

# Aperiodic Weighted Automata and Weighted First-Order Logic

Manfred Droste

Institut für Informatik, Universität Leipzig, Germany  
droste@informatik.uni-leipzig.de

Paul Gastin 

LSV, ENS Paris-Saclay, CNRS, Université Paris-Saclay, France  
paul.gastin@ens-paris-saclay.fr

---

## Abstract

By fundamental results of Schützenberger, McNaughton and Papert from the 1970s, the classes of first-order definable and aperiodic languages coincide. Here, we extend this equivalence to a quantitative setting. For this, weighted automata form a general and widely studied model. We define a suitable notion of a weighted first-order logic. Then we show that this weighted first-order logic and aperiodic polynomially ambiguous weighted automata have the same expressive power. Moreover, we obtain such equivalence results for suitable weighted sublogics and finitely ambiguous or unambiguous aperiodic weighted automata. Our results hold for general weight structures, including all semirings, average computations of costs, bounded lattices, and others.

**2012 ACM Subject Classification** Theory of computation → Quantitative automata; Theory of computation → Logic and verification

**Keywords and phrases** Weighted automata, weighted logic, aperiodic automata, first-order logic, unambiguous, finitely ambiguous, polynomially ambiguous

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2019.76

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/1902.08149>

**Funding** *Manfred Droste*: Partly supported by a visiting professorship at ENS Paris-Saclay  
*Paul Gastin*: Partly supported by the DFG Research Training Group QuantLA.

## 1 Introduction

Fundamental results of Schützenberger, McNaughton and Papert established that aperiodic, star-free and first-order definable languages, respectively, coincide [40, 32]. In this paper, we develop such an equivalence in a quantitative setting, i.e., for suitable notions of aperiodic weighted automata and weighted first-order logic.

Already Schützenberger [39] investigated weighted automata and characterized their behaviors as rational formal power series. Weighted automata can be viewed as classical finite automata in which the transitions are equipped with weights. These weights could model, e.g., the cost, reward or probability of executing a transition. The wide flexibility of this automaton model soon led to a wealth of extensions and applications, cf. [38, 28, 2, 36, 15]. Whereas traditionally weights are taken from a semiring, recently, motivated by practical examples, also average and discounted computations of weights were considered, cf. [8, 7].

In the boolean setting, the seminal Büchi-Elgot-Trakhtenbrot theorem [6, 21, 41] established the expressive equivalence of finite automata and monadic second-order logic (MSO). A weighted monadic second-order logic with the same expressive power as weighted automata was developed in [12, 13]. This led to various extensions to weighted automata and weighted logics on trees [19], infinite words [18], timed words [34], pictures [22], graphs [10], nested words [11], and data words [1], but also for more complicated weight structures including



© Manfred Droste and Paul Gastin;  
licensed under Creative Commons License CC-BY

44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019).

Editors: Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen; Article No. 76; pp. 76:1–76:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

average and discounted calculations [16] or multi-weights [17]. Recently, in [23], weighted MSO logic was revisited with a more structured syntax, called *core-wMSO*, and shown to be expressively equivalent to weighted automata, while permitting a uniform approach to semirings and these more complicated weight structures.

Here, we consider the first-order fragment *wFO* of this weighted logic. It extends the full classical boolean first-order logic quantitatively by adding weight constants and if-then-else applications, followed by a first-order (universal) product and then further if-then-else applications, finite sums, or first-order (existential) sums. We will show that its expressive power leads to aperiodic weighted automata which, moreover, are polynomially ambiguous. Natural subsets of connectives will correspond to unambiguous or finitely ambiguous aperiodic weighted automata. These various levels of ambiguity are well-known from classical automata theory [24, 42, 26, 25].

Following the approach of [23], we take an *arbitrary* set  $R$  of weights. A path in a weighted automaton over  $R$  then has the sequence of weights of its transitions as its value. The *abstract semantics* of the weighted automaton is defined as the function mapping each non-empty word to the multiset of weight sequences of the successful paths executing the given word. Correspondingly, we will define the abstract semantics of *wFO* sentences also as functions mapping non-empty words to multisets of sequences of weights. Our main result will be the following.

► **Theorem 1.** *Let  $\Sigma$  be an alphabet and  $R$  a set of weights. Then the following classes of weighted automata and weighted first-order logics are expressively equivalent:*

1. *Aperiodic polynomially ambiguous weighted automata (wA) and wFO sentences,*
2. *Aperiodic finitely ambiguous wA and wFO sentences without first-order sums,*
3. *Aperiodic unambiguous wA and wFO sentences without binary or first-order sums.*

Note that these characterizations hold without any restrictions on the weights. The above result applies not only to the abstract semantics. As immediate consequence, we obtain corresponding expressive equivalence results for classical weighted automata over arbitrary (even non-commutative) semirings, or with average or discounted calculations of weights, or bounded lattices as in multi-valued logics. All our constructions are effective. In fact, given a *wFO* sentence and deterministic aperiodic automata for its boolean subformulas, we can construct an equivalent aperiodic weighted automaton of exponential size. We give typical examples for our constructions. The class of arbitrary aperiodic weighted automata and its subclasses of polynomially resp. finitely ambiguous or unambiguous weighted automata form a proper hierarchy for each of the following semirings: natural numbers  $\mathbb{N}_{+, \times}$ , the max-plus-semiring  $\mathbb{N}_{\max, +}$  and the min-plus semiring  $\mathbb{N}_{\min, +}$  [14].

