid runs $r^p = p_0 \xrightarrow{u_1} p_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} p_{n-1} \xrightarrow{u_n} p$ and $r^q = q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \cdots \xrightarrow{u_{n-1}} q_{n-1} \xrightarrow{u_n} q$ of $\mathcal{A}$ on u. We also define $p_n = p$ and $q_n = q$.

We now show that the first occurrence of either p or q in the runs r^p and r^q serves as a "distinguisher" between the two runs. We let i be the smallest index with the property that $p_i \in \{p, q\}$. Similarly, we let j be the smallest index with the property that $q_j \in \{p, q\}$. We obtain valid runs $p_i \xrightarrow{u_{i+1} \cdots u_n} p$ and $q_j \xrightarrow{u_{j+1} \cdots u_n} q$.

Now assume it would hold that $i = j$ and $p_i = q_j$, i.e., the first occurrences are at the same position in the word, and also the states at this position are the same in both runs. Then with $t = u_{i+1} \cdots u_n$, we see that we have valid runs $p_i \xrightarrow{t} p$ and $p_i \xrightarrow{t} q$, where $p_i \in \{p, q\}$. Thus, $\mathcal{A}$ would satisfy the fork property. Since our assumption is that $\mathcal{A}$ does not satisfy the fork property, we have either $i \neq j$ or $p_i \neq q_j$.

This fundamental property is also used in the corresponding proof of [5], but our way of exploiting it differs from [5]. In their proof for word automata, Bala and Koniński use this property implicitly to show that certain states of a modified Schützenberger covering of $\mathcal{A}$ occur at most once in every run [5, Lemma 6]. They can therefore construct a new max-plus automaton which for each run keeps a record of all occurrences of these states. The above mentioned unambiguous automata are then obtained by separating runs with differing records into different automata. For tree automata, the number of these occurrences is unfortunately not bounded, for reasons which we will also indicate below.

For now, we continue outlining our new approach, which is to construct an automaton which adds a distinguishing marker to every run when first encountering one of the rivals p or q. This marker consists of a number, which is used to distinguish occurrences at different positions, and the state from $\{p, q\}$ which was visited first. Whenever reading a letter which causes some valid run to visit p or q for the first time, the automaton selects the smallest marker which was not used by any valid run on the prefix read so far, and annotates it to the run. For example, assume that neither p nor q occur in any valid run the word u, but that our run r on ua leads to p. Then r obtains the marker 1_p. Now assume there is a valid run on uaa which leads to p and which visited neither p nor q before that. Then this run obtains the marker 2_p, since 1_p is already assigned to r. Next, assume that after reading $uaaa$ another marker for p has to be assigned, and that r cannot be extended to a valid

run on uaa. Then we assign the marker 1_p, as now no valid run on uaa exists to which the marker 1_p is assigned. See Figure 4.4 for an example of this annotation process on the word aaa for the automaton depicted there.

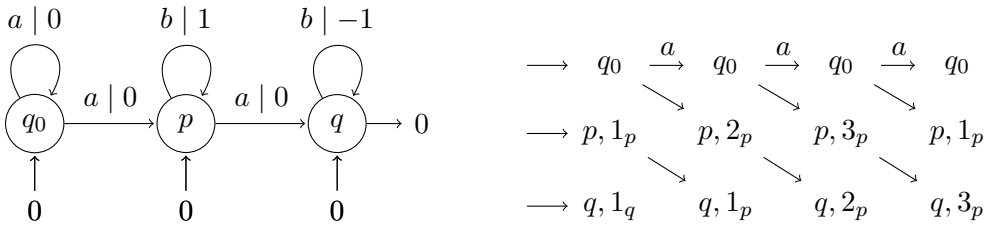


Figure 4.4: On the left, an unambiguous max-plus word automaton over the alphabet $\{a, b\}$ which does not satisfy the twins property but whose behavior is finitely sequential. On the right, an illustration of the runs of the automaton on the words ε, a, aa, and aaa together with appropriate markers. Arrows indicate a transition. The states p and q are rivals with witnesses $u = \varepsilon$ and $s = b$.

With this procedure, runs like r^p and r^q above receive different markers since either one run obtains a marker later than the other, and therefore a different marker, or at least the states they visit first are different, which also leads to different markers. To separate the rivals of $\mathcal{A}$, we can thus make a copy of $\mathcal{A}$ for every marker, and only allow runs which carry the respective automaton's marker. Whenever a different marker would be assigned, the execution of the run is blocked.

Note here that the number of markers we need for this annotation process is bounded. Since the automaton $\mathcal{A}$ is unambiguous, the number of *valid* runs on every given word is bounded by the number of states in $\mathcal{A}$. If this were not the case, there would exist two distinct valid runs on the same word which lead to the same state, from which a counterexample to the unambiguity of $\mathcal{A}$ could be constructed. In particular, the number of markers assigned at any given "time" is bounded by the number of states of $\mathcal{A}$.

All of this can easily be generalized to the situation where there is more than one pair of rivals. Then, runs simply obtain a marker for each pair of rivals of the automaton, and the copies of $\mathcal{A}$ allow a distinguished marker for each of these pairs.

Unfortunately, these ideas do not translate to trees as easily. For example, consider the runs in Figure 4.5. Intuitively, both runs should obtain the marker 1_p. However, since p and q are rivals, this marker does not serve the purpose of distinguishing runs as it does in the word case. The first p occurs in different subtrees of both runs, thus the annotation of distinct markers is not possible. Also, it is easy to construct an automaton where a rival p can occur at arbitrarily many pairwise prefix-independent positions, thus a simple lexicographic distinction is not possible. This is also the reason why the approach from [5] does not work for tree automata.

Our solution is to distribute not the runs of the automaton $\mathcal{A}$, but the runs of its *Schützenberger covering*. The Schützenberger covering of a max-plus automaton $\mathcal{A}$ is a max-plus automaton which possesses the same behavior as $\mathcal{A}$. It has already been employed in a number of decidability results for max-plus automata [63, 5, 4, 84]. Its

$$\begin{aligned} \nu(q) &= 0 \\ \mu(p,a,q) &= 0 \\ \mu(p,b,p) &= 1 \\ \mu(q,b,q) &= -1 \\ \mu(p,q_0,c,p) = \mu(q_0,p,c,q) &= 0 \\ \mu(d,p) = \mu(d,q_0) &= 0 \end{aligned}$$

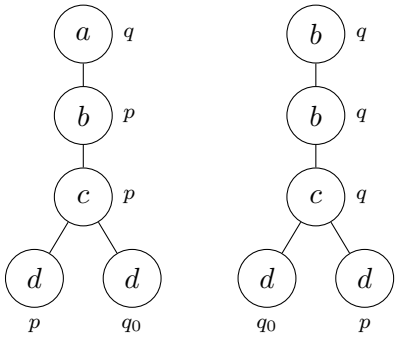


Figure 4.5: Two accepting runs of the max-plus tree automaton $\mathcal{A} = (\{q_0, p, q\}, \Gamma, \mu, \nu)$ over the ranked alphabet $\Gamma = \{a, b, c, d\}$ where $c \in \Gamma^{(2)}$, $a, b \in \Gamma^{(1)}$, and $d \in \Gamma^{(0)}$. All unspecified weights are assumed to be $-\infty$. The states p and q are rivals.

construction is inspired by a paper of Schützenberger [99] and was made explicit by Sakarovitch in [95], see also [96].

To better explain the idea behind its construction, we first point out a certain aspect of the classical powerset construction for finite automata [88]. Assume that $\mathcal{D}$ is the result of applying the powerset construction to an NFA $\mathcal{B}$. Then we might say that for a word $w = w_1 w_2$, the state which $\mathcal{D}$ is in after reading the prefix w_1 is the set of all states which $\mathcal{B}$ *could* be in after reading w_1. Similarly, the Schützenberger covering of a max-plus automaton $\mathcal{A}$ annotates to every state of a run of $\mathcal{A}$ on a word w the set of all states which "$\mathcal{A}$ could be in" at this point, i.e., which can be reached by some valid run on the considered prefix of w. Like the powerset construction, these ideas easily carry over to trees.

The reason we consider the Schützenberger covering of $\mathcal{A}$ is that each pair $\mathbf{p}, \mathbf{q}$ of its rivals satisfies the following property. For every tree t, either (1) $\mathbf{p}$ and $\mathbf{q}$ do not occur together in any run on t or (2) $\mathbf{p}$ and $\mathbf{q}$ occur only linearly, i.e., there is a distinguished branch of t such that for every run on t, all occurrences of $\mathbf{p}$ and $\mathbf{q}$ lie on this branch. In particular, the situation of Figure 4.5 is not possible. All pairs which satisfy the first condition can simply be separated into different automata, all pairs which satisfy the second condition can be handled like in the word case. The proof of this is non-trivial and needs some preparation. We begin with the formal definition of the Schützenberger covering.

For the rest of this section, let $\mathcal{A} = (Q, \Gamma, \mu, \nu)$ be a trim unambiguous max-plus-WTA which does not satisfy the tree fork property.

Definition 4.39 (Schützenberger covering, [95])**.** The Schützenberger covering $\mathcal{S} = (Q_\mathcal{S}, \Gamma, \mu_\mathcal{S}, \nu_\mathcal{S})$ of $\mathcal{A}$ is the trim part of the max-plus-WTA $(Q \times \mathcal{P}(Q), \Gamma, \mu', \nu')$ defined for $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and $(p_0, P_0), \ldots, (p_m, P_m) \in Q \times \mathcal{P}(Q)$ by

$$\mu'((p_1, P_1), \ldots, (p_m, P_m), a, (p_0, P_0)) =$$

$$\begin{cases} \mu(p_1, \ldots, p_m, a, p_0) & \text{if } P_0 = \{q_0 \in Q \mid \exists (q_1, \ldots, q_m) \in P_1 \times \ldots \times P_m \text{ with } \\ & \qquad \mu(q_1, \ldots, q_m, a, q_0) \neq -\infty\} \\ -\infty & \text{otherwise} \end{cases}$$

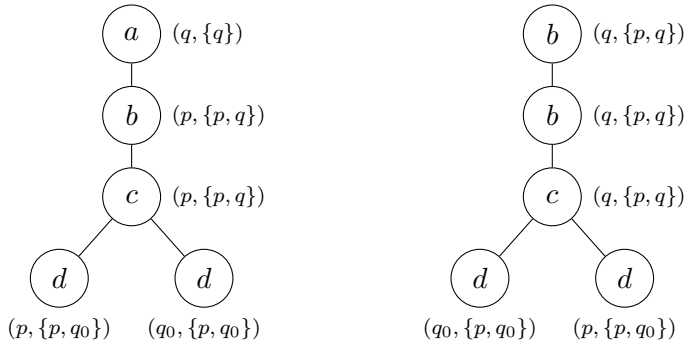


Figure 4.6: Two accepting runs of the Schützenberger covering of the automaton from Figure 4.5. The states $(p, \{p, q\})$ and $(q, \{p, q\})$ are rivals. The state $(p, \{p, q_0\})$ is not the rival of any state.

$$\nu'(p_0, P_0) = \nu(p_0).$$

We let $\pi_1 \colon Q \times \mathcal{P}(Q) \to Q$, $(p, P) \mapsto p$ and $\pi_2 \colon Q \times \mathcal{P}(Q) \to \mathcal{P}(Q)$, $(p, P) \mapsto P$ be the projections.

It is elementary to show that for a run of $\mathcal{S}$ on a tree t, the second entry of the state at a position w consists of all states of $\mathcal{A}$ which can be reached by a valid run of $\mathcal{A}$ on $t|_w$. In particular, every two runs on the same tree coincide on their second entries. Furthermore, projecting all states of a run of $\mathcal{S}$ to their first coordinate yields a run of $\mathcal{A}$, and the weights of these runs coincide. It follows that $\mathcal{S}$ is unambiguous and satisfies $[\![\mathcal{S}]\!] = [\![\mathcal{A}]\!]$. Also, $\mathcal{S}$ is trim by definition.

We can also make the following observation about the rivals of $\mathcal{S}$. Let $\mathbf{p}$ and $\mathbf{q}$ be rivals of $\mathcal{S}$ and let u and s be as in the definition of rivals. Since all runs of $\mathcal{S}$ on u coincide on the second entry of the state at the root, $\mathbf{p}$ and $\mathbf{q}$ also coincide on their second entry. Moreover, as projecting the runs of $\mathcal{S}$ on u and s to their first entries yields runs of $\mathcal{A}$ on u and s, respectively, we additionally see that the first entries of $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{A}$. Thus, if two states $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ are rivals in $\mathcal{S}$, then $\mathbf{p} = (p, P)$ and $\mathbf{q} = (q, P)$ for some set $P \subseteq Q$ and two states $p, q \in Q$ which are rivals in $\mathcal{A}$.

In the Schützenberger covering of the automaton from Figure 4.5, only the states $(p, \{p, q\})$ and $(q, \{p, q\})$ are rivals. See also Figure 4.6 for the runs of the Schützenberger covering on the trees from Figure 4.5. In the following lemma, we formally show that the properties we just described indeed hold for $\mathcal{S}$.

Lemma 4.40. *Let $t \in T_\Gamma$ be a tree. Then the following statements hold.*

(i) For every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ and position $w \in \mathrm{pos}(t)$ we have $\pi_2 \circ r(w) = \{p \in Q \mid \exists r' \in \mathrm{Run}_{\mathcal{A}}(t|_w, p)\}$.

(ii) For every two runs $r_1, r_2 \in \mathrm{Run}_{\mathcal{S}}(t)$, it holds that $\pi_2 \circ r_1 = \pi_2 \circ r_2$.

(iii) The projection π_1 induces a bijection $\pi_1 \colon \mathrm{Run}_{\mathcal{S}}(t) \to \mathrm{Run}_{\mathcal{A}}(t)$ by $r \mapsto \pi_1 \circ r$.

(iv) For every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ and every position $w \in \mathrm{pos}(t)$, we have $\pi_1 \circ r(w) \in \pi_2 \circ r(w)$.

(v) $\mathcal{S}$ *is trim, unambiguous, and satisfies* $[\![\mathcal{S}]\!] = [\![\mathcal{A}]\!]$.

(vi) For every Γ*-word* s *and two states* $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ *with* $\mathbf{p} \xrightarrow{s|x} \mathbf{q}$*, we have* $\pi_1(\mathbf{p}) \xrightarrow{s|x} \pi_1(\mathbf{q})$.

(vii) If two states $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ *are rivals in* $\mathcal{S}$*, then* $\mathbf{p} = (p, P)$ *and* $\mathbf{q} = (q, P)$ *for some set* $P \subseteq Q$ *and two states* $p, q \in Q$ *which are rivals in* $\mathcal{A}$.

Proof. (i) Let $t \in T_\Gamma$ and $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and for contradiction, let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (i) does not hold. We deduce that (i) holds for w. We let $a = t(w)$, $m = \mathrm{rk}_\Gamma(a)$, and write $r(w) = (p, P)$ and $r(wi) = (p_i, P_i)$ for $i \in \{1, \dots, m\}$.

First, let $q \in P$, then there are states $(q_1, \dots, q_m) \in P_1 \times \ldots \times P_m$ with $\mu(q_1, \dots, q_m, a, q) \neq -\infty$. By assumption, for every $i \in \{1, \dots, m\}$ we find a run $r_i \in \mathrm{Run}_{\mathcal{A}}(t{\restriction_{wi}}, q_i)$. Then the quasi-run $r' \colon \mathrm{pos}(t{\restriction_w}) \to Q$ defined by $r'(\varepsilon) = q$ and $r'(iv) = r_i(v)$ is a run of $\mathcal{A}$ on $t{\restriction_w}$ with $r'(\varepsilon) = q$.

On the other hand, let $r' \in \mathrm{Run}_{\mathcal{A}}(t{\restriction_w})$ and let $q = r'(\varepsilon)$. Then for every $i \in \{1, \dots, m\}$ we have that $r'{\restriction_i} \in \mathrm{Run}_{\mathcal{A}}(t{\restriction_{wi}})$, so by assumption, $r'(i) \in P_i$. Moreover, $\mu(r'(1), \dots, r'(m), a, q) \neq -\infty$, so $q \in P$. Thus, (i) holds for w, which is a contradiction, so w does not exist.

(ii) follows from (i).

(iii) Let $t \in T_\Gamma$. By definition of $\mu_{\mathcal{S}}$, it is clear that for $r \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\pi_1 \circ r \in \mathrm{Run}_{\mathcal{A}}(t)$. The injectivity of $\pi_1 \colon \mathrm{Run}_{\mathcal{S}}(t) \to \mathrm{Run}_{\mathcal{A}}(t)$ follows from (ii) since for every two runs $r_1, r_2 \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\pi_2 \circ r_1 = \pi_2 \circ r_2$. For surjectivity, we let $r' \in \mathrm{Run}_{\mathcal{A}}(t)$ and define a run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ inductively as follows. For a leaf $w \in \mathrm{pos}(t)$, we let $r(w) = (r'(w), \{p_0 \in Q \mid \mu(t(w), p_0) \neq -\infty\})$. For $w \in \mathrm{pos}(t)$ with $\mathrm{rk}_\Gamma(t(w)) = m$ such that r is defined on $w1, \dots, wm$ with $\pi_2 \circ r(wi) = P_i$, we let $r(w) = (r'(w), \{p_0 \in Q \mid \exists (p_1, \dots, p_m) \in P_1 \times \ldots \times P_m$ with $\mu(p_1, \dots, p_m, a, p_0) \neq -\infty\})$. Then $r \in \mathrm{Run}_{\mathcal{S}}(t)$ and $\pi_1 \circ r = r'$.