It should be noticed that standard constructions used to establish equivalence between automata and MSO logic cannot be applied. Indeed, starting from an automaton  $\mathcal{A}$ , one usually constructs an *existential* MSO sentence where the existential set quantifications are used to guess an accepting run and the easy first-order kernel is used to check that this guess indeed defines an accepting run. Here, we cannot use quantifications  $\exists X$  over set variables  $X$ , or their weighted equivalent  $\sum_X$ . Instead, we take advantage of the fine structure of possible paths of polynomially ambiguous automata, namely the fact that it must be unambiguous on strongly connected components (SCC-unambiguous), as employed for different goals already in [24, 42]. We first give a new construction of a *wFO* sentence without sums starting from an aperiodic and unambiguous automaton. Then, we extend the construction to polynomially ambiguous aperiodic automata using first-order sums  $\sum_x$  to guess positions where the run switches between the unambiguous SCCs. For part 2 of Theorem 1, we also prove that

for each *aperiodic* finitely ambiguous weighted automaton we can construct finitely many *aperiodic* unambiguous weighted automata whose disjoint union has the same semantics.

Again, for the implication from weighted formulas to weighted automata, we cannot simply use standard constructions which crucially rely on the fact that functions defined by weighted automata are closed under morphic images. This was used to handle first-order sums  $\sum_x$  and second-order sums  $\sum_X$ , but also in the more involved proof for the first-order product  $\prod_x$  applied to finitely valued weighted automata. But it is well-known that aperiodic languages are not closed under morphic images. Handling the first-order product  $\prod_x$  requires a completely new and highly non-trivial proof preserving aperiodicity properties.

Detailed proofs and additional examples are given in the full version [14].

**Related work.** In [27], polynomially ambiguous, finitely ambiguous and unambiguous weighted automata (without assuming aperiodicity) over commutative semirings were shown to be expressively equivalent to suitable fragments of weighted monadic second order logic. This was further extended in [33] to cover polynomial degrees and weighted tree automata.

A hierarchy of these classes of weighted automata (again without assuming aperiodicity) over the max-plus semiring was described in [26]. As a consequence of pumping lemmas for weighted automata, a similar hierarchy was obtained in [30] for the min-plus semiring.

We note that in [13, 20], an equivalence result for full weighted first-order logic was given, but only for very particular classes of semirings or strong bimonoids as weight structures.

A characterization of the full weighted first-order logic with transitive closure by weighted pebble automata was obtained in [5]. An equivalence result for fragments of weighted first-order logic, weighted LTL and weighted counter-free automata over the max-plus semiring with discounting was given in [29].

## 2 Preliminaries

A non-deterministic automaton is a tuple  $\mathcal{A} = (Q, \Sigma, \Delta)$  where  $Q$  is a finite set of states,  $\Sigma$  is a finite alphabet, and  $\Delta \subseteq Q \times \Sigma \times Q$  is the set of transitions. The automaton  $\mathcal{A}$  is complete if  $\Delta(q, a) \neq \emptyset$  for all  $q \in Q$  and  $a \in \Sigma$ . A run  $\rho$  of  $\mathcal{A}$  is a nonempty sequence of transitions  $\delta_1 = (p_1, a_1, q_1)$ ,  $\delta_2 = (p_2, a_2, q_2)$ ,  $\dots$ ,  $\delta_n = (p_n, a_n, q_n)$  such that  $q_i = p_{i+1}$  for all  $1 \leq i < n$ . We say that  $\rho$  is a run from state  $p_1$  to state  $q_n$  and that  $\rho$  reads, or has label, the word  $a_1 a_2 \dots a_n \in \Sigma^+$ . We denote by  $\mathcal{L}(\mathcal{A}_{p,q}) \subseteq \Sigma^*$  the set of labels of runs of  $\mathcal{A}$  from  $p$  to  $q$ . When  $p = q$ , we include the empty word  $\varepsilon$  in  $\mathcal{L}(\mathcal{A}_{p,q})$  and say that  $\varepsilon$  labels the empty run from  $p$  to  $p$ .

An automaton with accepting conditions is a tuple  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  where  $(Q, \Sigma, \Delta)$  is a non-deterministic automaton,  $I, F \subseteq Q$  are the sets of initial and final states respectively. The language defined by the automaton is  $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_{I,F}) = \bigcup_{p \in I, q \in F} \mathcal{L}(\mathcal{A}_{p,q})$ . Subsequently, we also consider automata with several accepting sets  $F, G, \dots$  so that the same automaton may define several languages  $\mathcal{L}(\mathcal{A}_{I,F})$ ,  $\mathcal{L}(\mathcal{A}_{I,G})$ ,  $\dots$ . An automaton  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  is *deterministic* if  $I = \{\iota\}$  is a singleton and the set  $\Delta$  of transitions is a partial function: for all  $(p, a) \in Q \times \Sigma$  there is at most one state  $q \in Q$  such that  $(p, a, q) \in \Delta$ .

Next, we consider degrees of ambiguity of automata. A run in an automaton is *successful*, if it leads from an initial to a final state. The automaton  $\mathcal{A}$  is called *polynomially ambiguous* if there is a polynomial  $p$  such that for each  $w \in \Sigma^+$  the number of successful paths in  $\mathcal{A}$  for  $w$  is at most  $p(|w|)$ . Then,  $\mathcal{A}$  is *finitely ambiguous* if  $p$  can be taken to be a constant. Further, for an integer  $k \geq 1$ ,  $\mathcal{A}$  is *k-ambiguous* if  $p = k$ , and *unambiguous* means 1-ambiguous. Notice that *k-ambiguous* implies  $(k + 1)$ -ambiguous. An automaton  $\mathcal{A}$  is at most *exponentially ambiguous*.

A non-deterministic automaton  $\mathcal{A} = (Q, \Sigma, \Delta)$  is *aperiodic* if there exists an integer  $m \geq 1$ , called aperiodicity index, such that for all states  $p, q \in Q$  and all words  $u \in \Sigma^+$ , we have  $u^m \in \mathcal{L}(\mathcal{A}_{p,q})$  iff  $u^{m+1} \in \mathcal{L}(\mathcal{A}_{p,q})$ . In other words, the non-deterministic automaton  $\mathcal{A}$  is aperiodic iff its transition monoid  $\text{Tr}(\mathcal{A})$  is aperiodic. It is well-known that aperiodic languages coincide with first-order definable languages, cf. [40, 32, 9].

The syntax of first-order logic is given in Section 4 (FO). The semantics is defined by structural induction on the formula and requires an interpretation of the free variables. Let  $\mathcal{V} = \{y_1, \dots, y_n\}$  be a finite set of first-order variables. Given a nonempty word  $u \in \Sigma^+$ , we let  $\text{pos}(u) = \{1, \dots, |u|\}$  be the set of positions of  $u$ . A valuation or interpretation is a map  $\sigma: \mathcal{V} \rightarrow \text{pos}(u)$  assigning positions of  $u$  to variables in  $\mathcal{V}$ . For a first-order formula  $\varphi$  having free variables contained in  $\mathcal{V}$ , we write  $u, \sigma \models \varphi$  when the word  $u$  satisfies  $\varphi$  under the interpretation defined by  $\sigma$ . When  $\varphi$  is a *sentence*, the valuation  $\sigma$  is not needed and we simply write  $u \models \varphi$ .

We extend the classical semantics by defining when the empty word  $\varepsilon$  satisfies a sentence. We have  $\varepsilon \models \top$  and if  $\forall x\psi$  is a sentence then  $\varepsilon \models \forall x\psi$ . The semantics  $\varepsilon \models \varphi$  is extended to all sentences  $\varphi$  since they are boolean combinations of the basic cases above. Notice that if  $\varphi$  has free variables then  $\varepsilon \models \varphi$  is not defined. When  $\varphi$  is a sentence we denote by  $\mathcal{L}(\varphi) \subseteq \Sigma^*$  the set of words satisfying  $\varphi$ . Notice that  $\mathcal{L}(\forall x\perp) = \{\varepsilon\}$  where  $\perp = \neg\top$ .

► **Theorem 2** ([40, 32, 9]). *Let  $\mathcal{A}$  be an aperiodic non-deterministic automaton. For all states  $p, q$  of  $\mathcal{A}$  we can construct a first-order sentence  $\varphi_{p,q}$  such that  $\mathcal{L}(\mathcal{A}_{p,q}) = \mathcal{L}(\varphi_{p,q})$ .*

For the converse of Theorem 2, we need a stronger statement to deal with formulas having free variables. As usual, we encode a pair  $(u, \sigma)$  where  $u \in \Sigma^+$  is a nonempty word and  $\sigma: \mathcal{V} \rightarrow \text{pos}(u)$  is a valuation by a word  $\bar{u}$  over the extended alphabet  $\Sigma_{\mathcal{V}} = \Sigma \times \{0, 1\}^{\mathcal{V}}$ . A word  $\bar{u}$  over  $\Sigma_{\mathcal{V}}$  is a *valid* encoding if for each variable  $y \in \mathcal{V}$ , its projection on the  $y$ -component belongs to  $0^*10^*$ . Throughout the paper, we identify a valid word  $\bar{u}$  with its encoded pair  $(u, \sigma)$ .

► **Theorem 3** ([40, 32, 9]). *For each FO-formula  $\varphi$  having free variables contained in  $\mathcal{V}$ , we can build a deterministic, complete and aperiodic automaton  $\mathcal{A}_{\varphi, \mathcal{V}} = (Q, \Sigma_{\mathcal{V}}, \Delta, \iota, F, G)$  over the extended alphabet  $\Sigma_{\mathcal{V}}$  such that for all words  $\bar{u} \in \Sigma_{\mathcal{V}}^+$  we have:*

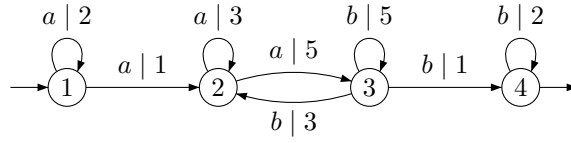
- $\Delta(\iota, \bar{u}) \in F$  iff  $\bar{u}$  is a valid encoding of a pair  $(u, \sigma)$  with  $(u, \sigma) \models \varphi$ ,
- $\Delta(\iota, \bar{u}) \in G$  iff  $\bar{u}$  is a valid encoding of a pair  $(u, \sigma)$  with  $(u, \sigma) \models \neg\varphi$ ,
- $\Delta(\iota, \bar{u}) \notin F \cup G$  otherwise, i.e., iff  $\bar{u}$  is not a valid encoding of a pair  $(u, \sigma)$ .

Given  $u \in \Sigma^+$  and integers  $k, \ell$ , we denote by  $u[k, \ell]$  the factor of  $u$  between positions  $k$  and  $\ell$ . By convention  $u[k, \ell] = \varepsilon$  is the empty word when  $\ell < k$  or  $\ell = 0$  or  $k > |u|$ .

We will apply the equivalence of Theorem 2 to prefixes, infixes or suffixes of words. Towards this, we use the classical *relativization* of sentences. Let  $\varphi$  be a first-order sentence and let  $x, y \in \mathcal{V}$  be first-order variables. We define below the relativizations  $\varphi^{<x}$ ,  $\varphi^{(x,y)}$  and  $\varphi^{>y}$  so that for all words  $u \in \Sigma^+$ , and all positions  $i, j \in \text{pos}(u) = \{1, \dots, |u|\}$  we have

$$\begin{array}{lll}
 u, x \mapsto i \models \varphi^{<x} & \text{iff} & u[1, i-1] \models \varphi \\
 u, x \mapsto i, y \mapsto j \models \varphi^{(x,y)} & \text{iff} & u[i+1, j-1] \models \varphi \\
 u, x \mapsto j \models \varphi^{>x} & \text{iff} & u[j+1, |u|] \models \varphi
 \end{array}$$

Notice that, when  $i = 1$  or  $j \leq i + 1$  or  $j = |u|$ , the relativization is on the empty word, this is why we had to define when  $\varepsilon \models \psi$  for sentences  $\psi$ . The relativization is defined by