(iv) follows from (i) and (iii).

(v) $\mathcal{S}$ is trim by definition. Let $t \in T_\Gamma$. By definition of $\mu_{\mathcal{S}}$, for every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ we have $\mathrm{wt}_{\mathcal{S}}(t, r) = \mathrm{wt}_{\mathcal{A}}(t, \pi_1 \circ r)$. By definition of $\nu_{\mathcal{S}}$, we also have $\nu_{\mathcal{S}}(r(\varepsilon)) = \nu(\pi_1 \circ r(\varepsilon))$. By (iii), we thus have $|\mathrm{Acc}_{\mathcal{S}}(t)| = |\mathrm{Acc}_{\mathcal{A}}(t)| \leq 1$, which means $\mathcal{S}$ is unambiguous, and $[\![\mathcal{S}]\!](t) = [\![\mathcal{A}]\!](t)$.

(vi) Let s be a Γ-word and $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ be two states with $\mathbf{p} \xrightarrow{s|x} \mathbf{q}$, then there exists a run $r \in \mathrm{Run}^{\diamond}_{\mathcal{S}}(\mathbf{p}, s, \mathbf{q})$ with $\mathrm{wt}^{\diamond}_{\mathcal{S}}(s, r) = x$. By definition of $\mu_{\mathcal{S}}$, we have $\pi_1 \circ r \in \mathrm{Run}^{\diamond}_{\mathcal{A}}(s)$ and $\mathrm{wt}^{\diamond}_{\mathcal{S}}(s, r) = \mathrm{wt}^{\diamond}_{\mathcal{A}}(s, \pi_1 \circ r)$, so $\pi_1(\mathbf{p}) \xrightarrow{s|x} \pi_1(\mathbf{q})$.

(vii) Let $\mathbf{p}, \mathbf{q} \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ and write $\mathbf{p} = (p, P_p)$, $\mathbf{q} = (q, P_q)$. Let $u \in T_\Gamma$ and $s \in T_{\Gamma_\diamond}$ be as in the definition of rivals and let $r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{p})$ and $r^{\mathbf{q}} \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{q})$. By (ii), we have $P_p = \pi_2 \circ r^{\mathbf{p}}(\varepsilon) = \pi_2 \circ r^{\mathbf{q}}(\varepsilon) = P_q$. By (iii), we have $\pi_1 \circ r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{A}}(u, p)$ and $\pi_1 \circ r^{\mathbf{q}} \in \mathrm{Run}_{\mathcal{A}}(u, q)$, so p and q are siblings. Finally, from $(p, P_p) \xrightarrow{s|x} (p, P_p)$ and $(q, P_q) \xrightarrow{s|y} (q, P_q)$, we obtain by (vi) that $p \xrightarrow{s|x} p$ and $q \xrightarrow{s|y} q$. Since $x \neq y$, p and q are rivals in $\mathcal{A}$. □

In the theorems to follow, we will use fact (vii) of Lemma 4.40 without explicit further notice.

In order to prove some deeper results about the rivals of $\mathcal{S}$, we need two preparatory lemmata. As a first simplification, we show that we may assume that two rivals p and q of $\mathcal{A}$ are always comparable with respect to the relation $\preceq$. To see this, note that by condition (ii) of the tree fork property, p and q may not occur in prefix-independent positions in a run. If in addition, p and q can also not appear in prefix-dependent positions in a run, they never appear together in the same run of $\mathcal{A}$. Thus, we can create two copies of $\mathcal{A}$, one in which we remove p and one in which we remove q, and the pointwise maximum of these two automata will be equivalent to the behavior of $\mathcal{A}$.

Lemma 4.41. *We may assume that for all rivals $p, q \in Q$ we have either $p \preceq q$ or $q \preceq p$, or both.*

Proof. Let $p, q \in Q$ be rivals for which neither $p \preceq q$ nor $q \preceq p$. Then we can show that p and q never occur together in the same run as follows. Assume we have a tree $t \in T_\Gamma$, a run $r \in \mathrm{Run}_\mathcal{A}(t)$, and positions $w_1, w_2 \in \mathrm{pos}(t)$ with $r(w_1) = p$ and $r(w_2) = q$. Then w_1 and w_2 may not be prefix-independent since p and q are rivals, and by assumption $\mathcal{A}$ does not satisfy condition (ii) of the tree fork property. However, if w_1 and w_2 are prefix-dependent, we have a witness for either $p \preceq q$ or $q \preceq p$. This is a contradiction, and thus r as chosen does not exist.

We let $Q_1 = Q \setminus \{p\}$, $Q_2 = Q \setminus \{q\}$, and let $\mathcal{A}_i = (Q_i, \Gamma, \mu_i, \nu_i)$ for $i = 1, 2$, where μ_i and ν_i are the appropriate restrictions of μ and ν to the state sets Q_i. As p and q do not occur together in any run of $\mathcal{A}$, every run of $\mathcal{A}$ is also a run of at least one of the automata $\mathcal{A}_1, \mathcal{A}_2$. Thus, we have $[\![\mathcal{A}]\!] = \max_{i=1}^2 [\![\mathcal{A}_i]\!]$ and both $\mathcal{A}_1$ and $\mathcal{A}_2$ are trim and unambiguous and do not satisfy the tree fork property.

This procedure can be iterated to separate all rivals which are not in $\preceq$-relation. The termination of this procedure is guaranteed by the fact that the set of states becomes strictly smaller with every iteration. Eventually, we find trim unambiguous max-plus-WTA $\mathcal{A}_1, \ldots, \mathcal{A}_n$, all of which do not satisfy the tree fork property, such that $[\![\mathcal{A}]\!] = \max_{i=1}^n [\![\mathcal{A}_i]\!]$ and all rivals in an automaton $\mathcal{A}_i$ are pairwise in $\preceq$-relation. □

Next, we note an elementary statement about self-maps $f\colon X \to X$. Namely, if X is a finite set and $f\colon X \to X$ a mapping, then for every $a \in X$ there exists some element $b \in X$ and an integer $n \geq 1$ such that after n iterations of f, both a and b are mapped to b. To see this, consider the elements $a, f(a), f^2(a), \ldots, f^{|X|}(a)$. By pigeon hole principle, there are numbers $0 \leq m_1 < m_2 \leq |X|$ with $f^{m_1}(a) = f^{m_2}(a)$. Then if we choose $n \geq m_1$ as a multiple of $m_2 - m_1$ and $b = f^n(a)$, we see that $f^n(a) = b = f^n(b)$.

Lemma 4.42. *Let X be a finite set and $f\colon X \to X$ a mapping. Then for every $a \in X$, there exists an element $b \in X$ and an integer $n \geq 1$ with $f^n(a) = b = f^n(b)$. Here, f^n is the n-th iterate of f, i.e., $f^0 = \mathrm{id}_X$ and $f^{m+1} = f \circ f^m$.*

We now identify the first important property which all rivals of $\mathcal{S}$ satisfy. Namely, if $P \subseteq Q$ is the second entry of *some* rival, then it cannot occur in the form of a

"triangle" in any valid run of $\mathcal{S}$. More precisely, if we have a run r and positions w, wv_1, and wv_2 such that the second entry of $r(w)$, $r(wv_1)$, and $r(wv_2)$ is P, then wv_1 and wv_2 are prefix-dependent.

Lemma 4.43. *Let $(p,P),(q,P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$. Furthermore, let $t' \in T_\Gamma$ be a tree, $r' \in \mathrm{Run}_{\mathcal{S}}(t')$ a run of $\mathcal{S}$ on t', and $w_1, w_2 \in \mathrm{pos}(t')$ be positions in t'. If $\pi_2 \circ r'(\varepsilon) = \pi_2 \circ r'(w_1) = \pi_2 \circ r'(w_2) = P$, then w_1 and w_2 are prefix-dependent.*

Proof. We proceed by contradiction and assume that t', r', w_1, w_2 as in the statement of the lemma exist such that w_1 and w_2 are prefix-independent. We show that then, $\mathcal{A}$ satisfies condition (i) of the tree fork property. For the rivals (p,P) and (q,P), let u and s be as in the definition of rivals and let $v = \Diamond_1(s)$. As the proof is rather technical, we first provide a proof sketch and then follow up with a more precise presentation of the argumentation. See also Figure 4.7 for some visual aid.

By assumption, u can reach (p,P) and s can loop in (p,P), thus the trees $s^{|P|}(u)$ and $s^{|P|^{|P|}}(u)$ can reach (p,P). Due to the construction of $\mathcal{S}$, this means both of these trees can also reach the states of r' at w_1 and w_2. In particular, there exists a run of $\mathcal{S}$ on the tree $t = t'\langle s^{|P|}(u) \to w_1\rangle\langle s^{|P|^{|P|}}(u) \to w_2\rangle$ and for this run, the second entry of every state at the beginning or end of an s-loop is P. In addition, t leads to a state with second entry P, so there in fact exist $|P|$ runs of $\mathcal{S}$ on t, one for each state in P. We let $r_1, \dots, r_{|P|}$ be the projections of these runs to their first entry and obtain $|P|$ runs of $\mathcal{A}$ on t where for each run the state at the root and all states at the beginning or end of an s-loop are from P.

By pigeonhole principle, there is some subloop s^n below w_2 which loops in all runs at the same time, i.e., where for some n_1 we have $r_i(w_2v^{n_1}) = r_i(w_2v^{n_1+n})$ for all runs r_i. For each r_i, we let $q_i = r_i(w_2v^{n_1}) \in P$ be the state which r_i loops in and let x_i be the weight of this loop.

If $x_i \neq x_j$ for some i and j, the states q_i and q_j are rivals in $\mathcal{A}$ with witnesses u and s^n. By Lemma 4.41, we may therefore assume $q_i \preceq q_j$. Again by pigeon hole principle, the run r_i loops below w_1 in s^m for some $m \geq 1$ with some state $p_i \in P$, say with weight y_i. Due to $x_i \neq x_j$, we have $mx_i \neq ny_i$ or $mx_j \neq ny_i$. Since u can reach every state from P, the state p_i is thus a rival of q_i or q_j with witnesses u and s^{nm}. From the existence of r_i and the assumption that $q_i \preceq q_j$, we see that p_i can occur prefix-independently both from q_i and from q_j. This is a contradiction to the assumption that $\mathcal{A}$ does not satisfy the tree fork property. It must therefore hold that $x_1 = \ldots = x_{|P|}$.

We let x and y be the weights such that $\mathcal{A}$ loops s in p with weight x and in q with weight y. Then from $x \neq y$ it follows that $nx \neq x_1$ or $ny \neq x_1$, so the states q_i are either all rivals of p or all rivals of q with witnesses u and s^n. We assume all q_i to be rivals of p and apply Lemma 4.42 to the mapping $f\colon P \to \{q_1, \dots, q_{|P|}\}, r_i(\varepsilon) \mapsto q_i$ with $a = p$ to obtain $q_j \in P$ and $m \geq 1$ such that $f^m(p) = q_j = f^m(q_j)$. Then with $\tilde{s} = t\langle \diamond \to w_2v^{n_1}\rangle$, we see that the Γ-word $\tilde{s}^m$ is a q_j-p-fork, i.e., $\mathcal{A}$ satisfies condition (i) of the tree fork property.

We now turn to the more technical presentation of the proof. We define the tree $t = t'\langle s^{|P|}(u) \to w_1\rangle\langle s^{|P|^{|P|}}(u) \to w_2\rangle$ and construct a run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ of $\mathcal{S}$ on t as follows. By assumption, there exists a run $r^{\mathbf{P}} \in \mathrm{Run}_{\mathcal{S}}(u,(p,P))$ and a run

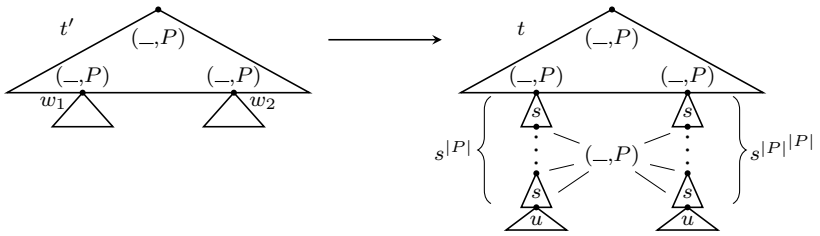


Figure 4.7: An illustration for the proof of Lemma 4.43.

$r_s \in \mathrm{Run}^{\diamond}_{\mathcal{S}}((p,P),s,(p,P))$. We let $r'_1 = r_s^{|P|\langle v\rangle}\langle r^{\mathbf{p}} \to v^{|P|}\rangle$ and $r'_2 = r_s^{|P|^{|P|}\langle v\rangle}\langle r^{\mathbf{p}} \to v^{|P|^{|P|}}\rangle$. Then $r'_1 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|}(u),(p,P))$ and $r'_2 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|^{|P|}}(u),(p,P))$.

By Lemma 4.40(iv), we have $\pi_1 \circ r'(w_1), \pi_1 \circ r'(w_2) \in P$, so by Lemma 4.40(i) we can find $r''_1 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|}(u))$ with $r''_1(\varepsilon) = \pi_1 \circ r'(w_1)$ and $r''_2 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|^{|P|}}(u))$ with $r''_2(\varepsilon) = \pi_1 \circ r'(w_2)$. Then $r = r'\langle \pi_1^{-1}(r''_1) \to w_1\rangle\langle \pi_1^{-1}(r''_2) \to w_2\rangle \in \mathrm{Run}_{\mathcal{S}}(t)$ is a run of $\mathcal{S}$ on t and we have $\pi_2 \circ r(w_1 v^i) = P$ for $0 \le i \le |P|$ and $\pi_2 \circ r(w_2 v^i) = P$ for $0 \le i \le |P|^{|P|}$.

By Lemma 4.40(i) and because $\pi_2 \circ r(\varepsilon) = P$, we can now find $|P|$ runs $r_1, \ldots, r_{|P|} \in \mathrm{Run}_{\mathcal{A}}(t)$ on t such that $\{r_1(\varepsilon), \ldots, r_{|P|}(\varepsilon)\} = P$. We have $r_j(w_2 v^i) \in P$ for every $j \in \{1,\ldots,|P|\}$ and every $i \in \{0,\ldots,|P|^{|P|}\}$. For each $i \in \{0,\ldots,|P|^{|P|}\}$, we define the tuple $\bar{q}_i = (r_1(w_2 v^i), \ldots, r_{|P|}(w_2 v^i))$. Since $\bar{q}_i \in P^{|P|}$ for every i, we can find $n_1 < n_2$ with $\bar{q}_{n_1} = \bar{q}_{n_2}$ by pigeonhole principle. Let $n = n_2 - n_1$ and write $\bar{q}_{n_1} = (q_1, \ldots, q_{|P|})$.

We now show that $q_1, \ldots, q_{|P|}$ are either all rivals of p, or they are all rivals of q. For this, note first that $q_j \xrightarrow{s^n|x_j} q_j$ for all $j \in \{1,\ldots,|P|\}$ with weights $x_1, \ldots, x_{|P|} \in \mathbb{R}$. Also, by the existence of the run $r^{\mathbf{p}}$ on u and Lemma 4.40(i), all states in P are siblings.

We show first that $x_1 = \ldots = x_{|P|}$. We assume that by contradiction, $x_i \neq x_j$ for some $i \neq j$. Then q_i and q_j are rivals in $\mathcal{A}$ with witnesses u and s^n. By Lemma 4.41, we can therefore assume that $q_i \preceq q_j$ or $q_j \preceq q_i$. We assume $q_i \preceq q_j$ and let s_j^i be a Γ-word such that there exists a run $r_j^i \in \mathrm{Run}^{\diamond}_{\mathcal{S}}(q_j, s_j^i, q_i)$. Furthermore, by pigeonhole principle, we can find $m_1, m_2 \in \{0,\ldots,|P|\}$ with $r_i(w_1 v^{m_1}) = r_i(w_2 v^{m_2})$ and $m_1 < m_2$. We let $p_i = r_i(w_1 v^{m_1})$ and $m = m_2 - m_1$ and show that p_i is a rival of either q_i or q_j. We have $p_i \xrightarrow{s^m|y_i} p_i$ for some weight $y_i \in \mathbb{R}$. Since $p_i \in P$, we know that p_i, q_i, and q_j are all siblings. Also, we have $p_i \xrightarrow{s^{nm}|ny_i} p_i$, $q_i \xrightarrow{s^{nm}|mx_i} q_i$, and $q_j \xrightarrow{s^{nm}|mx_j} q_j$. Since $x_i \neq x_j$, we have $ny_i \neq mx_i$ or $ny_i \neq mx_j$, or both. Thus, p_i is a rival of either q_i or of q_j.

Under these assumptions, we see that $\mathcal{A}$ satisfies condition (ii) of the tree fork property as follows. Either the 2-Γ-context $t_1 = t\langle \diamond \to w_1 v^{m_1}\rangle\langle \diamond \to w_2 v^{n_1}\rangle$ together with the run $r_i{\restriction}_{\mathrm{pos}(t_1)}$ or the 2-Γ-context $t_2 = t_1(\diamond, s_j^i)$ together with the run $r_i{\restriction}_{\mathrm{pos}(t_1)}\langle r_j^i \to \Diamond_2(t_1)\rangle$ is a witness for condition (ii) to be satisfied. Since our assumption for this section is that $\mathcal{A}$ does not satisfy the tree fork property, this is a contradiction. In conclusion, $x_1 = \ldots = x_{|P|}$.

To see that $q_1, \dots, q_{|P|}$ are either all rivals of p, or they are all rivals of q, consider the following. Using the same arguments as above, we find for every $i \in \{1, \dots, |P|\}$ a run $r^{q_i} \in \mathrm{Run}_{\mathcal{A}}(u, q_i)$. Furthermore, we have $p \xrightarrow{s^n|nx} p$, $q \xrightarrow{s^n|ny} q$, and $q_i \xrightarrow{s^n|x_1} q_i$ for every $i \in \{1, \dots, |P|\}$. Since $x \neq y$, we have either $nx \neq x_1$ or $ny \neq x_1$. Without loss of generality, we assume $nx \neq x_1$, thus all q_i are rivals of p.

We now show that $\mathcal{A}$ satisfies condition (i) of the tree fork property. We define a mapping $f\colon P \to \{q_1, \dots, q_{|P|}\}$ by $r_i(\varepsilon) \mapsto q_i$ for $i \in \{1, \dots, |P|\}$; recall that $\{q_1, \dots, q_{|P|}\} \subseteq P$, $\{r_1(\varepsilon), \dots, r_{|P|}(\varepsilon)\} = P$, and $r_i(\varepsilon) \neq r_j(\varepsilon)$ for $i \neq j$. By Lemma 4.42, there exists $m \geq 1$ and $i \in \{1, \dots, |P|\}$ with $f^m(p) = q_i = f^m(q_i)$. From this, we obtain that with $\tilde{s} = t\langle \diamond \to w_2 v^{n_1} \rangle$ we have $q_i \xrightarrow{\tilde{s}^m|z} q_i$ and $q_i \xrightarrow{\tilde{s}^m|z'} p$ for weights $z, z' \in \mathbb{R}$. As p and q_i are rivals, this means that $\mathcal{A}$ satisfies condition (i) of the tree fork property. □

In the previous lemma, we showed that if P is the second entry of some rival from $\mathcal{S}$, then states with second entry P do not occur in the form of a triangle. In the next lemma, we show that even prefix-independent occurrences are restricted to a certain degree. Namely, if we have two rivals (p, P) and (q, P) with $p \preceq q$, then all occurrences of P as a second entry are prefix-dependent on (p, P).

Lemma 4.44. *Let $(p, P), (q, P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ with $p \preceq q$. Furthermore, let $t' \in T_\Gamma$ be a tree, $r' \in \mathrm{Run}_{\mathcal{S}}(t')$ a run of $\mathcal{S}$ on t', and $w_1 \in \mathrm{pos}(t')$ a position in t' with $r'(w_1) = (p, P)$. Then all positions $w_2 \in \mathrm{pos}(t')$ with $\pi_2 \circ r'(w_2) = P$ are prefix-dependent on w_1.*

Proof. We proceed by contradiction and take $(p, P), (q, P), t', r', w_1$ as in the statement of the lemma and assume that there exists a position $w_2 \in \mathrm{pos}(t')$ which is prefix-independent from w_1 and for which $\pi_2 \circ r'(w_2) = P$. We show that under these assumptions, $\mathcal{A}$ satisfies condition (ii) of the tree fork property. For the rivals (p, P) and (q, P), let u and s be as in the definition of rivals and let $v = \Diamond_1(s)$. As in the proof of the previous lemma, we first provide a short proof sketch, see also Figure 4.8 for some visual aid.

As we have seen in the proof of Lemma 4.43, the tree $s^{|P|}(u)$ can reach (p, P), so due to the construction of $\mathcal{S}$, it can also reach the state of r' at w_2. Thus, there exists a run of $\mathcal{S}$ on the tree $t = t'\langle s^{|P|}(u) \to w_2 \rangle$ for which the state at w_1 is (p, P) and for which the second entry of every state at the beginning or end of an s-loop is P. We let r be the projection of this run to the first entries of the states.

By pigeonhole principle, we find some subloop s^n below w_2 in r which loops in a state $p' \in P$. Let z be the weight of this loop and let x and y be the weights such that $\mathcal{A}$ loops s in p with weight x and in q with weight y. Due to $x \neq y$, we have $nx \neq z$ or $ny \neq z$. Since u can reach every state from P, the state p' is a rival of p or q with witnesses u and s^n. From the fact that $r(w_1) = p$ and the assumption that $p \preceq q$, we see that p' can occur prefix-independently both from p and from q. This is a contradiction to the assumption that $\mathcal{A}$ does not satisfy the tree fork property.

In more detail, the proof is as follows. We define the tree $t = t'\langle s^{|P|}(u) \to w_2 \rangle$ and construct a run $r \in \mathrm{Run}_{\mathcal{A}}(t)$ of $\mathcal{A}$ on t as follows. By assumption, there

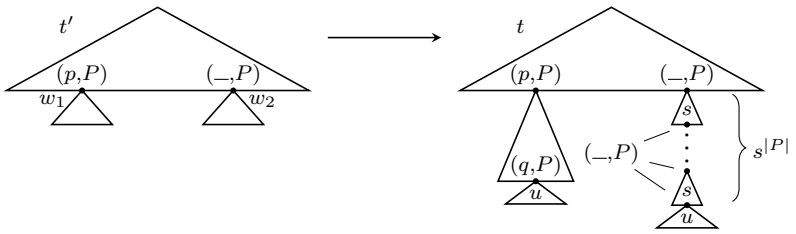


Figure 4.8: An illustration for the proof of Lemma 4.44.

exists a run $r^{\mathbf{p}} \in \mathrm{Run}_{\mathcal{S}}(u,(p,P))$ and a run $r_s \in \mathrm{Run}^{\diamond}_{\mathcal{S}}((p,P),s,(p,P))$. We let $r'_2 = r_s^{|P|\langle v\rangle}\langle r^{\mathbf{p}} \to v^{|P|}\rangle$. Then $r'_2 \in \mathrm{Run}_{\mathcal{S}}(s^{|P|}(u),(p,P))$.

By Lemma 4.40(iv), we have $\pi_1 \circ r'(w_2) \in P$, so by Lemma 4.40(i) we can find $r''_2 \in \mathrm{Run}_{\mathcal{A}}(s^{|P|}(u))$ with $r''_2(\varepsilon) = \pi_1 \circ r'(w_2)$. Then $r = \pi_1(r')\langle r''_2 \to w_2\rangle \in \mathrm{Run}_{\mathcal{A}}(t)$ is a run of $\mathcal{A}$ on t and we have $r(w_2v^i) \in P$ for $0 \le i \le |P|$.

By pigeonhole principle, we can find $n_1, n_2 \in \{0,\dots,|P|\}$ with $r(w_2v^{n_1}) = r(w_2v^{n_2})$ and $n_1 < n_2$. We let $p' = r(w_1v^{n_1})$ and $n = n_2 - n_1$ and show that p' is a rival of either p or q. We know that $p' \xrightarrow{s^n|z} p'$ for some weight $z \in \mathbb{R}$. Since $p' \in P$, we can also find a run $r^{p'} \in \mathrm{Run}_{\mathcal{A}}(u,p')$ which means that p' is a sibling of both p and q. We now have $p' \xrightarrow{s^n|z} p'$, $p \xrightarrow{s^n|nx} p$, and $q \xrightarrow{s^n|ny} q$. Since $x \neq y$, we have $nx \neq z$ or $ny \neq z$, or both. Thus, p' is a rival of either p or of q.

We see that $\mathcal{A}$ satisfies condition (ii) of the tree fork property as follows. Since we assumed $p \preceq q$, there exists a Γ-word s^p_q and a run $r^p_q \in \mathrm{Run}^{\diamond}_{\mathcal{A}}(q, s^p_q, p)$. Therefore, either the 2-Γ-context $t_1 = t\langle\diamond \to w_1\rangle\langle\diamond \to w_2v^{n_1}\rangle$ together with the run $r\restriction_{\mathrm{pos}(t_1)}$ or the 2-Γ-context $t_2 = t_1(s^p_q, \diamond)$ together with the run $r\restriction_{\mathrm{pos}(t_1)}\langle r^p_q \to \Diamond_1(t_1)\rangle$ is a witness for condition (ii) to be satisfied. Since our assumption for this section is that $\mathcal{A}$ does not satisfy the tree fork property, this is a contradiction. □

We can now prove that every run of $\mathcal{S}$ satisfies at least one of the following two conditions. If (p,P) and (q,P) are rivals in $\mathcal{S}$ with $p \preceq q$, then for every run r of $\mathcal{S}$ on a tree t either (i) (p,P) does not occur in r or (ii) all states with second entry P occur along a distinguished branch of t. This property enables us to apply the idea from the word case of using markers to indicate the first visit of a rival in a run. If u is a witness for (p,P) and (q,P) to be siblings, there is in particular a run on u which leads to (p,P). This run then satisfies condition (ii) and since by Lemma 4.40(ii) the second entries of runs on the same tree coincide, *all* states with second entry P occur along a distinguished branch of u in *every* run of $\mathcal{S}$ on u. This is true in particular for the two rivals (p,P) and (q,P).

Theorem 4.45. *Let $(p,P),(q,P) \in Q_{\mathcal{S}}$ be rivals in $\mathcal{S}$ with $p \preceq q$. Then for every tree $t \in T_\Gamma$ and every run $r \in \mathrm{Run}_{\mathcal{S}}(t)$ of $\mathcal{S}$ on t, at least one of the following two conditions holds.*

(i) The state (p,P) does not occur in r, i.e., $r(w) \neq (p,P)$ for all $w \in \mathrm{pos}(t)$.

(ii) All states with second entry P occur linearly in r, i.e., for all $w_1, w_2 \in \mathrm{pos}(t)$ with $\pi_2 \circ r(w_1) = \pi_2 \circ r(w_2) = P$ we have $w_1 \le_p w_2$ or $w_2 \le_p w_1$.

Proof. Let $(p, P), (q, P), t, r$ be as in the statement of the theorem. Assume that (i) does not hold, i.e., there is a position $w \in \mathrm{pos}(t)$ with $r(w) = (p, P)$. Let $w_1, w_2 \in \mathrm{pos}(t)$ be two positions with $\pi_2 \circ r(w_1) = \pi_2 \circ r(w_2) = P$. By Lemma 4.44, we see that then w_1 and w_2 are prefix-dependent on w. From the definition of the prefix relation, we see that if either $w_1 \leq_p w$ or $w_2 \leq_p w$, then all three positions are in prefix relation. We thus consider the case that $w \leq_p w_1$ and $w \leq_p w_2$. In this case, we see from Lemma 4.43 that w_1 and w_2 are prefix-dependent as follows. We write $w_1 = wv_1$ and $w_2 = wv_2$ and define $t' = t|_w$ and $r' = r|_w$. Then we have $r' \in \mathrm{Run}_{\mathcal{S}}(t')$, $r'(\varepsilon) = (p, P)$, and $\pi_2 \circ r'(v_1) = \pi_2 \circ r'(v_2) = P$. Thus, by Lemma 4.43 the positions v_1 and v_2 are prefix-dependent. □

In the following example, we illustrate some more complex interactions which may exist between rivals, in particular between the rivals of a Schützenberger covering.

Example 4.46. We extend the max-plus-WTA from Figure 4.5 to an automaton $\mathcal{A} = (\{q_0, p, p', p'', q\}, \Gamma, \mu, \nu)$ over the alphabet $\Gamma = \{a, b, c, d, e, f\}$ where $f \in \Gamma^{(3)}$, $c \in \Gamma^{(2)}$, $a, b, e \in \Gamma^{(1)}$, and $d \in \Gamma^{(0)}$. As this example is somewhat complex, we first give some intuition of what we are trying to show with the example and how we achieve this.

Let $P = \{p, p', p'', q\}$ and let $\mathcal{S}$ be the Schützenberger covering of $\mathcal{A}$. We construct $\mathcal{A}$ such that it satisfies the following conditions.

(i) $\mathcal{A}$ is unambiguous and does not satisfy the tree fork property. We achieve unambiguity simply by making $\mathcal{A}$ top-down deterministic.

(ii) The problem showcased in Figure 4.5 still occurs, i.e., a nonlinearity in the first occurrence of rivals.

(iii) The state q is a rival of all of p, p', and p''.

(iv) We have $p'' \preceq q \preceq p \preceq q \preceq p'$. In particular, we cannot trivially separate these states to different automata.

(v) In $\mathcal{S}$, the state (q, P) is a rival of all of $(p, P), (p', P)$, and (p'', P).

(vi) In $\mathcal{S}$, we have $(p'', P) \preceq (q, P) \preceq (p, P) \preceq (q, P)$, i.e., these three states cannot be trivially separated, and we have $(p'', P) \preceq (p', P)$.

(vii) In $\mathcal{S}$, the state (p', P) may occur at arbitrarily many pairwise prefix-independent positions in the same run.

The sole purpose of the letter c is to ensure condition (ii). The purpose of b is to ensure conditions (iii) and (v), the purpose of a is to ensure the first part of condition (vi), the purpose of e is to ensure the second part of condition (vi), and the purpose of f is to ensure condition (vii).

It is surprising that an automaton with the properties above exists since (1) Theorem 4.45 tells us that whenever (p'', P) occurs in a run, then all states with second entry P occur at pairwise prefix-dependent positions, (2) both (p'', P) and

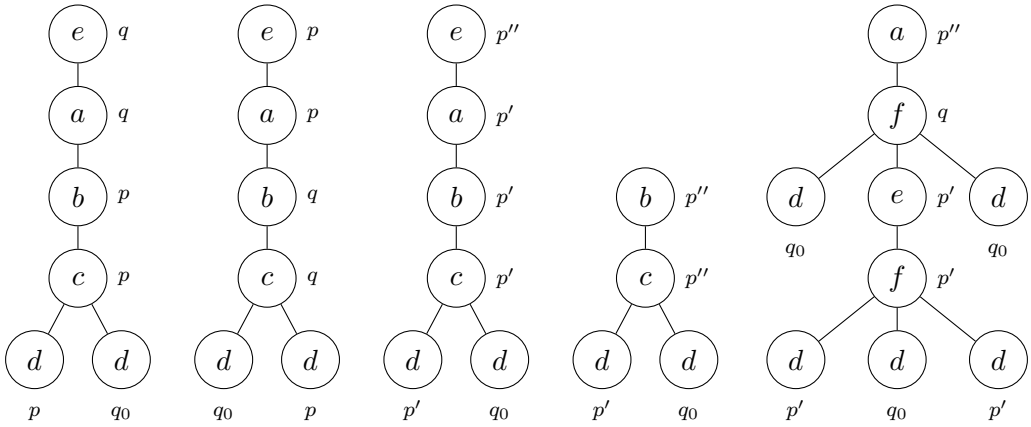


Figure 4.9: An illustration of the transitions of $\mathcal{A}$.

(p', P) may occur together in the same run, and (3) the state (p', P) may occur at two prefix-independent positions in the same run. We define μ and ν as follows.

$$\begin{aligned}
\mu(d, q_0) = \mu(d, p) = \mu(d, p') &= 0 \\
\mu(p, q_0, c, p) = \mu(q_0, p, c, q) = \mu(p', q_0, c, p') = \mu(p', q_0, c, p'') &= 0 \\
\mu(p, b, p) = \mu(p', b, p') = \mu(p'', b, p'') &= 1 \\
\mu(q, b, q) &= -1 \\
\mu(p, a, q) = \mu(q, a, p) = \mu(p', a, p') = \mu(q, a, p'') &= 0 \\
\mu(p, e, p) = \mu(q, e, q) = \mu(p', e, p') = \mu(p', e, p'') &= 0 \\
\mu(q_0, p', q_0, f, q) = \mu(p', q_0, p', f, p') &= 0 \\
\nu(p'') &= 0
\end{aligned}$$

All unspecified weights are $-\infty$. The trees in Figure 4.9 together with the runs given on them showcase the above transitions in a more graphical way. With witnesses $u = c(d, d)$ and $s = b(\diamond)$, we see that conditions (iii) and (v) above are satisfied. Due to $(q, P) \xrightarrow{a(\diamond)|0} (p, P) \xrightarrow{a(\diamond)|0} (q, P) \xrightarrow{a(\diamond)|0} (p'', P)$ and $(p', P) \xrightarrow{e(\diamond)|0} (p'', P)$, we see that condition (vi) is also satisfied. Let $P_0 = \{q_0, p, p'\}$, then the tree in Figure 4.10 together with the run of $\mathcal{S}$ on it illustrates that (p', P) may occur nonlinearly, i.e., condition (vii) is satisfied as well.

We note that the states p and p' are also rivals in $\mathcal{A}$ with witnesses $u = d$ and $s = a(b(a(\diamond)))$. Furthermore, $\mathcal{S}$ contains many more rivals than the ones mentioned above, among others the rivals $(p', \{p', q\})$ and $(q, \{p', q\})$ with witnesses $u = f(d, d, d)$ and $s = b(\diamond)$ and the rivals $(p, \{p, p', p''\})$ and $(p', \{p, p', p''\})$ with witnesses $u = a(f(d, d, d))$ and $s = a(b(a(\diamond)))$.

We are now ready to construct the automaton which tracks the first occurrences of rivals, and whose runs we will later distribute across multiple automata in order to separate all rivals.