■ **Figure 1** A weighted automaton, which is both aperiodic and polynomially ambiguous.

structural induction on the formulas as follows:

$$\begin{aligned} \top^{<x} &= \top & (P_a(z))^{<x} &= P_a(z) & (y \leq z)^{<x} &= (y \leq z) \\ (\neg\psi)^{<x} &= \neg(\psi^{<x}) & (\psi_1 \wedge \psi_2)^{<x} &= \psi_1^{<x} \wedge \psi_2^{<x} & (\forall z\psi)^{<x} &= \forall z(z < x \implies \psi^{<x}) \end{aligned}$$

The relativizations  $\varphi^{(x,y)}$  and  $\varphi^{>x}$  are defined similarly. Notice that when  $\varphi$  is a sentence, i.e., a boolean combination of formulas of the form  $\top$  or  $\forall z\psi$ , then the above equivalences hold even when  $i = 1$  for  $\varphi^{<x}$ , or when  $i = |u|$  for  $\varphi^{>x}$ , or when  $j \leq i + 1$  for  $\varphi^{(x,y)}$ .

### 3 Weighted Automata

Given a set  $X$ , we let  $\mathbb{N}\langle X \rangle$  be the collection of all finite multisets over  $X$ , i.e., all functions  $f: X \rightarrow \mathbb{N}$  such that  $f(x) \neq 0$  only for finitely many  $x \in X$ . The union  $f \uplus g$  of two multisets  $f, g \in \mathbb{N}\langle X \rangle$  is defined by pointwise addition of functions:  $(f \uplus g)(x) = f(x) + g(x)$  for  $x \in X$ .

For a set  $\mathbb{R}$  of weights, an  $\mathbb{R}$ -weighted automaton over  $\Sigma$  is a tuple  $\mathcal{A} = (Q, \Sigma, \Delta, \text{wt})$  where  $(Q, \Sigma, \Delta)$  is a non-deterministic automaton and  $\text{wt}: \Delta \rightarrow \mathbb{R}$  assigns a weight to every transition. The weight sequence of a run  $\rho = \delta_1 \delta_2 \cdots \delta_n$  is  $\text{wt}(\rho) = \text{wt}(\delta_1) \text{wt}(\delta_2) \cdots \text{wt}(\delta_n) \in \mathbb{R}^+$ . The *abstract semantics* of  $\mathcal{A}$  from state  $p$  to state  $q$  is the map  $\{\mathcal{A}_{p,q}\}: \Sigma^+ \rightarrow \mathbb{N}\langle \mathbb{R}^+ \rangle$  which assigns to a word  $u \in \Sigma^+$  the multiset of weight sequences of runs from  $p$  to  $q$  with label  $u$ :

$$\{\mathcal{A}_{p,q}\}(u) = \{\{\text{wt}(\rho) \mid \rho \text{ is a run from } p \text{ to } q \text{ with label } u\}\}.$$

Notice that  $\{\mathcal{A}_{p,q}\}(u) = \emptyset$  is the empty multiset when there are no runs of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ , i.e., when  $u \notin \mathcal{L}(\mathcal{A}_{p,q})$ . When we consider a weighted automaton  $\mathcal{A} = (Q, \Sigma, \Delta, \text{wt}, I, F)$  with initial and final sets of states, for all  $u \in \Sigma^+$  the semantics  $\{\mathcal{A}\}$  is defined as the multiset union:  $\{\mathcal{A}\}(u) = \biguplus_{p \in I, q \in F} \{\mathcal{A}_{p,q}\}(u)$ . Hence,  $\{\mathcal{A}\}$  assigns to every word  $u \in \Sigma^+$  the multiset of all weight sequences of accepting runs of  $\mathcal{A}$  reading  $u$ . The support of  $\mathcal{A}$  is the set of words  $u \in \Sigma^+$  such that  $\{\mathcal{A}\}(u) \neq \emptyset$ , i.e.,  $\text{supp}(\mathcal{A}) = \mathcal{L}(\mathcal{A})$ .

For instance, consider the weighted automaton  $\mathcal{A}$  of Figure 1. We have  $\text{supp}(\mathcal{A}) = a^+ a (a+b)^* b^+$ . Consider  $w = a^m (ba)^n b^p$  with  $m > 1$  and  $p > 0$ . We have  $w \in \text{supp}(\mathcal{A})$  and  $\{\mathcal{A}\}(w) = \{\{2^{k-1} \cdot 1 \cdot 3^{m-k-1} \cdot 5 \cdot (3 \cdot 5)^n \cdot 5^{\ell-1} \cdot 1 \cdot 2^{p-\ell} \mid 1 \leq k < m \text{ and } 1 \leq \ell \leq p\}\}$ .

A concrete semantics over semirings, or valuation monoids, or valuation structures can be obtained from the abstract semantics defined above by applying the suitable aggregation operator  $\text{aggr}: \mathbb{N}\langle \mathbb{R}^+ \rangle \rightarrow S$  as explained in [23], see also [14]. For the natural semiring  $(\mathbb{N}, +, \times, 0, 1)$ , the sum-product aggregation operator  $\text{aggr}_{\text{sp}}(f)$  gives the sum over all sequences  $s_1 s_2 \cdots s_k$  in the multiset  $f$  of the products  $s_1 \times s_2 \times \cdots \times s_k$  in  $\mathbb{N}$ . We continue the example with the automaton  $\mathcal{A}$  of Figure 1 and the word  $w = a^m (ba)^n b^p$  with  $m > 1$  and  $p > 0$ . The concrete semantics is given by

$$\llbracket \mathcal{A} \rrbracket(w) = \text{aggr}_{\text{sp}}(\{\mathcal{A}\}(w)) = \sum_{1 \leq k < m} \sum_{1 \leq \ell \leq p} 2^{k-1+p-\ell} 3^{m-k-1+n} 5^{n+\ell}.$$

In further examples, we also use the max-plus semiring  $\mathbb{N}_{\max,+} = (\mathbb{N} \cup \{-\infty\}, \max, +, -\infty, 0)$  and the min-plus semiring  $\mathbb{N}_{\min,+} = (\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$ .