Construction 4.47. Let $R_1, \ldots, R_n \subseteq Q_{\mathcal{S}}$ be an enumeration of all (unordered) pairs of rivals of $\mathcal{S}$, i.e., for all $i \in \{1, \ldots, n\}$ we have $R_i = \{(p_i, P_i), (q_i, P_i)\}$ such

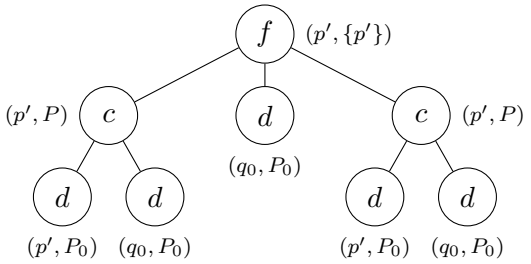


Figure 4.10: The state (p', P) may occur nonlinearly.

that (p_i, P_i) and (q_i, P_i) are rivals in $\mathcal{S}$ and for every two rivals $(p, P), (q, P) \in Q_{\mathcal{S}}$, we have $R_i = \{(p, P), (q, P)\}$ for some $i \in \{1, \ldots, n\}$. Since by Lemma 4.41, we may assume that all rivals in $\mathcal{A}$ are in $\preceq$-relation, we assume in the following that p_i and q_i are named such that $p_i \preceq q_i$ for all $i \in \{1, \ldots, n\}$.

For each pair of rivals R_i, we define a set of markers by $I_i = \{0, |Q| + 1\} \cup (\{1, \ldots, |Q|\} \times R_i)$. The set of all combined records of markers is defined by $I = I_1 \times \ldots \times I_n$. For $\bar{a} \in I$, we denote by $\bar{a}[i]$ the i-th entry of $\bar{a}$.

Intuitively, the states of our new automaton will consist of a state from $\mathcal{S}$ together with a record of markers from I. However, in order to properly update markers, we need to know in each step the records of *all* other runs as well. Thus, our states will be from $Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$.

In order to define the transition function of our new automaton, we first define how markers are updated. In some sense, this is similar to the *context successor* defined in [5]. Assume we transition into the state $\mathbf{q} \in Q_{\mathcal{S}}$, we have m subtrees below our current position in the tree, the runs we consider on these subtrees have obtained markers $\bar{a}_1, \ldots, \bar{a}_m \in I$, and the sets of states we *could* be in on these trees, together with their markers, are given by $A_1, \ldots, A_m \subseteq Q_{\mathcal{S}} \times I$.

Every pair $(\mathbf{p}, \bar{a}) \in A_k$ corresponds to exactly one run of $\mathcal{S}$ on the k-th subtree together with its markers. Since $\mathcal{S}$ is unambiguous, we can therefore assume that $|A_k| \leq |Q|$. Also, since $\bar{a}_k$ is the marker of a run on the k-th subtree, we may assume that $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$.

For $k \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, n\}$, we define the sets of unassigned counters $B_k[i] \subseteq \{1, \ldots, |Q|\}$ by

$$B_k[i] = \{1, \ldots, |Q|\} \setminus \{j \mid \exists (\mathbf{p}, \bar{a}) \in A_k \text{ with } \bar{a}[i] \in \{j\} \times R_i\}.$$

Then if for all $k \in \{1, \ldots, m\}$ we have $|A_k| \leq |Q|$ and $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$, we

define the record of markers $\bar{b}$ for our current position by (explanations below)

$$\bar{b}[i] = \begin{cases} 0 & \text{if } m = 0 \text{ and } \mathbf{q} \notin R_i \\ (1, \mathbf{q}) & \text{if } m = 0 \text{ and } \mathbf{q} \in R_i \\ \bar{a}_k[i] & \text{if } k \in \{1, \dots, m\} \text{ satisfies:} \\ & \bar{a}_l[i] = 0 \text{ for all } l \neq k \text{ and either } \bar{a}_k[i] \neq 0 \text{ or } \mathbf{q} \notin R_i \\ (\min B_k[i], \mathbf{q}) & \text{if } \mathbf{q} \in R_i \text{ and } k \in \{1, \dots, m\} \text{ satisfies:} \\ & \bar{a}_k[i] = 0 \text{ and for all } l \neq k \text{ and all } (\mathbf{p}, \bar{a}) \in A_l \colon \bar{a}[i] = 0 \\ |Q| + 1 & \text{otherwise} \end{cases}$$

for $i \in \{1, \dots, n\}$. If $|A_k| > |Q|$ or $Q_{\mathcal{S}} \times \{\bar{a}_k\} \cap A_k = \emptyset$ for some k, we let $\bar{b}[1] = \dots = \bar{b}[n] = |Q| + 1$.

Note that $\min B_k[i]$ in above case distinction always exists since $|A_k| \leq |Q|$, $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$, and in the case in question we have $\bar{a}_k[i] = 0$. We define $\mathcal{I}(\mathbf{q}, \bar{a}_1, \dots, \bar{a}_m, A_1, \dots, A_m) = \bar{b}$.

Case 1 of the definition above means our current position is a leaf and $\mathbf{q}$ is not from R_i, so we assign the dummy marker 0. Case 2 means our current position is a leaf and $\mathbf{q}$ is from R_i, so we assign the marker $(1, \mathbf{q})$. Case 3 means that either (1) there is exactly one subtree below our current position which already obtained a marker different from 0 and we keep this marker for our current position, or (2) the markers of all subtrees are 0 and $\mathbf{q}$ is also not from R_i, so we continue with the dummy marker 0.

Case 4 means the markers of all subtrees below our current position are 0, the state $\mathbf{q}$ is from R_i, and there is at most one subtree on which runs exist that obtained a marker for R_i. Then, we take the smallest number which is not already used in a marker for R_i in any run on this subtree, and use this number together with $\mathbf{q}$ as the marker for our current position.

Case 5, the "otherwise-case", applies in two situations. This case means that either (1) two distinct subtrees below our current position have already obtained a marker, or that (2) all markers below our current position are 0 and $\mathbf{q}$ is from R_i, but we cannot apply case 4 as there are two distinct subtrees on which runs exist which obtained markers for R_i. In other words, markers were assigned nonlinearly, and our run satisfies only condition (i) of Theorem 4.45. In this case, we assign the dummy marker $|Q| + 1$.

The extra case covers the situation where in case 4, the set $B_k[i]$ would be empty. This case is necessary to ensure our definition is formally complete, but in our applications of the operator $\mathcal{I}$ it will not actually occur.

We define our "run-marking" max-plus-WTA $\mathcal{B} = (\tilde{Q}, \Gamma, \tilde{\mu}, \tilde{\nu})$ as follows. We let $\tilde{Q}' = Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$ and let $\mathcal{B}$ be the trim part of the automaton $\mathcal{B}' = (\tilde{Q}', \Gamma, \tilde{\mu}', \tilde{\nu}')$ defined for $a \in \Gamma$ with $\mathrm{rk}_\Gamma(a) = m$ and $(\mathbf{p}_0, \bar{a}_0, A_0), \dots, (\mathbf{p}_m, \bar{a}_m, A_m) \in Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I)$ by

$$\tilde{\mu}'((\mathbf{p}_1, \bar{a}_1, A_1), \ldots, (\mathbf{p}_m, \bar{a}_m, A_m), a, (\mathbf{p}_0, \bar{a}_0, A_0)) =$$

$$\begin{cases} \mu_{\mathcal{S}}(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}_0) & \text{if } \bar{a}_0 = \mathcal{I}(\mathbf{p}_0, \bar{a}_1, \ldots, \bar{a}_m, A_1, \ldots, A_m) \text{ and} \\ & A_0 = \{(\mathbf{q}_0, \bar{b}_0) \in Q_{\mathcal{S}} \times I \mid \exists((\mathbf{q}_1, \bar{b}_1), \ldots, (\mathbf{q}_m, \bar{b}_m)) \in \\ & A_1 \times \ldots \times A_m \text{ with } \mu_{\mathcal{S}}(\mathbf{q}_1, \ldots, \mathbf{q}_m, a, \mathbf{q}_0) \neq -\infty \text{ and} \\ & \bar{b}_0 = \mathcal{I}(\mathbf{q}_0, \bar{b}_1, \ldots, \bar{b}_m, A_1, \ldots, A_m)\} \\ -\infty & \text{otherwise} \end{cases}$$

$$\tilde{\nu}'(\mathbf{p}_0, \bar{a}_0, A_0) = \nu_{\mathcal{S}}(\mathbf{p}_0).$$

For the rest of this section, we show that the automaton $\mathcal{B}$ "does what we want": We show that $\mathcal{B}$ is unambiguous, that it has the same behavior as $\mathcal{A}$, and that we can indeed separate its rivals by distributing runs with a different marker across different automata which then satisfy the twins property.

Let $\tilde{\pi}_1 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to Q_{\mathcal{S}}$, $(\mathbf{p}, \bar{a}, A) \mapsto \mathbf{p}$, $\tilde{\pi}_2 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to I$, $(\mathbf{p}, \bar{a}, A) \mapsto \bar{a}$, and $\tilde{\pi}_3 \colon Q_{\mathcal{S}} \times I \times \mathcal{P}(Q_{\mathcal{S}} \times I) \to \mathcal{P}(Q_{\mathcal{S}} \times I)$, $(\mathbf{p}, \bar{a}, A) \mapsto A$ be the projections. We prove the following basic observations about $\mathcal{B}$.

Lemma 4.48. *Let $t \in T_\Gamma$ be a tree. Then the following statements hold.*

(i) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ we have $(\tilde{\pi}_1 \circ r(w), \tilde{\pi}_2 \circ r(w)) \in \tilde{\pi}_3 \circ r(w)$. In particular, the only applications of the operator $\mathcal{I}$ are for sets A_k and tuples $\bar{a}_k$ with $(Q_{\mathcal{S}} \times \{\bar{a}_k\}) \cap A_k \neq \emptyset$.

(ii) For every two runs $r_1, r_2 \in \mathrm{Run}_{\mathcal{B}}(t)$ and every position $w \in \mathrm{pos}(t)$ we have $\tilde{\pi}_3 \circ r_1(w) = \tilde{\pi}_3 \circ r_2(w)$.

(iii) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and position $w \in \mathrm{pos}(t)$ we have $\tilde{\pi}_3 \circ r(w) = \{(\mathbf{q}, \bar{b}) \in Q_{\mathcal{S}} \times I \mid \exists r' \in \mathrm{Run}_{\mathcal{B}}(t{\restriction_w}) \text{ with } r'(\varepsilon) = (\mathbf{q}, \bar{b}, \tilde{\pi}_3 \circ r(w))\}$.

(iv) The projection $\tilde{\pi}_1$ induces a bijection $\tilde{\pi}_1 \colon \mathrm{Run}_{\mathcal{B}}(t) \to \mathrm{Run}_{\mathcal{S}}(t)$ by $r \mapsto \tilde{\pi}_1 \circ r$.

(v) $\mathcal{B}$ is trim, unambiguous, and satisfies $[\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$.

(vi) For every run $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and position $w \in \mathrm{pos}(t)$ we have $|\tilde{\pi}_3 \circ r(w)| \leq |Q|$. In particular, the only applications of the operator $\mathcal{I}$ are for sets A_k with $|A_k| \leq |Q|$.

(vii) For every Γ-word s and two states $\tilde{p}, \tilde{q} \in \tilde{Q}$ with $\tilde{p} \xrightarrow{s|x} \tilde{q}$, we have $\tilde{\pi}_1(\tilde{p}) \xrightarrow{s|x} \tilde{\pi}_1(\tilde{q})$.

Proof. (i) Let $t \in T_\Gamma$ and $r \in \mathrm{Run}_{\mathcal{B}}(t)$ and for contradiction, let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (i) does not hold. We let $m = \mathrm{rk}_\Gamma(t(w))$ and write $r(w) = (\mathbf{p}, \bar{a}, A)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \ldots, m\}$. Since r is a run of $\mathcal{B}$ on t, we have $\mu_{\mathcal{S}}(\mathbf{p}_1, \ldots, \mathbf{p}_m, a, \mathbf{p}) \neq -\infty$ and $\bar{a} = \mathcal{I}(\mathbf{p}, \bar{a}_1, \ldots, \bar{a}_m, A_1, \ldots, A_m)$. By assumption, we have $(\mathbf{p}_j, \bar{a}_j) \in A_j$ for all $j \in \{1, \ldots, m\}$, so $(\mathbf{p}, \bar{a}) \in A$ follows from the definition of $\tilde{\mu}$. This is a contradiction, thus w does not exist.

(ii) Let $t \in T_\Gamma$ and $r_1, r_2 \in \mathrm{Run}_{\mathcal{B}}(t)$ and let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (ii) does not hold. From the definition of $\tilde{\mu}$, it is immediately clear that $\tilde{\pi}_3 \circ r_1(w) = \tilde{\pi}_3 \circ r_2(w)$, so w does not exist.

(iii) Let $t \in T_\Gamma$ and $r \in \mathrm{Run}_\mathcal{B}(t)$ and let $w \in \mathrm{pos}(t)$ be a prefix-maximal position for which (iii) does not hold. We will deduce that (iii) holds for w. We let $m = \mathrm{rk}_\Gamma(t(w))$ and write $r(w) = (\mathbf{p}, \bar{a}, A)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \dots, m\}$.

First, let $(\mathbf{q}, \bar{b}) \in A$, then there are states $((\mathbf{q}_1, \bar{b}_1), \dots, (\mathbf{q}_m, \bar{b}_m)) \in A_1 \times \dots \times A_m$ with $\mu(\mathbf{q}_1, \dots, \mathbf{q}_m, a, \mathbf{q}) \neq -\infty$ and $\bar{b} = \mathcal{I}(\mathbf{q}, \bar{b}_1, \dots, \bar{b}_m, A_1, \dots, A_m)$. By assumption on w, for every j we find $r_j \in \mathrm{Run}_\mathcal{B}(t\restriction_{wj})$ with $r_j(\varepsilon) = (\mathbf{q}_j, \bar{b}_j, A_j)$. Then by definition of $\tilde{\mu}$, we see that the quasi-run $r' \colon \mathrm{pos}(t\restriction_w) \to \tilde{Q}$ defined by $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$ and $r'(jv) = r_j(v)$ is a run of $\mathcal{B}$ on $t\restriction_w$ with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$.

On the other hand, let $r' \in \mathrm{Run}_\mathcal{B}(t\restriction_w)$, with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$ for some $(\mathbf{q}, \bar{b}) \in Q_\mathcal{S} \times I$. Then from (i) we obtain $(\mathbf{q}, \bar{b}) \in A$. Thus, (iii) holds for w.

(iv) Let $t \in T_\Gamma$. By definition of $\tilde{\mu}$, it is clear that for $r \in \mathrm{Run}_\mathcal{B}(t)$ we have $\tilde{\pi}_1 \circ r \in \mathrm{Run}_\mathcal{S}(t)$. For the injectivity of $\tilde{\pi}_1 \colon \mathrm{Run}_\mathcal{B}(t) \to \mathrm{Run}_\mathcal{S}(t)$, let $r_1, r_2 \in \mathrm{Run}_\mathcal{B}(t)$ with $\tilde{\pi}_1 \circ r_1 = \tilde{\pi}_1 \circ r_2$. Let $w \in \mathrm{pos}(t)$ be a prefix-maximal position from the set $\{v \in \mathrm{pos}(t) \mid r_1(v) \neq r_2(v)\}$. Then $\tilde{\pi}_1 \circ r_1(w) = \tilde{\pi}_1 \circ r_2(w)$ and for all $j \in \{1, \dots, \mathrm{rk}_\Gamma(t(w))\}$ we have $r_1(wj) = r_2(wj)$. From the definition of $\tilde{\mu}$, it is immediately clear that $r_1(w) = r_2(w)$ follows, i.e., w as chosen does not exist.

For surjectivity, we let $r' \in \mathrm{Run}_\mathcal{S}(t)$ and define a run $r \in \mathrm{Run}_\mathcal{B}(t)$ inductively as follows. For a leaf $w \in \mathrm{pos}(t)$, we let $\mathbf{p} = r'(w)$, $\bar{a} = \mathcal{I}(\mathbf{p})$, $A = \{(\mathbf{q}_0, \mathcal{I}(\mathbf{q}_0) \mid \mu_\mathcal{S}(t(w), \mathbf{q}_0) \neq -\infty\}$, and $r(w) = (\mathbf{p}, \bar{a}, A)$.

Now let $w \in \mathrm{pos}(t)$ with $\mathrm{rk}_\Gamma(t(w)) = m$ such that r is defined on $w1, \dots, wm$. We write $\mathbf{p} = r'(w)$ and $r(wj) = (\mathbf{p}_j, \bar{a}_j, A_j)$ for $j \in \{1, \dots, m\}$. We let $\bar{a}_0 = \mathcal{I}(\mathbf{p}_0, \bar{a}_1, \dots, \bar{a}_m, A_1, \dots, A_m)$ and $A = \{(\mathbf{q}_0, \bar{b}_0) \in Q_\mathcal{S} \times I \mid \exists((\mathbf{q}_1, \bar{b}_1), \dots, (\mathbf{q}_m, \bar{b}_m)) \in A_1 \times \dots \times A_m$ with $\mu_\mathcal{S}(\mathbf{q}_1, \dots, \mathbf{q}_m, a, \mathbf{q}_0) \neq -\infty$ and $\bar{b}_0 = \mathcal{I}(\mathbf{q}_0, \bar{b}_1, \dots, \bar{b}_m, A_1, \dots, A_m)\}$, and $r(w) = (\mathbf{p}, \bar{a}, A)$. Thus, we obtain a run $r \in \mathrm{Run}_\mathcal{B}(t)$ with $\tilde{\pi}_1 \circ r(w) = r'$.

(v) $\mathcal{B}$ is trim by definition. Let $t \in T_\Gamma$. By definition of $\tilde{\mu}$, for every run $r \in \mathrm{Run}_\mathcal{S}(t)$ we have $\mathrm{wt}_\mathcal{B}(t, r) = \mathrm{wt}_\mathcal{S}(t, \tilde{\pi}_1 \circ r)$. By definition of $\tilde{\nu}$, we also have $\tilde{\nu}(r(\varepsilon)) = \nu(\tilde{\pi}_1 \circ r(\varepsilon))$. By (iv), we have $|\mathrm{Acc}_\mathcal{B}(t)| = |\mathrm{Acc}_\mathcal{S}(t)| \leq 1$, which means $\mathcal{B}$ is unambiguous, and we have $[\![\mathcal{B}]\!](t) = [\![\mathcal{S}]\!](t) = [\![\mathcal{A}]\!](t)$.

(vi) The automaton $\mathcal{A}$ is assumed to be trim and unambiguous, so we have $|\mathrm{Run}_\mathcal{A}(t)| \leq |Q|$ for every $t \in T_\Gamma$. Furthermore, the projections π_1 and $\tilde{\pi}_1$ are bijections by Lemma 4.40(iii) and (iv) above. Let $t \in T_\Gamma$, $r \in \mathrm{Run}_\mathcal{B}(t)$, and $w \in \mathrm{pos}(t)$. From (iii), we see that $|\tilde{\pi}_3 \circ r(w)| \leq |\mathrm{Run}_\mathcal{B}(t\restriction_w)| = |\mathrm{Run}_\mathcal{S}(t\restriction_w)| = |\mathrm{Run}_\mathcal{A}(t\restriction_w)| \leq |Q|$.

(vii) Let s be a Γ-word and $\tilde{p}, \tilde{q} \in \tilde{Q}$ be two states with $\tilde{p} \xrightarrow{s|x} \tilde{q}$, then there is a run $r \in \mathrm{Run}^\diamond_\mathcal{B}(\tilde{p}, s, \tilde{q})$ with $\mathrm{wt}^\diamond_\mathcal{B}(s, r) = x$. By definition of $\tilde{\mu}$, we have $\tilde{\pi}_1 \circ r \in \mathrm{Run}^\diamond_\mathcal{S}(s)$ and $\mathrm{wt}^\diamond_\mathcal{B}(s, r) = \mathrm{wt}^\diamond_\mathcal{S}(s, \tilde{\pi}_1(r))$, so we have $\tilde{\pi}_1(\tilde{p}) \xrightarrow{s|x} \tilde{\pi}_1(\tilde{q})$. □

Next, we prove two basic statements about how $\mathcal{B}$ sets markers. Assume we have some run in which a state $(\mathbf{p}, \bar{a}, A)$ occurs. First, we show that if $\bar{a}[i] \neq 0$ for some i, then in the past, we must have visited one of the rivals in R_i. Second, we show that if A contains a state $(\mathbf{q}, \bar{b})$ with $\bar{b}[i] \neq 0$ for some i, then we must have visited some state with second entry P_i in the past.

Lemma 4.49. *Let $t \in T_\Gamma$ be a tree, $r \in \mathrm{Run}_\mathcal{B}(t)$ be a run of $\mathcal{B}$ on t, let $w \in \mathrm{pos}(t)$ be a position in t, assume that $r(w) = (\mathbf{p}, \bar{a}, A)$, and let $i \in \{1, \ldots, n\}$. Then the following statements hold.*

(i) If $\bar{a}[i] \neq 0$, then there is a position $v \in \mathrm{pos}(t)$ with $w \leq_p v$ and $\tilde{\pi}_1 \circ r(v) \in R_i$.

(ii) If there exists $(\mathbf{q}, \bar{b}) \in A$ with $\bar{b}[i] \neq 0$, then there is a position $v \in \mathrm{pos}(t)$ with $w \leq_p v$ such that $\pi_2 \circ \tilde{\pi}_1 \circ r(v) = P_i$.

Proof. (i) Assume $\bar{a}[i] \neq 0$. We choose v prefix-maximal from the set $\{w' \in \mathrm{pos}(t) \mid w \leq_p w'$ and $r(w') = (\mathbf{q}, \bar{b}, B)$ with $\bar{b}[i] \neq 0\}$. This set is not empty since it contains w. We write $r(v) = (\mathbf{q}, \bar{b}, B)$. If $\mathbf{q} \notin R_i$ would hold, we see from the definition of $\tilde{\mu}$, the definition of the operator $\mathcal{I}$, and the fact that we chose v prefix-maximal from above set, that either case 1 or case 3 of the definition of $\mathcal{I}$ would apply in the definition of $\bar{b}[i]$. Thus, $\bar{b}[i] = 0$ would hold, which is not the case. Therefore, $\mathbf{q} \in R_i$ holds.

(ii) Assume there is $(\mathbf{q}, \bar{b}) \in A$ with $\bar{b}[i] \neq 0$. By Lemma 4.48(iii), there is a run $r' \in \mathrm{Run}_\mathcal{B}(t{\restriction_w})$ with $r'(\varepsilon) = (\mathbf{q}, \bar{b}, A)$. Then by (i), there exists $v \in \mathrm{pos}(t{\restriction_w})$ with $\tilde{\pi}_1 \circ r'(v) \in R_i$. Furthermore, we have $r{\restriction_w} \in \mathrm{Run}_\mathcal{B}(t{\restriction_w})$. Combining Lemma 4.48(iv) and Lemma 4.40(ii), we have $P_i = \pi_2 \circ \tilde{\pi}_1 \circ r'(v) = \pi_2 \circ \tilde{\pi}_1 \circ r{\restriction_w}(v)$. Thus, we see that $\pi_2 \circ \tilde{\pi}_1 \circ r(wv) = P_i$. □

Next, we essentially prove that markers for R_i are properly set in runs where states with P_i as a second entry occur only linearly. That is, we show that in these runs, a marker for R_i is only set when a rival from R_i is actually visited, and that it cannot be altered afterwards.

Lemma 4.50. *Let $t \in T_\Gamma$, $i \in \{1, \ldots, n\}$, and $r \in \mathrm{Run}_\mathcal{B}(t)$ such that for all positions $v_1, v_2 \in \mathrm{pos}(t)$ with $\pi_2 \circ \tilde{\pi}_1 \circ r(v_1) = \pi_2 \circ \tilde{\pi}_1 \circ r(v_2) = P_i$ we have $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. If $w \in \mathrm{pos}(t)$ is the prefix-largest position of t with $\tilde{\pi}_1 \circ r(w) \in R_i$ then the following properties are satisfied*

(i) The marker $\tilde{\pi}_2 \circ r(w)$ is defined using case 2 or case 4 of the definition of the operator $\mathcal{I}$.

(ii) For all positions $v \in \mathrm{pos}(t)$ with $v \leq_p w$ we have $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(w)[i] \in \{1, \ldots, |Q|\} \times \{\tilde{\pi}_1 \circ r(w)\}$.

(iii) For all positions $v \in \mathrm{pos}(t) \setminus \{w\}$ such that either v and w are prefix-independent or $w \leq_p v$, we have $\tilde{\pi}_2 \circ r(v)[i] = 0$.

Proof. Let $m = \mathrm{rk}_\Gamma(t(w))$. If $m = 0$, $\tilde{\pi}_2 \circ r(w)[i]$ is obviously defined using case 2. Otherwise, since w is the prefix-largest among all positions w' with $\tilde{\pi}_1 \circ r(w') \in R_i$, we have by Lemma 4.49(i) that $\tilde{\pi}_2 \circ r(v)[i] = 0$ for all $v \in \mathrm{pos}(t) \setminus \{w\}$ with $w \leq_p v$. In particular, we have $\tilde{\pi}_2 \circ r(wj)[i] = 0$ for all $j \in \{1, \ldots, m\}$. Thus, by Lemma 4.49(ii) and our assumptions on r and w, we see that case 4 of the definition of the operator $\mathcal{I}$ applies in the definition of $\tilde{\pi}_2 \circ r(w)[i]$. Thus, $\tilde{\pi}_2 \circ r(w)[i] \in \{1, \ldots, |Q|\} \times \{\tilde{\pi}_1 \circ r(w)\}$.

We show (ii). For contradiction, let $v \in \mathrm{pos}(t)$ be the prefix-largest position with $v \leq_p w$ and $\tilde{\pi}_2 \circ r(v)[i] \neq \tilde{\pi}_2 \circ r(w)[i]$. Let $m = \mathrm{rk}_\Gamma(t(v))$ and $j \in \{1, \ldots, m\}$ such that $w = vjv'$ for some v'. Then $\tilde{\pi}_2 \circ r(vj)[i] = \tilde{\pi}_2 \circ r(w)[i] \neq 0$, and by Lemma 4.49(i) and our assumption on r, we have $\tilde{\pi}_2 \circ r(vk)[i] = 0$ for all $k \neq j$. Thus, case 3 of the definition of $\mathcal{I}$ applies in the definition of $\tilde{\pi}_2 \circ r(v)[i]$, so $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(vj)[i] = \tilde{\pi}_2 \circ r(w)[i]$. This means v as chosen does not exist.

Finally let $v \in \mathrm{pos}(t) \setminus \{w\}$ be such that either v and w are prefix-independent or $w \leq_p v$. Then from Lemma 4.49(i) and our assumption on r we immediately obtain $\tilde{\pi}_2 \circ r(v)[i] = 0$. □

In the next lemma, we show that if two states are rivals in $\mathcal{B}$, then their records of markers differ. The reasoning for this is exactly the same as in our intuitive description at the beginning of this section.

Lemma 4.51. *If $(\mathbf{p}, \bar{a}, A)$ and $(\mathbf{q}, \bar{b}, B)$ are rivals in $\mathcal{B}$, then $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{S}$ and for $i \in \{1, \ldots, n\}$ with $R_i = \{\mathbf{p}, \mathbf{q}\}$, we have $\bar{a}[i] \neq \bar{b}[i]$ and $\bar{a}[i], \bar{b}[i] \in \{1, \ldots, |Q|\} \times \{\mathbf{p}, \mathbf{q}\}$.*

Proof. Let $\tilde{p} = (\mathbf{p}, \bar{a}, A)$ and $\tilde{q} = (\mathbf{q}, \bar{b}, B)$ be rivals in $\mathcal{B}$. Let u and s be as in the definition of rivals and let $r^{\tilde{p}} \in \mathrm{Run}^\diamond_{\mathcal{B}}(u, \tilde{p})$ and $r^{\tilde{q}} \in \mathrm{Run}^\diamond_{\mathcal{B}}(u, \tilde{q})$. Then by Lemma 4.48(iv), we have $\tilde{\pi}_1(r^{\tilde{p}}) \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{p})$ and $\tilde{\pi}_1(r^{\tilde{q}}) \in \mathrm{Run}_{\mathcal{S}}(u, \mathbf{q})$. By Lemma 4.48(vii), we also have $\mathbf{p} \xrightarrow{s|x} \mathbf{p}$ and $\mathbf{q} \xrightarrow{s|y} \mathbf{q}$ with $x \neq y$, thus $\mathbf{p}$ and $\mathbf{q}$ are rivals in $\mathcal{S}$. Let $i \in \{1, \ldots, n\}$ with $R_i = \{\mathbf{p}, \mathbf{q}\}$. We may assume that $\mathbf{p} = (p_i, P_i)$ and $\mathbf{q} = (q_i, P_i)$.

We show that $\bar{a}[i], \bar{b}[i] \notin \{0, |Q| + 1\}$. We let $r^{\mathbf{p}} = \tilde{\pi}_1 \circ r^{\tilde{p}}$ and $r^{\mathbf{q}} = \tilde{\pi}_1 \circ r^{\tilde{q}}$. We have $r^{\mathbf{p}}(\varepsilon) = (p_i, P_i)$ and we assumed $p_i \preceq q_i$, so by Theorem 4.45 we obtain that for every two positions $v_1, v_2 \in \mathrm{pos}(u)$ with $\pi_2 \circ r^{\mathbf{p}}(v_1) = \pi_2 \circ r^{\mathbf{p}}(v_2) = P_i$, we have $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. This also holds for $r^{\mathbf{q}}$ since by by Lemma 4.40(ii) we have $\pi_2 \circ r^{\mathbf{p}} = \pi_2 \circ r^{\mathbf{q}}$.

Let $w_p \in \mathrm{pos}(u)$ be the prefix-largest position of u with $r^{\mathbf{p}}(w_p) \in R_i$ and $w_q \in \mathrm{pos}(u)$ be the prefix-largest position with $r^{\mathbf{q}}(w_q) \in R_i$. That w_p and w_q exist is clear from the fact that $r^{\mathbf{p}}(\varepsilon) \in R_i$ and $r^{\mathbf{q}}(\varepsilon) \in R_i$. By Lemma 4.50(ii), we have $\bar{a}[i] = \tilde{\pi}_2 \circ r^{\tilde{p}}(w_p)[i] \in \{1, \ldots, |Q|\} \times \{\tilde{\pi}_1 \circ r^{\tilde{p}}(w_p)\}$ and $\bar{b}[i] = \tilde{\pi}_2 \circ r^{\tilde{q}}(w_q)[i] \in \{1, \ldots, |Q|\} \times \{\tilde{\pi}_1 \circ r^{\tilde{q}}(w_q)\}$.

We show that $\bar{a}[i] \neq \bar{b}[i]$ and consider two cases. First, if $w_p = w_q$ we assume for contradiction that $\bar{a}[i] = \bar{b}[i]$. Then we see that $r^{\mathbf{p}}(w_p) = r^{\mathbf{q}}(w_q) \in R_i$, and we also have $r^{\mathbf{p}}(\varepsilon) = \mathbf{p}$ and $r^{\mathbf{q}}(\varepsilon) = \mathbf{q}$. It follows that with $s = u\langle \diamond \to w_p \rangle$, we have $\pi_1 \circ r^{\mathbf{p}}(w_p) \xrightarrow{s|z_1} p_i$ and $\pi_1 \circ r^{\mathbf{p}}(w_p) \xrightarrow{s|z_2} q_i$ for weights $z_1, z_2 \in \mathbb{R}$. Thus, $\mathcal{A}$ satisfies condition (i) of the tree fork property, which is a contradiction. Therefore, $\bar{a}[i] = \bar{b}[i]$ cannot hold when $w_p = w_q$.

Now assume without loss of generality that $w_p \leq_p w_q$ with $w_p \neq w_q$ and write $w_q = w_p jv$ and $r^{\tilde{q}}(w_p j) = (\mathbf{q}', \bar{b}', A_j)$. By Lemma 4.48(i) and Lemma 4.48(ii), we then have $(\mathbf{q}', \bar{b}') \in A_j = \tilde{\pi}_3 \circ r^{\tilde{p}}(w_p j)$. By Lemma 4.50(i), we know that $\tilde{\pi}_2 \circ r^{\tilde{p}}(w_p)[i]$ is defined using case 4 of the definition of $\mathcal{I}$, so $\bar{a}[i] \neq \bar{b}[i]$ must hold. □

We turn to our final construction where we distribute the runs of $\mathcal{B}$ across multiple automata. For every record of markers $\bar{c} \in I$, we construct one automaton $\mathcal{B}_{\bar{c}}$ which

for each pair of rivals R_i admits only runs using the markers 0 and $\bar{c}[i]$. All runs in which rivals occur nonlinearly are covered by admitting the marker $|Q|+1$. All other runs are covered by admitting an appropriate marker from $\{1,\dots,|Q|\} \times R_i$.

Construction 4.52. For every tuple $\bar{c} \in I$, we define a max-plus-WTA $\mathcal{B}_{\bar{c}} = (\tilde{Q}_{\bar{c}}, \Gamma, \tilde{\mu}, \tilde{\nu})$ by removing states from $\mathcal{B}$ through

$$\begin{aligned}\tilde{Q}_{\bar{c}} = \{(\mathbf{p}, \bar{a}, A) \in \tilde{Q} \mid \text{for all } i \in \{1,\dots,n\} \text{ it holds:}\\ \text{if } \bar{c}[i] = |Q|+1 \text{ then } \mathbf{p} \neq (p_i, P_i) \text{ and}\\ \text{if } \bar{c}[i] \neq |Q|+1 \text{ then } \bar{a}[i] \in \{0, \bar{c}[i]\}\}.\end{aligned}$$

Finally, we formally prove that the automata $\mathcal{B}_{\bar{c}}$ are unambiguous, that their pointwise maximum is equivalent to the behavior of $\mathcal{A}$, and that they all satisfy the twins property, which means that they can be determinized.

Theorem 4.53. *We have $[\![\mathcal{A}]\!] = \max_{\bar{c}\in I}[\![\mathcal{B}_{\bar{c}}]\!]$ and for every $\bar{c} \in I$, the automaton $\mathcal{B}_{\bar{c}}$ is unambiguous and satisfies the twins property.*

Proof. The unambiguity of $\mathcal{B}_{\bar{c}}$ follows from the unambiguity of $\mathcal{B}$. To see that $\mathcal{B}_{\bar{c}}$ satisfies the twins property, let $(\mathbf{p}, \bar{a}, A), (\mathbf{q}, \bar{b}, B) \in \tilde{Q}_{\bar{c}}$ be rivals in $\mathcal{B}_{\bar{c}}$. Then $(\mathbf{p}, \bar{a}, A)$ and $(\mathbf{q}, \bar{b}, B)$ are also rivals in $\mathcal{B}$, so by Lemma 4.51 for some $i \in \{1,\dots,n\}$ we have $\bar{a}[i] \neq \bar{b}[i]$ and $\bar{a}[i], \bar{b}[i] \notin \{0, |Q|+1\}$. By definition of $\mathcal{B}_{\bar{c}}$, this means $(\mathbf{p}, \bar{a}, A), (\mathbf{q}, \bar{b}, B) \in \tilde{Q}_{\bar{c}}$ is impossible, so there are no rivals in $\mathcal{B}_{\bar{c}}$ and $\mathcal{B}_{\bar{c}}$ satisfies the twins property.