Now, we investigate finitely ambiguous weighted automata. It was shown in [26] that over the max-plus semiring  $\mathbb{N}_{\max,+}$  they are expressively equivalent to finite disjoint unions of unambiguous weighted automata. Moreover, it was proved in [37] that a  $K$ -valued rational transducer can be decomposed into  $K$  unambiguous transducers. In particular this implies that a  $K$ -ambiguous weighted automaton can be decomposed into  $K$  unambiguous weighted automata. We can prove that the same holds for aperiodic weighted automata [14].

► **Theorem 4.** *Let  $K \geq 1$ . Given an aperiodic  $K$ -ambiguous weighted automaton  $\mathcal{A}$ , we can construct aperiodic unambiguous weighted automata  $\mathcal{B}_1, \dots, \mathcal{B}_K$  such that  $\{\mathcal{A}\} = \{\mathcal{B}_1 \uplus \dots \uplus \mathcal{B}_K\} = \{\mathcal{B}_1\} \uplus \dots \uplus \{\mathcal{B}_K\}$ .*

Our proof is based on lexicographic ordering of runs. The proof of [37] uses lexicographic coverings. It would be interesting to see whether this proof also preserves aperiodicity and to compare the complexity of the constructions.

## 4 Weighted First-Order Logic

In this section, we define the syntax and semantics of our weighted first-order logic. In [12, 13], weighted MSO used the classical syntax of MSO logic; only the semantics over a semiring was changed to use sums for disjunction and existential quantifications, and products for conjunctions and universal quantifications. The possibility to express boolean properties in wMSO was obtained via so-called unambiguous formulae. To improve readability, a more structured syntax was later used [3, 16, 27], separating a boolean MSO layer with classical boolean semantics from the higher level of weighted formulas using *products* ( $\prod_X, \prod_x$  corresponding to  $\forall X, \forall x$ ) and *sums* ( $\sum_X, \sum_x$  corresponding to  $\exists X, \exists x$ ) with quantitative semantics. As shown in [12, 13], in general, to retain equivalence with weighted automata, wMSO has to be restricted. Products  $\prod_X$  over set variables are disallowed, and first-order products  $\prod_x$  must be restricted to finitely valued series where the pre-image of each value is recognizable. This basically means that first-order products cannot be nested or applied after first-order or second-order sums  $\sum_x$  or  $\sum_X$ . This motivated the equivalent and even more structured syntax of *core-wMSO* introduced in [23].

As in Section 3, we consider a set  $\mathbb{R}$  of weights. The syntax of wFO is obtained from core-wMSO by removing set variables, set quantifications and set sums. In addition to the classical boolean first-order logic (FO), it has two weighted layers. Step formulas defined in (step-wFO) consist of constants and if-then-else applications, where the conditions are formulated in boolean first-order logic. Finally, wFO builds on this by performing products of step formulas and then applying if-then-else, finite sums, or existential sums.

$$\varphi ::= \top \mid P_a(x) \mid x \leq y \mid \neg\varphi \mid \varphi \wedge \varphi \mid \forall x\varphi \quad (\text{FO})$$

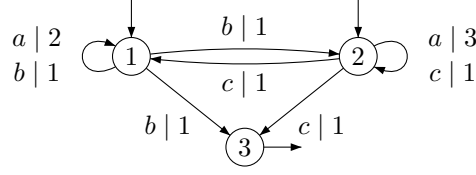
$$\Psi ::= r \mid \varphi ? \Psi : \Psi \quad (\text{step-wFO})$$

$$\Phi ::= \mathbf{0} \mid \prod_x \Psi \mid \varphi ? \Phi : \Phi \mid \Phi + \Phi \mid \sum_x \Phi \quad (\text{wFO})$$

with  $a \in \Sigma$ ,  $r \in \mathbb{R}$  and  $x, y$  first-order variables.

The semantics of step-wFO formulas is defined inductively. As above, let  $u \in \Sigma^+$  be a nonempty word and  $\sigma: \mathcal{V} \rightarrow \text{pos}(u) = \{1, \dots, |u|\}$  be a valuation. For step-wFO formulas whose free variables are contained in  $\mathcal{V}$ , we define the  $\mathcal{V}$ -semantics as

$$\llbracket r \rrbracket_{\mathcal{V}}(u, \sigma) = r \quad \llbracket \varphi ? \Psi_1 : \Psi_2 \rrbracket_{\mathcal{V}}(u, \sigma) = \begin{cases} \llbracket \Psi_1 \rrbracket_{\mathcal{V}}(u, \sigma) & \text{if } u, \sigma \models \varphi \\ \llbracket \Psi_2 \rrbracket_{\mathcal{V}}(u, \sigma) & \text{otherwise.} \end{cases}$$



■ **Figure 2** A weighted automaton, which is both aperiodic and unambiguous.

Notice that the semantics of a **step-wFO** formula is always a single weight from  $\mathbb{R}$ .