To show that $[\![\mathcal{A}]\!] = \max_{\bar{c}\in I}[\![\mathcal{B}_{\bar{c}}]\!]$, we show that for every tree $t \in T_\Gamma$ we have $\mathrm{Run}_{\mathcal{B}}(t) = \bigcup_{\bar{c}\in I} \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. From this, it follows that $\max_{\bar{c}\in I}[\![\mathcal{B}_{\bar{c}}]\!] = [\![\mathcal{B}]\!] = [\![\mathcal{A}]\!]$. The inclusion "$\supseteq$" is clear.

Let $t \in T_\Gamma$, $r \in \mathrm{Run}_{\mathcal{B}}(t)$, and let $O = \{i \in \{1,\dots,n\} \mid \text{there is a position } w \in \mathrm{pos}(t) \text{ with } \tilde{\pi}_1 \circ r(w) = (p_i, P_i)\}$. Let $i \in O$ and assume we have two positions $v_1, v_2 \in \mathrm{pos}(t)$ such that $\pi_2 \circ \tilde{\pi}_1 \circ r(v_1) = \pi_2 \circ \tilde{\pi}_1 \circ r(v_2) = P_i$. Then, since $\tilde{\pi}_1(r) \in \mathrm{Run}_{\mathcal{S}}(t)$ by Lemma 4.48(iv), we obtain by Theorem 4.45 that $v_1 \leq_p v_2$ or $v_2 \leq_p v_1$. We can therefore let $w_i \in \mathrm{pos}(t)$ be the prefix-largest position in t with $\tilde{\pi}_1 \circ r(w_i) \in R_i$. Then from Lemma 4.50(ii) and Lemma 4.50(iii), we obtain that for all positions $v \in \mathrm{pos}(t)$ with $v \leq_p w_i$ we have $\tilde{\pi}_2 \circ r(v)[i] = \tilde{\pi}_2 \circ r(w_i)[i] \in \{1,\dots,|Q|\} \times \{\tilde{\pi}_1 \circ r(w_i)\}$, and for all other positions $v \in \mathrm{pos}(t)$ we have $\tilde{\pi}_2 \circ r(v)[i] = 0$.

We define a tuple $\bar{c} \in I$ as follows. If $i \in O$, we let $\bar{c}[i] = \tilde{\pi}_2 \circ r(w_i)[i]$, where w_i is defined as above. If $i \notin O$, we let $\bar{c}[i] = |Q|+1$. Then we have $r \in \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. Thus, $\mathrm{Run}_{\mathcal{B}}(t) = \bigcup_{\bar{c}\in I} \mathrm{Run}_{\mathcal{B}_{\bar{c}}}(t)$. □

We now obtain a finitely sequential representation of $\mathcal{A}$ by applying Theorem 4.29 to the automata $\mathcal{B}_{\bar{c}}$. In particular, we see that the behavior of a trim unambiguous max-plus-WTA is finitely sequential if it does not satisfy the tree fork property. This concludes the proof of Theorem 4.36.

5

Monitor Logics

> My suspicion, and fear, is that the genie is out of the bottle. New and more exotic techniques of surveillance and control are constantly being developed.
>
> Noam Chomsky, *Interview with Stuart Alan Becker*

In this chapter, we develop a logic which is expressively equivalent to *quantitative monitor automata*. Quantitative monitor automata have been introduced very recently by Chatterjee et al. [20] and operate on infinite words. A quantitative monitor automaton is 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 are 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. All of these properties can be described using quantitative monitor automata. Reading an infinite sequence of restocks and demands, we can start a counter at every restock and count 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 [19]. Note that behaviors like these cannot be modeled using *weighted Büchi-automata* [40, 41] or their extension with valuation functions [30]. 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. The main results of this section 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.

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. The biggest challenge for this characterization 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.

An extended abstract of the results of this chapter appeared at the 42nd International Symposium on Mathematical Foundations of Computer Science (MFCS) in 2017 [85].

5.1 Quantitative Monitor Automata

Muller Automata

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 $a_0 a_1 \ldots \in \Sigma^\omega$ 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 $L = \mathcal{L}_\omega(\mathcal{A})$.

Valuation Functions

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 $\textsc{Ces}((z_i)_{i\geq 0}) = \limsup_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} z_i$, 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 $\mathrm{Val}((z_i)_{i\geq 0}) = \mathrm{Val}((z_{i_k})_{k\geq 0})$.

Büchi and Muller Automata with Monitor Counters

We recall quantitative monitor automata as introduced in [20]. 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 $w = a_0 a_1 \ldots \in \Sigma^\omega$ 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 $k \in \{l+1, \ldots, i-1\}$ we have $u^k_j \in \mathbb{Z}$, and (5) infinitely often some counter is activated, i.e.,

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

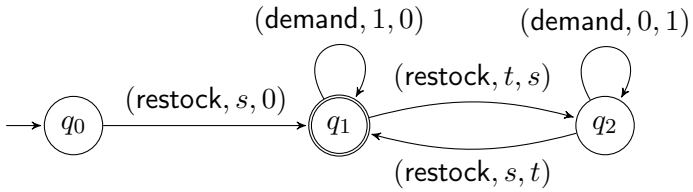
is an infinite set. The set of accepting runs on a word $w \in \Sigma^\omega$ is denoted by $\mathrm{Acc}_{\mathcal{A}}(w)$.

An accepting run r defines a sequence $(z_i)_{i\geq 0}$ from $\mathbb{Z} \cup \{\mathbb{1}\}$ as follows. If $u_j^i = s$ for some $j \in \{1, \ldots, n\}$ and $l > i$ is such that $u_j^l = t$ and for all $k \in \{i+1, \ldots, l-1\}$ we have $u_j^k \in \mathbb{Z}$, then $z_i = \sum_{k=i+1}^{l-1} u_j^k$. If $u_j^i \neq s$ for all $j \in \{1, \ldots, n\}$, 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.

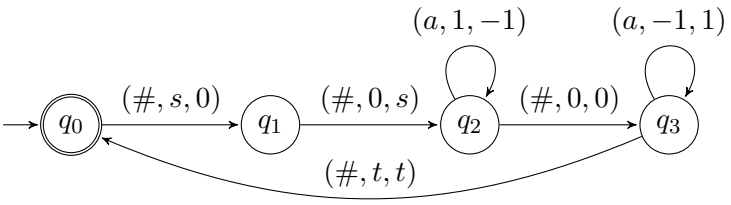
Büchi automata with monitor counters use a Büchi acceptance condition, i.e., at least one accepting state has to appear infinitely often. In Lemma 5.3 we show that using a Muller acceptance condition does not influence the expressive power.

Example 5.1. Consider the alphabet $\Sigma = \{\mathsf{demand}, \mathsf{restock}\}$ with the ω-valuation function Val = 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 5.2 ([20])**.** 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 $\#\#a^{m_1}\#a^{m_2}\#\#\#\ldots$ shall be $\sup_{i\geq 1} |m_{2i-1} - m_{2i}|$. With the choice Val = 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 $(m_1 - m_2, -m_1 + m_2, \mathbb{1}, \ldots, \mathbb{1})$. 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 $|m_1 - m_2|$.

5.2 Closure Properties

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

Lemma 5.3. *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 [32].

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.

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

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 5.4. *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 [35].

(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 5.5. *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 5.6. *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 $[\![\mathcal{A}'']\!](w) = \infty$. □

5.3 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 [30], 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 [20].

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 [27, 32, 30, 45, 73, 26] use quantifiers that act unconditionally on the whole input, i.e., on the whole word, tree, or picture. However, in Lemma 5.11 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.

Let Σ be an alphabet and let $\sigma = (\{P_a \mid a \in \Sigma\} \cup \{\leq\}, \mathrm{ar}_\sigma)$ be the signature of a word over Σ as described in Example 2.1. In the following, we identify every infinite word $w \in \Sigma^\omega$ with its corresponding σ-structure $\mathfrak{w}$, also as described in Example 2.1. Furthermore, as σ is uniquely determined by Σ, we denote the set $\mathrm{MSO}(\sigma)$ simply by $\mathrm{MSO}(\Sigma)$ and say that a formula $\beta \in \mathrm{MSO}(\Sigma)$ is a *Boolean* or an MSO formula over Σ. Let $\mathcal{V}$ be a countable set of first and second order variables and Val an ω-valuation function. We define a three step logic over Σ and Val according to the

following grammars.

$$\psi ::= k \mid \beta \,?\, \psi : \psi$$
$$\zeta_x ::= \mathbb{1} \mid \beta \,?\, \zeta_x : \zeta_x \mid \bigoplus^{x,X} y.\psi$$
$$\varphi ::= \beta \,?\, \varphi : \varphi \mid \min(\varphi, \varphi) \mid \inf x.\varphi \mid \inf X.\varphi \mid \mathrm{Val}\, x.\zeta_x$$

where $\beta \in \mathrm{MSO}(\Sigma)$, $x, y \in \mathcal{V}$ are first order variables, $X \in \mathcal{V}$ is a second order variable, $a \in \Sigma$, and $k \in \mathbb{Z}$. The formulas ψ are called *almost Boolean* formulas, the formulas ζ_x *x-summing* formulas, and the formulas φ *monitor MSO* (mMSO) formulas. We denote the sets of almost Boolean and x-summing formulas over Σ by $\mathrm{mMSO}^{\text{a-bool}}(\Sigma)$ and $\mathrm{mMSO}^{x}(\Sigma)$, respectively, and the set of monitor MSO formulas over Σ and Val by $\mathrm{mMSO}(\Sigma, \mathrm{Val})$.

Note that while almost Boolean formulas as defined here differ in notation from the way we defined almost Boolean formulas in Section 3.4, the semantics of the formulas from $\mathrm{mMSO}^{\text{a-bool}}(\Sigma)$ as we will define them in fact coincide with the semantics of the formulas from $\mathrm{wMSO}^{\text{a-bool}}(\sigma, \mathbb{Z})$ as defined in Section 3.4.

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 5.7 later on.

The notions of free variables and sentences are defined as for MSO formulas, with the addition that the operators inf and Val also bind variables and that in $\bigoplus^{x,X} y.\psi$, the variable y is bound.

For the rest of this chapter, we will always assume that $\mathcal{V}$ denotes a finite set of first and second order variables. Let $w = a_0 a_1 \ldots \in \Sigma^\omega$. We encode $(\mathcal{V}, w)$-assignments ρ 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. In particular, ρ is then always a total mapping. 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 $\beta \in \mathrm{MSO}(\Sigma)$ 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 [16], namely that whenever $\mathrm{Free}(\beta) \subseteq \mathcal{V}$, 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., $L = \mathcal{L}(\beta)$.

Next, we define the semantics of mMSO. 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.

$$[\![k]\!]_{\mathcal{V}}(w,\rho) = k$$

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

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

$$[\![\min(\varphi_1,\varphi_2)]\!]_{\mathcal{V}}(w,\rho) = \min\{[\![\varphi_1]\!]_{\mathcal{V}}(w,\rho), [\![\varphi_2]\!]_{\mathcal{V}}(w,\rho)\}$$

$$[\![\inf x.\varphi]\!]_{\mathcal{V}}(w,\rho) = \inf_{i\in\mathbb{N}} [\![\varphi]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to i])$$

$$[\![\inf X.\varphi]\!]_{\mathcal{V}}(w,\rho) = \inf_{I\subseteq\mathbb{N}} [\![\varphi]\!]_{\mathcal{V}\cup\{X\}}(w,\rho[X\to I])$$

$$[\![\mathrm{Val}\, x.\zeta_x]\!]_{\mathcal{V}}(w,\rho) = \mathrm{Val}((([\![\zeta_x]\!]_{\mathcal{V}\cup\{x\}}(w,\rho[x\to i]))_{i\in\mathbb{N}})$$

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

Remark 5.7. 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 5.12 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 5.8. 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

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

Example 5.9. Consider Example 5.1 again, i.e., the alphabet $\Sigma = \{\mathsf{demand}, \mathsf{restock}\}$ with the ω-valuation function $\mathrm{Val} = \textsc{Ces}$. Then the formula

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

describes the average number of demands between two restocks. We recall that ∞ is simply an abbreviation for the formula $\mathrm{Val}\,x.\mathbb{1}$. As in Example 5.1, if we take Inf or 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 5.10 (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 [27]. We first show that the statement also holds for Boolean, almost Boolean, and x-summing formulas.

Let β be of the form $P_a(x)$, $x\leq y$, 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

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

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 5.4(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 5.4(ii) and Lemma 5.6, $N_{\mathcal{V}} \cap h^{-1}([\![\varphi]\!])$ 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 5.11. *Consider the unrestricted sum quantifier from Remark 5.8*

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

the ω-valuation function Val *defined by*

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

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

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

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

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 5.12. *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

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

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

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

and therefore

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

We obtain

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

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

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

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 5.11, 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 $\mathcal{A}$. □

5.4 Equivalence

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 5.14, 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 5.13. *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 5.14. *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 $X_1, \ldots, X_k$ we define the MSO formulas

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

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 $v\colon \{1, \ldots, k\} \to \delta$. For $i \in \{1, \ldots, k\}$ we write X_i for $X_{v(i)}$ and $\bar{X}$ for $(X_1, \ldots, X_k)$. Furthermore, we fix second order variables $Y_1, \ldots, Y_n$ and write $\bar{Y}$ for $(Y_1, \ldots, Y_n)$. For $j \in \{1, \ldots, n\}$ 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 $\mathrm{muller}(\bar{X})$

which checks that $\bar{X}$ encodes a run of $\mathcal{A}$ satisfying the Muller acceptance condition, and the MSO formula $\mathrm{accept}(\bar{X})$ which checks that $\bar{X}$ encodes an accepting run. The MSO formula $\mathrm{accept}^*(\bar{X}, \bar{Y})$ 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.

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

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

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

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

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

$$\begin{aligned}
\psi^j_{k-1} &= (y \in X_{k-1}\,?\,\mathrm{wt}_j(v(k-1)) : \mathrm{wt}_j(v(k))) \\
\psi^j_i &= \left(y \in X_i\,?\,\mathrm{wt}_j(v(i)) : \psi^j_{i+1}\right) \\
\zeta_{n+1} &= \mathbb{1} \\
\zeta_j &= \left((u_j(x) = s)\,?\,\textstyle\bigoplus^{x,Y_j} y.\psi^j_1 : \zeta_{j+1}\right).
\end{aligned}$$

Then with

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

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 $\text{wt}_j(v(i))$ 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 $\zeta \in \text{mMSO}^x(\Sigma)$, the semantics of $\text{Val}\, x.\zeta$ is Val-MC-recognizable. For the inductive part, we show that if we have mMSO formulas $\varphi_1, \varphi_2 \in \text{mMSO}(\Sigma, \text{Val})$ whose semantics are Val-MC-recognizable and an MSO formula $\beta \in \text{MSO}(\Sigma)$, then the semantics of $\beta \,?\, \varphi_1 : \varphi_2$, $\min(\varphi_1, \varphi_2)$, $\inf x.\varphi_1$, 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 $\beta \,?\, \varphi_1 : \varphi_2$.

Lemma 5.15. *Let $\beta \in \text{MSO}(\Sigma)$ be an MSO formula and $\varphi_1, \varphi_2 \in \text{mMSO}(\Sigma, \text{Val})$ such that $[\![\varphi_1]\!]$ and $[\![\varphi_2]\!]$ are* Val*-MC-recognizable. Then with $\varphi = \beta \,?\, \varphi_1 : \varphi_2$, the series $[\![\varphi]\!]$ is also* Val*-MC-recognizable.*

Proof. Let $\mathcal{V} = \text{Free}(\varphi)$. Then we have $\text{Free}(\varphi_1) \subseteq \mathcal{V}$ and $\text{Free}(\varphi_2) \subseteq \mathcal{V}$ and hence by Lemma 5.10 $[\![\varphi_1]\!]_\mathcal{V}$ and $[\![\varphi_2]\!]_\mathcal{V}$ are Val-MC-recognizable. Due to $\text{Free}(\beta) \subseteq \mathcal{V}$, 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 5.5 and Lemma 5.6, $[\![\varphi]\!] = \min(\mathcal{L}_\mathcal{V}(\beta) \cap [\![\varphi_1]\!]_\mathcal{V}, \mathcal{L}_\mathcal{V}(\neg\beta) \cap [\![\varphi_2]\!]_\mathcal{V})$ 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 5.16. *Let $\varphi_1, \varphi_2 \in \text{mMSO}(\Sigma, \text{Val})$ 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 $\mathcal{V} = \text{Free}(\varphi_1) \cup \text{Free}(\varphi_2)$, then by Lemma 5.10, $[\![\varphi_1]\!]_\mathcal{V}$ and $[\![\varphi_2]\!]_\mathcal{V}$ are Val-MC-recognizable. Hence by Lemma 5.5, $[\![\varphi]\!] = \min([\![\varphi_1]\!]_\mathcal{V}, [\![\varphi_2]\!]_\mathcal{V})$ 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 5.17. *Let $\varphi \in \text{mMSO}(\Sigma, \text{Val})$ 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 5.10 shows that $[\![\varphi]\!]_{\mathcal{V}\cup\{x\}}$ is Val-MC-recognizable and therefore by Lemma 5.4(i), the series $[\![\inf x.\varphi]\!]_{\mathcal{V}} = h([\![\varphi]\!]_{\mathcal{V}\cup\{x\}})$ is Val-MC-recognizable as well. □

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

Lemma 5.18. *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 $\beta_1,\dots,\beta_n \in \mathrm{MSO}(\Sigma)$ and weights $z_1,\dots,z_n \in \mathbb{Z}$ 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 $n = 1$, $\beta_1 = \mathtt{true}$, and $z_1 = k$.