For **wFO** formulas  $\Phi$  whose free variables are contained in  $\mathcal{V}$ , we define the  $\mathcal{V}$ -semantics  $\{\Phi\}_{\mathcal{V}}: \Sigma_{\mathcal{V}}^+ \rightarrow \mathbb{N}\langle\mathbb{R}^+\rangle$ . First, we let  $\{\Phi\}_{\mathcal{V}}(\bar{u}) = \emptyset$  be the empty multiset when  $\bar{u} \in \Sigma_{\mathcal{V}}^+$  is not a valid encoding of a pair  $(u, \sigma)$ . Assume now that  $\bar{u} = (u, \sigma)$  is a valid encoding of a nonempty word  $u \in \Sigma^+$  and a valuation  $\sigma: \mathcal{V} \rightarrow \text{pos}(u)$ . The semantics of **wFO** formulas is also defined inductively:  $\{\mathbf{0}\}_{\mathcal{V}}(u, \sigma) = \emptyset$  is the empty multiset, and

$$\begin{aligned} \{\prod_x \Psi\}_{\mathcal{V}}(u, \sigma) &= \{\{r_1 r_2 \cdots r_{|u|}\}\} \text{ where } r_i = \llbracket \Psi \rrbracket_{\mathcal{V} \cup \{x\}}(u, \sigma[x \mapsto i]) \text{ for } 1 \leq i \leq |u| \\ \{\varphi? \Phi_1 : \Phi_2\}_{\mathcal{V}}(u, \sigma) &= \begin{cases} \{\Phi_1\}_{\mathcal{V}}(u, \sigma) & \text{if } u, \sigma \models \varphi \\ \{\Phi_2\}_{\mathcal{V}}(u, \sigma) & \text{otherwise} \end{cases} \\ \{\Phi_1 + \Phi_2\}_{\mathcal{V}}(u, \sigma) &= \{\Phi_1\}_{\mathcal{V}}(u, \sigma) \uplus \{\Phi_2\}_{\mathcal{V}}(u, \sigma) \\ \{\sum_x \Phi\}_{\mathcal{V}}(u, \sigma) &= \bigoplus_{i \in \text{pos}(u)} \{\Phi\}_{\mathcal{V} \cup \{x\}}(u, \sigma[x \mapsto i]). \end{aligned}$$

The semantics of the product (first line), is a singleton multiset which consists of a weight sequence whose length is  $|u|$ . We deduce that all weight sequences in a multiset  $\{\Phi\}_{\mathcal{V}}(u, \sigma)$  have the same length and  $\{\Phi\}_{\mathcal{V}}(u, \sigma) \in \mathbb{N}\langle\mathbb{R}^{|u|}\rangle$ . We simply write  $\llbracket \Psi \rrbracket$  and  $\{\Phi\}$  when the set  $\mathcal{V}$  of variables is clear from the context.

As explained in Section 3, applying an aggregation function allows to recover the semantics  $\llbracket \Phi \rrbracket$  over semirings such as  $\mathbb{N}_{+, \times}$ ,  $\mathbb{N}_{\max, +}$ , etc. For instance, consider the function  $f: \{a, b\}^+ \rightarrow \mathbb{N}$  which assign to a word  $w \in \{a, b\}^+$  the length of the maximal  $a$ -block, i.e.,  $f(w) = n$  if  $a^n$  is a factor of  $w$  but  $a^{n+1}$  is not. Over  $\mathbb{N}_{\max, +}$ , we have  $f = \llbracket \Phi \rrbracket$  where

$$\Phi = \sum_{y, z} (\forall u (y \leq u \leq z) \rightarrow P_a(u)) ? (\prod_x (y \leq x \leq z) ? 1 : 0) : (\prod_x 0).$$

We refer to [13] for further examples of quantitative specifications in weighted logic.

## 5 From Weighted Automata to Weighted FO

We say that a non-deterministic automaton  $\mathcal{A} = (Q, \Sigma, \Delta)$  is *unambiguous from state  $p$  to state  $q$*  if for all words  $u \in \Sigma^+$ , there is at most one run of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ .

► **Theorem 5.** *Let  $\mathcal{A}$  be an aperiodic weighted automaton which is unambiguous from  $p$  to  $q$ . We can construct a wFO sentence  $\Phi_{p,q} = \varphi_{p,q} ? \prod_x \Psi_{p,q} : \mathbf{0}$  where  $\varphi_{p,q}$  is a first-order sentence and  $\Psi_{p,q}(x)$  is a step-wFO formula with a single free variable  $x$  such that  $\{\mathcal{A}_{p,q}\} = \{\Phi_{p,q}\}$ .*

Before proving Theorem 5, we start with an example. The automaton  $\mathcal{A}$  of Figure 2 is unambiguous and it accepts the language  $\mathcal{L}(\mathcal{A}) = (a^*b + a^*c)^+ = (a + b + c)^*(b + c)$ . We define a wFO sentence  $\Phi_{1,3} = \varphi_{1,3} ? \prod_x \Psi_{1,3}(x) : \mathbf{0}$  as follows. The FO sentence  $\varphi_{1,3}$  checks that  $\mathcal{A}$  has a run from state 1 to state 3 on the input word  $w$ , i.e., that  $w \in a^*b(a^*b + a^*c)^*$ :

$$\varphi_{1,3} = \exists y (P_b(y) \wedge \forall z (z < y \implies P_a(z))) \wedge \exists y (\neg P_a(y) \wedge \forall z (z \leq y))$$

When this is the case, the step-wFO formula  $\Psi_{1,3}(x)$  computes the weight of the transition taken at a position  $x$  in the input word:

$$\Psi_{1,3}(x) = (P_b(x) \vee P_c(x)) ? 1 : \exists y (x < y \wedge P_b(y) \wedge \forall z (x < z < y \implies P_a(z))) ? 2 : 3.$$

Notice that the same formula  $\Psi = \Psi_{2,3} = \Psi_{1,3}$  also allows to compute the sequence of weights for the accepting runs starting in state 2. Therefore,  $\mathcal{A}$  is equivalent to the wFO sentence

$$\Phi = \exists y (\neg P_a(y) \wedge \forall z (z \leq y)) ? \prod_x \Psi(x) : \mathbf{0}.$$

**Proof of Theorem 5.** Let  $\mathcal{A} = (Q, \Sigma, \Delta, \text{wt})$  be the aperiodic weighted automaton. By Theorem 2, for every pair of states  $r, s \in Q$  there is a first-order sentence  $\varphi_{r,s}$  such that  $\mathcal{L}(\mathcal{A}_{r,s}) = \mathcal{L}(\varphi_{r,s})$ . This gives in particular the first-order sentence  $\varphi_{p,q}$  which is used in  $\Phi_{p,q}$ .