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

Lemma 5.19. *Let $\zeta \in \mathrm{mMSO}^x(\Sigma)$ be an x-summing formula and $\mathcal{V} \supseteq \mathrm{Free}(\zeta)$. Then there exist MSO formulas $\beta_1,\dots,\beta_n \in \mathrm{MSO}(\Sigma)$ and formulas $\zeta_1,\dots,\zeta_n$ 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 5.18. For $\zeta = \mathbb{1}$ we choose $n = 0$, i.e., there are no formulas β. For $\zeta = \bigoplus^{x,X} y.\psi$, we choose $\beta_1 = \mathtt{true}$ and $\zeta_1 = \zeta$.

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

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

Theorem 5.20. *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 [27, 32]. Let $\beta_1, \ldots, \beta_n$ and $\zeta_1, \ldots, \zeta_n$ be the formulas we can find for ζ according to Lemma 5.19. 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 5.18.

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 $\Sigma_{\mathcal{V}}^\omega$. Consider a valid $(w, \rho) \in \Sigma_{\mathcal{V}}^\omega$. 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 $[\![\mathrm{Val}\, x.\zeta]\!]$.

In the following, we present the formal proof. 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 5.19 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 $(w, \rho) \in \Sigma^{\omega}_{\mathcal{W}}$ 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 $(w, \rho, v) \in \tilde{\Sigma}^{\omega}_{\mathcal{W}}$ 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 $(w, \rho, v) \in \tilde{\Sigma}^{\omega}_{\mathcal{V}}$. 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 5.18 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 $(w, \rho, v) \in \tilde{\Sigma}^{\omega}_{\mathcal{W}'}$ 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 $(w, \rho, v) \in \tilde{\Sigma}^{\omega}_{\mathcal{V}}$. 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 $\{Y_1, \ldots, Y_n\} \subseteq \mathcal{V}$. For $i \in \{1, \ldots, n\}$, $\star \in \{s, t, 1, \ldots, n_i\}$, and $i' \in \{i, i+n\}$, we define the abbreviations

$$Y_i(z) = \bigvee_{\substack{(a,g,h) \in \tilde{\Sigma}_{\mathcal{V}} \\ g(Y_i) = 1}} P_{(a,g,h)}(z)$$

$$(h_{i'}(z) = \star) = \bigvee_{\substack{(a,g,h) \in \tilde{\Sigma}_{\mathcal{V}} \\ h(i') = \star}} P_{(a,g,h)}(z)$$

$$\begin{aligned}\text{even}(x,X) = \exists Y.\exists Z.\forall x'.((x' \in X \wedge x' < x) \rightarrow (x' \in Y \leftrightarrow \neg x' \in Z)) \wedge \\ \forall x'.((x' \in Y \vee x' \in Z) \rightarrow (x' \in X \wedge x' < x)) \wedge \\ \forall z.(z \in Z \rightarrow \exists y.(y \in Y \wedge y < z)) \wedge \\ \forall y.(y \in Y \rightarrow \exists z.(z \in Z \wedge y < z)) \wedge \\ \forall y_1.\forall y_2.((y_1 \in Y \wedge y_2 \in Y \wedge y_1 < y_2) \\ \qquad \rightarrow \exists z.(z \in Z \wedge y_1 < z \wedge z < y_2)) \wedge \\ \forall z_1.\forall z_2.((z_1 \in Z \wedge z_2 \in Z \wedge z_1 < z_2) \\ \qquad \rightarrow \exists y.(y \in Y \wedge z_1 < y \wedge y < z_2)).\end{aligned}$$

For every $w \in \Sigma^\omega$, we have $(w, [x \rightarrow k, X \rightarrow I]) \models \text{even}(x, X)$ 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 5.8. Now for $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n_i\}$, and $i' \in \{i, i+n\}$, we define

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

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 5.21. *Let $(w, \rho) \in \Sigma_{\mathcal{V}}^\omega$ be valid, then there exists exactly one mapping $v\colon \mathbb{N} \rightarrow H$ 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 $i' \in \{1, \ldots, 2n\}$ and $[\![\zeta]\!]_{\mathcal{V}\cup\{x\}}(w, \rho[x \rightarrow k]) = \mathbb{1}$, or we have $v(k)(i') = s$ for exactly one $i' \in \{1, \ldots, 2n\}$ and with $l = \min\{\iota > k \mid v(\iota)(i') = t\}$ we have*

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

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 $i' \leq n$, (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) $v_1(k)(i') = \bot$. 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

$$\begin{aligned} &((w,\rho,v_1),[x \to k]) \models \\ &\beta_i'''(x) \wedge \exists X.\Big(\forall z.(z \in X \leftrightarrow Y_i(z)) \wedge \mathrm{notLast}(x,X) \wedge \mathrm{even}(x,X)\Big). \end{aligned} \tag{5.4.1}$$

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 (5.4.1) does not depend on v_1. Hence, the formula in (5.4.1) 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

$$((w,\rho,v),[x \to k]) \models \beta_i'''(x) \wedge \exists X.\Big(\forall z.(z \in X \leftrightarrow Y_i(z)) \wedge \mathrm{notLast}(x,X)\Big), \tag{5.4.2}$$

we redefine $v(k)(i') = s$. Note that the validity of (5.4.2) 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

$$\begin{aligned} &((w,\rho,v),[x \to k]) \models \\ &Y_i(x) \wedge \exists z.\Big((z < x) \wedge (h_{i'}(z) = s) \wedge \forall z'.\Big((z' < x \wedge Y_i(z')) \to z' \leq z\Big)\Big). \end{aligned} \tag{5.4.3}$$

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 (5.4.3) 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 5.8, 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

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

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

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

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

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

if and only if $(w, \rho[x \to k]) \models \beta_i \wedge \mathrm{notLast}(x, Y_i)$. 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 $i \in \{1, \dots, n\}$ 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

$$\begin{aligned}
&((w, \rho, v), [x \to l]) \models \\
&Y_i(x) \wedge \exists z.\Big((z < x) \wedge (h_{i'}(z) = s) \wedge \forall z'.\Big((z' < x \wedge Y_i(z')) \to z' \leq z\Big)\Big).
\end{aligned} \tag{5.4.4}$$

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 (5.4.4) 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

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

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

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

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 5.21, 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 5.21 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 $\{k \in \mathbb{N} \mid v(k)(i') = s \text{ for some } i' \in \{1, \ldots, 2n\}\}$ is infinite. Since $\mathrm{Acc}_{\mathcal{A}}(w, \rho, v) = \emptyset$, the run r is not accepting, so we see by Lemma 5.21 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 $(z_j)_{j \geq 0}$ 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 5.21 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 5.21 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 $(w, \rho) \in \Sigma^{\omega}_{\mathcal{V}}$, we know by Lemma 5.21 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 5.4, $h([\![\mathcal{A}]\!])$ is Val-MC-recognizable. □

By combining Theorem 5.20 and Lemmata 5.15, 5.16, and 5.17, we obtain that the semantics of every mMSO formula $\varphi \in \mathrm{mMSO}(\Sigma, \mathrm{Val})$ is Val-MC-recognizable. Together with Lemma 5.14, this concludes the proof of Theorem 5.13.

Bibliography

> When you steal from one author, it's plagiarism;
> if you steal from many, it's research.
>
> Wilson Mizner

[1] Athanasios Alexandrakis and Symeon Bozapalidis. Weighted grammars and Kleene's theorem. *Information Processing Letters*, 24(1):1–4, 1987.

[2] Cyril Allauzen and Mehryar Mohri. Efficient algorithms for testing the twins property. *Journal of Automata, Languages and Combinatorics*, 8(2):117–144, 2003.

[3] Rajeev Alur and Parthasarathy Madhusudan. Adding nesting structure to words. *Journal of the ACM*, 56(3):16:1–16:43, 2009.

[4] Sebastian Bala. Which finitely ambiguous automata recognize finitely sequential functions? In Krishnendu Chatterjee and Jiří Sgall, editors, *38th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 8087 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2013.

[5] Sebastian Bala and Artur Koniński. Unambiguous automata denoting finitely sequential functions. In Adrian-Horia Dediu, Carlos Martín-Vide, and Bianca Truthe, editors, *7th International Conference on Language and Automata Theory and Applications (LATA)*, volume 7810 of *Lecture Notes in Computer Science*, pages 104–115. Springer, 2013.

[6] Marie-Pierre Béal, Olivier Carton, Christophe Prieur, and Jacques Sakarovitch. Squaring transducers: an efficient procedure for deciding functionality and sequentiality. *Theoretical Computer Science*, 292(1):45–63, 2003.

[7] Jean Berstel and Christophe Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18:115–148, 1982.

[8] Jean Berstel and Christophe Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1988.

[9] Garrett Birkhoff. *Lattice Theory*. Colloquium publications. American Mathematical Society, 1948. 4th printing.

[10] Johanna Björklund, Frank Drewes, and Niklas Zechner. An efficient best-trees algorithm for weighted tree automata over the tropical semiring. In Adrian-Horia Dediu, Enrico Formenti, Carlos Martín-Vide, and Bianca Truthe, editors, *9th International Conference on Language and Automata Theory and Applications (LATA)*, volume 8977 of *Lecture Notes in Computer Science*, pages 97–108. Springer, 2015.

[11] Meera Blattner and Tom Head. Automata that recognize intersections of free submonoids. *Information and Control*, 35(3):173–176, 1977.

[12] Manuel Blum and Carl Hewitt. Automata on a 2-dimensional tape. In *8th Annual Symposium on Switching and Automata Theory (SWAT)*, pages 155–160. IEEE Computer Society, 1967.

[13] Alexander Bockmayr, Volker Weispfenning, and Michael Maher. Solving numerical constraints. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume 1, chapter 12, pages 751 – 842. Elsevier and MIT Press, 2001.

[14] Benedikt Bollig, Paul Gastin, and Benjamin Monmege. Weighted specifications over nested words. In Frank Pfenning, editor, *16th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 7794 of *Lecture Notes in Computer Science*, pages 385–400. Springer, 2013.

[15] Björn Borchardt. A pumping lemma and decidability problems for recognizable tree series. *Acta Cybernetica*, 16(4):509–544, 2004.

[16] Julius Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.

[17] Matthias Büchse and Anja Fischer. Deciding the twins property for weighted tree automata over extremal semifields. In Frank Drewes and Marco Kuhlmann, editors, *2nd Workshop on Applications of Tree Automata Techniques in Natural Language Processing (ATANLP)*, pages 11–20. The Association for Computer Linguistics, 2012.

[18] Matthias Büchse, Jonathan May, and Heiko Vogler. Determinization of weighted tree automata using factorizations. *Journal of Automata, Languages and Combinatorics*, 15(3/4):229–254, 2010.

[19] Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Transactions on Computational Logic*, 11(4):23:1–23:38, 2010.

[20] Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Quantitative monitor automata. In Xavier Rival, editor, *23rd Static Analysis Symposium (SAS)*, volume 9837 of *Lecture Notes in Computer Science*, pages 23–38. Springer, 2016.

[21] Krishnendu Chatterjee, Thomas A. Henzinger, and Jan Otop. Nested weighted automata. *ACM Transactions on Computational Logic*, 18(4):31:1–31:44, 2017.

[22] Noam Chomsky. Three models for the description of language. *IRE Transactions on Information Theory*, 2(3):113–124, 1956.

[23] Bruno Courcelle. Monadic second-order definable graph transductions: A survey. *Theoretical Computer Science*, 126(1):53–75, 1994.

[24] Laure Daviaud, Pierre Guillon, and Glenn Merlet. Comparison of max-plus automata and joint spectral radius of tropical matrices. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

[25] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4(5):406–451, 1970.

[26] Manfred Droste and Stefan Dück. Weighted automata and logics on graphs. In Giuseppe F. Italiano, Giovanni Pighizzini, and Donald Sannella, editors, *40th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 9234 of *Lecture Notes in Computer Science*, pages 192–204. Springer, 2015.

[27] Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2):69–86, 2007.

[28] Manfred Droste and Doreen Heusel. The supports of weighted unranked tree automata. *Fundamenta Informaticae*, 136(1-2):37–58, 2015.

[29] Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.

[30] Manfred Droste and Ingmar Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Information and Computation*, 220:44–59, 2012.

[31] Manfred Droste and Erik Paul. A Feferman-Vaught decomposition theorem for weighted MSO logic. In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 76:1–76:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[32] Manfred Droste and George Rahonis. Weighted automata and weighted logics on infinite words. In Oscar H. Ibarra and Zhe Dang, editors, *10th International Conference on Developments in Language Theory (DLT)*, volume 4036 of *Lecture Notes in Computer Science*, pages 49–58. Springer, 2006.

[33] Manfred Droste and George Rahonis. Weighted automata and weighted logics with discounting. *Theoretical Computer Science*, 410(37):3481–3494, 2009.

[34] Manfred Droste and Heiko Vogler. Weighted tree automata and weighted logics. *Theoretical Computer Science*, 366(3):228–247, 2006.

[35] Manfred Droste and Heiko Vogler. Weighted automata and multi-valued logics over arbitrary bounded lattices. *Theoretical Computer Science*, 418:14–36, 2012.

[36] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49(2):129–141, 1961.

[37] Samuel Eilenberg. *Automata, Languages, and Machines*, volume A of *Pure and Applied Mathematics*. Academic Press, 1974.

[38] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961.

[39] Zoltán Ésik and Werner Kuich. Formal tree series. *Journal of Automata, Languages and Combinatorics*, 8(2):219–285, 2003.

[40] Zoltán Ésik and Werner Kuich. A semiring-semimodule generalization of ω-regular languages I. *Journal of Automata, Languages and Combinatorics*, 10(2/3):203–242, 2005.

[41] Zoltán Ésik and Werner Kuich. A semiring-semimodule generalization of ω-regular languages II. *Journal of Automata, Languages and Combinatorics*, 10(2/3):243–264, 2005.

[42] Zoltán Ésik and Werner Kuich. On iteration semiring-semimodule pairs. *Semigroup Forum*, 75(1):129–159, 2007.

[43] Javier Esparza, Pierre Ganty, Stefan Kiefer, and Michael Luttenberger. Parikh's theorem: A simple and direct automaton construction. *Information Processing Letters*, 111(12):614–619, 2011.

[44] Solomon Feferman and Robert L. Vaught. The first order properties of products of algebraic systems. *Fundamenta Mathematicae*, 47(1):57–103, 1959.

[45] Ina Fichtner. Weighted picture automata and weighted logics. *Theory of Computing Systems*, 48(1):48–78, 2011.

[46] Emmanuel Filiot, Ismaël Jecker, Nathan Lhote, Guillermo A. Pérez, and Jean-François Raskin. On delay and regret determinization of max-plus automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE Computer Society, 2017.

[47] Zoltán Fülöp and Heiko Vogler. Weighted tree automata and tree transducers. In Droste et al. [29], chapter 9, pages 313–403.

[48] Ferenc Gécseg and Magnus Steinby. Tree languages. In Rozenberg and Salomaa [94], chapter 1, pages 1–68.

[49] Ferenc Gécseg and Magnus Steinby. Tree automata. *CoRR*, abs/1509.06233, 2015. Available at `http://arxiv.org/abs/1509.06233`.

[50] Siegfried Gottwald. *A Treatise on Many-Valued Logics*, volume 9 of *Studies in Logic and Computation*. Research Studies Press, 2001.

[51] Yuri Gurevich. Modest theory of short chains. I. *The Journal of Symbolic Logic*, 44(4):481–490, 1979.

[52] Yuri Gurevich. Chapter XIII: Monadic second-order theories. In Jon Barwise and Solomon Feferman, editors, *Model-Theoretic Logics*, volume 8 of *Perspectives in Mathematical Logic*, pages 479–506. Springer, 1985.

[53] Petr Hájek. *Metamathematics of Fuzzy Logic*, volume 4 of *Trends in Logic*. Kluwer Academic Publishers, 1998.

[54] Kōsaburō Hashiguchi. Algorithms for determining relative star height and star height. *Information and Computation*, 78(2):124–169, 1988.

[55] Kōsaburō Hashiguchi, Kenichi Ishiguro, and Shūji Jimbo. Decidability of the equivalence problem for finitely ambiguous finance automata. *International Journal of Algebra and Computation*, 12(3):445–461, 2002.

[56] Hendrik Jan Hoogeboom and Paulien ten Pas. Monadic second-order definable text languages. *Theory of Computing Systems*, 30(4):335–354, 1997.

[57] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 2000.

[58] Daniel Kirsten. A Burnside approach to the termination of Mohri's algorithm for polynomially ambiguous min-plus-automata. *Informatique Théorique et Applications*, 42(3):553–581, 2008.

[59] Daniel Kirsten. The support of a recognizable series over a zero-sum free, commutative semiring is recognizable. *Acta Cybernetica*, 20(2):211–221, 2011.

[60] Daniel Kirsten. Decidability, undecidability, and PSPACE-completeness of the twins property in the tropical semiring. *Theoretical Computer Science*, 420:56–63, 2012.

[61] Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In Susanne Albers and Jean-Yves Marion, editors, *26th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 589–600. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2009.

[62] Stephen Cole Kleene. Representation of events in nerve nets and finite automata. In Claude Shannon and John McCarthy, editors, *Automata Studies*, pages 3–41. Princeton University Press, 1956.

[63] Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004.

[64] Jan Komenda, Sébastien Lahaye, Jean-Louis Boimond, and Ton van den Boom. Max-plus algebra in the history of discrete event systems. *Annual Reviews in Control*, 45:240–249, 2018.

[65] Adam Koprowski and Johannes Waldmann. Max/plus tree automata for termination of term rewriting. *Acta Cybernetica*, 19(2):357–392, 2009.

[66] Daniel Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4(3):405–426, 1994.

[67] Werner Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 9, pages 609–677. Springer, 1997.

[68] Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs in Theoretical Computer Science*. Springer, 1986.

[69] Hans Läuchli and John Leonard. On the elementary theory of linear order. *Fundamenta Mathematicae*, 59(1):109–116, 1966.

[70] Johann A. Makowsky. Algorithmic uses of the Feferman-Vaught theorem. *Annals of Pure and Applied Logic*, 126(1):159–213, 2004.

[71] Christian Mathissen. *Weighted Automata and Weighted Logics over Tree-like Structures*. PhD thesis, Leipzig University, Germany, 2009.

[72] Christian Mathissen. Definable transductions and weighted logics for texts. *Theoretical Computer Science*, 411(3):631–659, 2010.

[73] Christian Mathissen. Weighted logics for nested words and algebraic formal power series. *Logical Methods in Computer Science*, 6(1):1–34, 2010.

[74] Filip Mazowiecki and Cristian Riveros. Pumping lemmas for weighted automata. In Rolf Niedermeier and Brigitte Vallée, editors, *35th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[75] Mehryar Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

[76] Mehryar Mohri. Weighted automata algorithms. In Droste et al. [29], chapter 6, pages 213–254.

[77] Andrzej Mostowski. On direct products of theories. *The Journal of Symbolic Logic*, 17(1):1–31, 1952.

[78] David Eugene Muller. Infinite sequences and finite machines. In *4th Annual Symposium on Switching Circuit Theory and Logical Design (SWCT)*, pages 3–16. IEEE Computer Society, 1963.

[79] Jan Mycielski, Pavel Pudlák, and Alan S. Stern. *A Lattice of Chapters of Mathematics: Interpretations between Theorems*. Number 426 in Memoirs of the American Mathematical Society. American Mathematical Society, 1990.

[80] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[81] Rohit Jivanlal Parikh. On context-free languages. *Journal of the ACM*, 13(4):570–581, 1966.

[82] Erik Paul. Weighted tree automata and quantitative logics with a focus on ambiguity. Diplomarbeit, Leipzig University, 2015. Available at `https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-164548`.

[83] Erik Paul. On finite and polynomial ambiguity of weighted tree automata. In Srečko Brlek and Christophe Reutenauer, editors, *20th International Conference on Developments in Language Theory (DLT)*, volume 9840 of *Lecture Notes in Computer Science*, pages 368–379. Springer, 2016.

[84] Erik Paul. The equivalence, unambiguity and sequentiality problems of finitely ambiguous max-plus tree automata are decidable. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

[85] Erik Paul. Monitor logics for quantitative monitor automata. In Kim G. Larsen, Hans L. Bodlaender, and Jean-François Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 14:1–14:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

[86] Erik Paul. Finite sequentiality of unambiguous max-plus tree automata. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 126 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

[87] Slav Petrov. Latent variable grammars for natural language parsing. In *Coarse-to-Fine Natural Language Processing*, Theory and Applications of Natural Language Processing, chapter 2, pages 7–46. Springer, 2012.

[88] Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3(2):114–125, 1959.

[89] Michael Oser Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[90] Frank P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, series 2, 30:264–286, 1930.

[91] Elena V. Ravve, Zeev Volkovich, and Gerhard-Wilhelm Weber. Effective optimization with weighted automata on decomposable trees. *Optimization*, 63(1):109–127, 2014.

[92] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Volume 1: Word, Language, Grammar*. Springer, 1997.

[93] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Volume 2: Linear Modeling: Background and Application*. Springer, 1997.

[94] Grzegorz Rozenberg and Arto Salomaa, editors. *Handbook of Formal Languages, Volume 3: Beyond Words*. Springer, 1997.

[95] Jacques Sakarovitch. A construction on finite automata that has remained hidden. *Theoretical Computer Science*, 204(1-2):205–231, 1998.

[96] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.

[97] Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Texts and Monographs in Computer Science. Springer, 1978.

[98] Marcel-Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, 1961.

[99] Marcel-Paul Schützenberger. Sur les relations rationnelles entre monoïdes libres. *Theoretical Computer Science*, 3(2):243–259, 1976.

[100] Helmut Seidl. On the finite degree of ambiguity of finite tree automata. *Acta Informatica*, 26(6):527–542, 1989.

[101] Saharon Shelah. The monadic theory of order. *Annals of Mathematics*, 102(3):379–419, 1975.

[102] Imre Simon. Limited subsets of a free monoid. In *19th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 143–150. IEEE Computer Society, 1978.

[103] Imre Simon. Recognizable sets with multiplicities in the tropical semiring. In Michal P. Chytil, Ladislav Janiga, and Václav Koubek, editors, *13th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 324 of *Lecture Notes in Computer Science*, pages 107–120. Springer, 1988.

[104] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.

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

[106] Johannes Waldmann. Weighted automata for proving termination of string rewriting. *Journal of Automata, Languages and Combinatorics*, 12(4):545–570, 2007.

[107] Andreas Weber. Finite-valued distance automata. *Theoretical Computer Science*, 134(1):225–251, 1994.

[108] Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, 1991.

List of Symbols

[...] and 'X' never, ever, marks the spot.

INDIANA JONES

$\lvert t\rvert$	Number of positions of tree t
$\lvert w\rvert$	Length of word w
$\lvert w\rvert_a$	Number of occurrences of letter a in word w
$\lvert X\rvert$	Cardinality of set X
$\leq_l$	Lexicographic ordering on $\mathbb{N}^*$
$\leq_p$	Prefix ordering on $\mathbb{N}^*$
$\preceq$	Reachability relation on states of an automaton
$\approx$	Mutual reachability relation on states of an automaton
$[p]$	Equivalence class of state p with respect to mutual reachability of states
$\Diamond_i(t)$	Position of lexicographically ordered i-th leaf of context t labeled $\diamond$
$\Rightarrow_G$	Derivation step of grammar G
$\bar{\beta}, \bar{p}, \bar{X}$	Tuples of elements
$[\![\mathcal{A}]\!]$	Behavior of automaton $\mathcal{A}$
$\mathfrak{A}_{\mathcal{V}}$	Set of all $(\mathcal{V}, \mathfrak{A})$-assignments
$\mathfrak{A} \sqcup \mathfrak{B}$	disjoint union of structures $\mathfrak{A}$ and $\mathfrak{B}$
$\mathfrak{A} \times \mathfrak{B}$	product of structures $\mathfrak{A}$ and $\mathfrak{B}$
$\mathrm{Acc}_{\mathcal{A}}(t)$	Set of all accepting runs of automaton $\mathcal{A}$ on tree t
$\mathrm{ar}_\sigma(R)$	Arity of relation symbol R of signature σ
$\mathbb{B}$	Boolean semiring
CES	Cesàro mean
$\mathrm{CON}^{\mathbb{1}}, \mathrm{CON}^{2}$	See page 18

$\Delta_{\mathcal{A}}$	Set of all transitions of automaton $\mathcal{A}$
$\mathrm{dom}(f)$	Domain of mapping f
ε	Empty word
$\langle\langle E \rangle\rangle$	Semantics of expression E
$\mathrm{Exp}_n(K)$	Set of all expressions over semiring K with variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$
$f\colon X \to Y$	Mapping from set X to set Y
$f\colon X \nrightarrow Y$	Partial mapping from set X to set Y
$f(X)$	Image of set X under mapping f
$f^{-1}(Y)$	Preimage of set Y under mapping f
$f\restriction_X$	Restriction of mapping f to set X
$f \circ g$	Composition of mappings f and g
$\mathrm{FO}(\sigma)$	Set of all first order formulas over signature σ
$\mathrm{Free}(\beta)$	Set of free variables of formula β
$\Gamma^{(m)}$	Set of all letters of rank m of ranked alphabet Γ
$\Gamma_\diamond$	Ranked alphabet Γ extended by a symbol $\diamond$ of rank 0
$\mathrm{height}(t)$	Height of tree t
$\mathrm{In}^Q(r)$	Set of states of Q appearing infinitely often in run r
$\mathcal{I}_{\mathfrak{A}}$	Interpretation of structure $\mathfrak{A}$
$\mathrm{label}_t(w)$	Label of position w of tree t
$L \cap S$	Intersection of language L and series S
$\mathcal{L}(\mathcal{A})$	Language accepted by automaton $\mathcal{A}$
$\mathcal{L}_\omega(\mathcal{A})$	Infinitary language accepted by automaton $\mathcal{A}$
$\mathcal{L}(\beta)$	Language defined by formula β
$\mathcal{L}(G)$	Language generated by grammar G
$\mathrm{MSO}(\sigma)$	Set of all monadic second order formulas over signature σ
$\mathrm{mMSO}^{\text{a-bool}}(\Sigma)$	Set of all almost Boolean formulas over alphabet Σ
$\mathrm{mMSO}^x(\Sigma)$	Set of all x-summing formulas over alphabet Σ
$\mathrm{mMSO}(\Sigma, \mathrm{Val})$	Set of all monitor MSO formulas over alphabet Σ and ω-valuation function Val
$\mathbb{N}$	Natural numbers including 0
$\mathbb{N}_+$	Natural numbers excluding 0
$\mathbb{p}(w)$	Parikh vector of word w
$\mathbb{p}(L)$	Parikh image of language L

$p \xrightarrow{s\|x} q$	Indication that on Γ-word s there exists a valid run from p to q with weight x
$\mathrm{pos}(t)$	Set of positions of tree t
$\mathcal{P}(X)$	Power set of set X
$\mathrm{PRD}^{\mathbb{1}}, \mathrm{PRD}^{\mathbb{2}}$	See page 18
$\mathbb{Q}$	Rational numbers
$\mathbb{Q}_{\max}$	Max-plus semiring of rational numbers
$r\restriction_w$	Restriction of run r to subtree at position w
$r\langle r' \to w\rangle$	Substitution of run r' on subtree at position w into run r
$r^{n\langle w\rangle}$	The n-th power of run r by substitution into position w
$\rho[x \to a]$	Update of variable assignment ρ to map variable x to a
$\rho\restriction_{\mathfrak{A}}$	Restriction of variable assignment ρ to structure $\mathfrak{A}$
$\rho \cup \varsigma$	Union of variable assignments ρ and ς
$\mathbb{R}$	Real numbers
$\mathbb{R}_{\max}$	Max-plus semiring of real numbers
$\mathbb{R}_{\min}$	Min-plus semiring of real numbers
$R^{\mathfrak{A}}$	Interpretation of relation symbol R in structure $\mathfrak{A}$
$\mathrm{range}(f)$	Range of mapping f
Rel_σ	Relation symbols of signature σ
$\mathrm{rk}_\Gamma(a)$	Rank of letter a from ranked alphabet Γ
$\mathrm{Run}_{\mathcal{A}}(t)$	Set of all valid runs of automaton $\mathcal{A}$ on tree t
$\mathrm{Run}_{\mathcal{A}}(t, q)$	Set of all valid runs of automaton $\mathcal{A}$ on tree t with state q at the root
$\mathrm{Run}^{\diamond}_{\mathcal{A}}(t)$	Set of all valid runs of automaton $\mathcal{A}$ on context t
$\mathrm{Run}^{\diamond}_{\mathcal{A}}(q_1, \ldots, q_n, t, q_0)$	Set of all valid runs of automaton $\mathcal{A}$ on context t from states $q_1, \ldots, q_n$ to state q_0
s^n	The n-th power of Γ-word s
Σ^*	Set of finite words over alphabet Σ
Σ^ω	Set of infinite words over alphabet Σ
$\Sigma_{\mathcal{V}}$	Abbreviation for $\Sigma \times \{0,1\}^{\mathcal{V}}$ for alphabet Σ and set of variables $\mathcal{V}$
$\mathrm{Str}(\sigma)$	Class of all σ-structures
$\mathrm{supp}(\mathcal{A})$	Support of automaton $\mathcal{A}$
$t(w)$	Label of position w of tree t
$t\restriction_w$	Subtree of tree t at position w
$t\langle s \to w\rangle$	Substitution of tree s into position w of tree t

$\mathbb{t}(t, r, w)$	Transition of run r at position w of tree t
T_Γ	Set of all trees over ranked alphabet Γ
$T_{\Gamma_\diamond}$	Set of all Γ-contexts for ranked alphabet Γ
$\mathcal{U}_\mathfrak{A}$	Universe of structure $\mathfrak{A}$
uv	Concatenation of words u and v
$[\![\varphi]\!]$	Semantics of formula φ
φ^{-x}	Formula obtained from φ by replacing all atomic subformulas containing variable x by `false`
$\Phi^\star(\mathfrak{A}, \varsigma)$	Φ-induced structure of structure $\mathfrak{A}$ and variable assignment ς
$\mathrm{wFO}(\sigma, K)$	Set of all weighted first order formulas over signature σ and semiring K
$\mathrm{wFO}^\neg(\sigma, L)$	Set of all wFO formulas over signature σ and De Morgan algebra L including involution $\neg$
$\mathrm{wFO}^{\bigotimes\text{-free}}(\sigma, K)$	Set of all product-free wFO formulas over signature σ and semiring K
$\mathrm{wMSO}(\sigma, K)$	Set of all weighted monadic second order formulas over signature σ and semiring K
$\mathrm{wMSO}^\neg(\sigma, L)$	Set of all wMSO formulas over signature σ and De Morgan algebra L including involution $\neg$
$\mathrm{wMSO}^{\bigotimes\text{-res}}(\sigma, K)$	Set of all product-restricted wMSO formulas over signature σ and semiring K
$\mathrm{wMSO}^{\text{a-bool}}(\sigma, K)$	Set of all almost-Boolean wMSO formulas over signature σ and semiring K
$\mathrm{wt}_\mathcal{A}(t, r)$	Weight of run r on tree t with respect to automaton $\mathcal{A}$
$\mathrm{wt}^\diamond_\mathcal{A}(t, r)$	Weight of run r on context t with respect to automaton $\mathcal{A}$
Y^X	Set of mappings from set X to set Y
$\mathbb{Z}$	Integers

Index

Wissenschaftlicher Werdegang

> It always takes longer than you expect,
> even when you take into account Hofstadter's Law.
>
> *Hofstadter's Law*

seit 10/2018	Wissenschaftlicher Mitarbeiter der Abteilung Automaten und Sprachen am Institut für Informatik der Universität Leipzig und assoziierter Promotionsstudent des DFG Graduiertenkollegs *Quantitative Logics and Automata* (QuantLA)
04/2017 – 09/2018	Wissenschaftlicher Mitarbeiter im DFG Graduiertenkolleg *Quantitative Logics and Automata* (QuantLA)
10/2015 – 03/2017	Promotionsstipendium im DFG Graduiertenkolleg *Quantitative Logics and Automata* (QuantLA)
seit 10/2015	Promotionsstudium der Informatik an der Universität Leipzig
10/2009 – 08/2015	Studium der Mathematik an der Universität Leipzig Abschluss: Diplom-Mathematiker Titel der Abschlussarbeit: *Weighted Tree Automata and Quantitative Logics with a Focus on Ambiguity* Betreuer: Prof. Dr. Manfred Droste Abschlussnote: mit Auszeichnung (1,0)
08/2000 – 06/2008	Geschwister-Scholl-Gymnasium, Taucha Abschluss: Abitur (Abschlussnote 1,4)

Dissertationsbezogene bibliographische Daten

Begutachtete Veröffentlichungen mit direktem Bezug zur Dissertation

- Erik Paul. Finite sequentiality of unambiguous max-plus tree automata. In *36th International Symposium on Theoretical Aspects of Computer Science (STACS)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 126, pp. 55:1–55:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.

- Manfred Droste, Erik Paul. A Feferman-Vaught decomposition theorem for weighted MSO logic. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 117, pp. 76:1–76:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

- Erik Paul. The equivalence, unambiguity and sequentiality problems of finitely ambiguous max-plus tree automata are decidable. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 83, pp. 53:1–53:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

- Erik Paul. Monitor logics for quantitative monitor automata. In *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS)*, Leibniz International Proceedings in Informatics (LIPIcs), vol. 83, pp. 14:1–14:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.

Weitere begutachtete Veröffentlichungen

- Erik Paul. On finite and polynomial ambiguity of weighted tree automata. In *20th International Conference on Developments in Language Theory (DLT)*, Lecture Notes in Computer Science (LNCS), vol. 9840, pp. 368–379. Springer, 2016.

Eingereichte Arbeiten

- Yun Shang, Xiaoya Cheng, Manfred Droste, Erik Paul. Nivat's Theorem for Turing Machines Based on Unsharp Quantum Logic. Eingereicht bei *Theoretical Computer Science*.

- Erik Paul. Finite sequentiality of unambiguous max-plus tree automata. Eingereicht bei *Theory of Computing Systems, Special Issue of STACS 2019*.

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien oder erbrachten Dienstleistungen als solche gekennzeichnet.

Leipzig, den 08. Januar 2020

. . . . . .
(Erik Paul)