▷ **Claim 6.** We can construct a step-wFO formula  $\Psi_{p,q}(x)$  such that for each word  $u \in \mathcal{L}(\mathcal{A}_{p,q})$  and each position  $1 \leq i \leq |u|$  in the word  $u$ , we have  $\llbracket \Psi_{p,q} \rrbracket(u, x \mapsto i) = \text{wt}(\delta)$  where  $\delta$  is the  $i$ th transition of the unique run  $\rho$  of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ .

Before proving this claim, let us show how we can deduce the statement of Theorem 5. Clearly, if a word  $u \in \Sigma^+$  is not in  $\mathcal{L}(\mathcal{A}_{p,q})$  then we have  $\{\mathcal{A}_{p,q}\}(u) = \emptyset = \{\Phi_{p,q}\}(u)$ . Consider now a word  $u = a_1 a_2 \cdots a_n \in \mathcal{L}(\mathcal{A}_{p,q})$  and the unique run  $\rho = \delta_1 \delta_2 \cdots \delta_n$  of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ . We have  $\{\mathcal{A}_{p,q}\}(u) = \{\{\text{wt}(\delta_1)\text{wt}(\delta_2) \cdots \text{wt}(\delta_n)\}\} = \{\prod_x \Psi_{p,q}\}(u)$  where the second equality follows from Claim 6. We deduce that  $\{\mathcal{A}_{p,q}\} = \{\Phi_{p,q}\}$ .

We turn now to the proof of Claim 6. Let  $\delta = (r, a, s) \in \Delta$  be a transition of  $\mathcal{A}$ . We define the FO-formula with one free variable  $\varphi_\delta(x) = \varphi_{p,r}^{\leq x} \wedge P_a(x) \wedge \varphi_{s,q}^{\geq x}$ .

▷ **Claim 7.** For each word  $u \in \Sigma^+$  and position  $1 \leq i \leq |u|$ , we have  $u, x \mapsto i \models \varphi_\delta$  iff  $u \in \mathcal{L}(\mathcal{A}_{p,q})$  and  $\delta$  is the  $i$ th transition of the unique run of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ .

Indeed, assume that  $u, x \mapsto i \models \varphi_\delta$ . Then,  $u[1, i-1] \models \varphi_{p,r}$  and there is a run  $\rho'$  of  $\mathcal{A}$  from  $p$  to  $r$  with label  $u[1, i-1]$ . Notice that if  $i = 1$  then  $p = r$  and  $\rho'$  is the empty run. Similarly, from  $u[i+1, |u|] \models \varphi_{s,q}$  we deduce that there is a run  $\rho''$  of  $\mathcal{A}$  from  $s$  to  $q$  with label  $u[i+1, |u|]$ . Finally,  $u, x \mapsto i \models P_a(x)$  means that the  $i$ th letter of  $u$  is  $a$ . We deduce that  $\rho = \rho' \delta \rho''$  is a run of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ , hence  $u \in \mathcal{L}(\mathcal{A}_{p,q})$ . Moreover,  $\rho$  is the unique such run since  $\mathcal{A}$  is unambiguous from  $p$  to  $q$ . Now,  $\delta$  is the  $i$ th transition of  $\rho$ , which concludes one direction. Conversely, assume that  $u \in \mathcal{L}(\mathcal{A}_{p,q})$  and  $\delta$  is the  $i$ th transition of the unique run of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ . Then,  $u[1, i-1] \models \varphi_{p,r}$ ,  $u[i+1, |u|] \models \varphi_{s,q}$ , and the  $i$ th letter of  $u$  is  $a$ . Therefore,  $u, x \mapsto i \models \varphi_\delta$ . This concludes the proof of Claim 7.

Now, choose an arbitrary enumeration  $\delta^1, \delta^2, \dots, \delta^k$  of the transitions in  $\Delta$  and define the step-wFO formula with one free variable

$$\Psi_{p,q}(x) = \varphi_{\delta^1}(x) ? \text{wt}(\delta^1) : \varphi_{\delta^2}(x) ? \text{wt}(\delta^2) : \cdots \varphi_{\delta^k}(x) ? \text{wt}(\delta^k) : \text{wt}(\delta^k).$$

We show that this formula satisfies the property of Claim 6. Consider a word  $u \in \mathcal{L}(\mathcal{A}_{p,q})$  and a position  $1 \leq i \leq |u|$ . Let  $\delta$  be the  $i$ th transition of the unique run of  $\mathcal{A}$  from  $p$  to  $q$  with label  $u$ . By Claim 7, we have  $u, x \mapsto i \models \varphi_{\delta^j}$  iff  $\delta^j = \delta$ . Therefore,  $\llbracket \Psi_{p,q} \rrbracket(u, x \mapsto i) = \text{wt}(\delta)$ , which concludes the proof of Claim 6. ◀

► **Corollary 8.**

1. Let  $\mathcal{A}$  be an aperiodic and unambiguous weighted automaton. We can construct a wFO sentence  $\Phi$  which does not use any  $\sum_x$  operator or  $+$  operator, and such that  $\{\mathcal{A}\} = \{\Phi\}$ .
2. Let  $\mathcal{A}$  be an aperiodic and finitely ambiguous weighted automaton. We can construct a wFO sentence  $\Phi$  which does not use any  $\sum_x$  operator, and such that  $\{\mathcal{A}\} = \{\Phi\}$ .



Let  $\mathcal{A} = (Q, \Sigma, \Delta)$  be a non-deterministic automaton. Two states  $p, q \in Q$  are in the same strongly connected component (SCC), denoted  $p \approx q$ , if  $p = q$  or there exist a run of  $\mathcal{A}$  from  $p$  to  $q$  and also a run of  $\mathcal{A}$  from  $q$  to  $p$ . Note that  $\approx$  is an equivalence relation on  $Q$ . We denote by  $[p]$  the strongly connected component of  $p$ , i.e., the equivalence class of  $p$  under  $\approx$ .

The automaton  $\mathcal{A}$  is *SCC-unambiguous* if it is unambiguous on each strongly connected component, i.e.,  $\mathcal{A}$  is unambiguous from  $p$  to  $q$  for all  $p, q$  such that  $p \approx q$ . A trimmed (all states are reachable and co-reachable) and unambiguous automaton is SCC-unambiguous.

For instance, the automaton  $\mathcal{A}$  of Figure 1 has three strongly connected components:  $\{1\}$ ,  $\{2, 3\}$  and  $\{4\}$ . It is not unambiguous from 1 to 4, but it is SCC-unambiguous.

► **Proposition 9** ([35, 24] and [42] Thm 4.1). *Let  $\mathcal{A} = (Q, \Sigma, \Delta, I, F)$  be a trimmed non-deterministic automaton. Then  $\mathcal{A}$  is polynomially ambiguous iff  $\mathcal{A}$  is SCC-unambiguous.*

► **Theorem 10.** *Let  $\mathcal{A}$  be an aperiodic weighted automaton which is SCC-unambiguous. For each pair of states  $p$  and  $q$ , we can construct a wFO sentence  $\Phi_{p,q}$  such that  $\{\!\|\mathcal{A}_{p,q}\!\|\} = \{\!\|\Phi_{p,q}\!\|\}$ . Moreover, we can construct a wFO sentence  $\Phi$  such that  $\{\!\|\mathcal{A}\!\|\} = \{\!\|\Phi\!\|\}$ .*

First, we give for the weighted automaton  $\mathcal{A}$  of Figure 1 the equivalent wFO formula  $\Phi_{1,4} = \sum_{y_1} \sum_{y_2} \varphi(y_1, y_2) ? \prod_x \Psi(x, y_1, y_2) : \mathbf{0}$  where  $\varphi$  and  $\Psi$  are defined below. When reading a word  $w \in \text{supp}(\mathcal{A})$ , the automaton makes two non-deterministic choices corresponding to the positions  $y_1$  and  $y_2$  at which the transitions *switching* between the strongly connected components are taken, i.e., transition from state 1 to state 2 is taken at position  $y_1$ , and transition from 3 to 4 is taken at position  $y_2$ . Since the automaton is SCC-unambiguous, given the input word and these two positions, the run is uniquely determined. Formula  $\varphi(y_1, y_2)$  states that it is possible to take the switching transitions at positions  $y_1$  and  $y_2$ :

$$\varphi(y_1, y_2) = y_1 < y_2 \wedge \forall z (z \leq y_1 \rightarrow P_a(z)) \wedge P_a(y_1 + 1) \wedge \forall z (y_2 \leq z \rightarrow P_b(z)).$$

When this is the case, the step-wFO formula  $\Psi(x, y_1, y_2)$  computes the weight of the transition taken at a position  $x$  in the input word:

$$\Psi(x, y_1, y_2) = (x < y_1 \vee y_2 < x) ? 2 : (x = y_1 \vee x = y_2) ? 1 : P_a(x + 1) ? 3 : 5.$$

With these definitions, we obtain  $\{\!\|\mathcal{A}\!\|\} = \{\!\|\Phi_{1,4}\!\|\}$ . The proof of Theorem 10 is in [14]. Intuitively, a run from  $p$  to  $q$  reading a word  $w \in \Sigma^+$  uses a sequence of transitions switching between connected components of  $\mathcal{A}$ . The positions where these switches are taken can be described by a sequence of  $\sum_y$ -operators. Since there are only finitely many sequences of switching transitions, they can be described by a finite sum of wFO sentences.

## 6 From Weighted FO to Weighted Automata

Let  $\mathcal{A} = (Q, \Sigma, \Delta)$  and  $\mathcal{A}' = (Q', \Sigma, \Delta')$  be two non-deterministic automata over  $\Sigma$ . Assuming that  $Q \cap Q' = \emptyset$ , we define their disjoint union as  $\mathcal{A} \uplus \mathcal{A}' = (Q \uplus Q', \Sigma, \Delta \uplus \Delta')$  and their product as  $\mathcal{A} \times \mathcal{A}' = (Q \times Q', \Sigma, \Delta'')$  where  $\Delta'' = \{((p, p'), a, (q, q')) \mid (p, a, p') \in \Delta \wedge (p', a, q') \in \Delta'\}$ .

► **Lemma 11.** *The following holds.*

1. *If  $\mathcal{A}$  and  $\mathcal{A}'$  are aperiodic, then  $\mathcal{A} \uplus \mathcal{A}'$  and  $\mathcal{A} \times \mathcal{A}'$  are also aperiodic.*
2. *If  $\mathcal{A}$  and  $\mathcal{A}'$  are SCC-unambiguous, then  $\mathcal{A} \uplus \mathcal{A}'$  and  $\mathcal{A} \times \mathcal{A}'$  are also SCC-unambiguous.*

Now let  $\varphi$  be an FO-formula with free variables contained in the finite set  $\mathcal{V}$ , and let  $\mathcal{A}_{\varphi, \mathcal{V}} = (Q, \Sigma_{\mathcal{V}}, \Delta, \iota, F, G)$  be the deterministic, complete, trim and aperiodic automaton given by Theorem 3. For  $i = 1, 2$ , let  $\mathcal{A}_i = (Q_i, \Sigma_{\mathcal{V}}, \Delta_i, \text{wt}_i, I_i, F_i)$  be two weighted automata over  $\Sigma_{\mathcal{V}}$  with  $Q_1 \cap Q_2 = \emptyset$ . Define the weighted automaton  $\mathcal{A}' = (Q', \Sigma_{\mathcal{V}}, \Delta', \text{wt}', I', F')$  by

- $Q' = Q \times Q_1 \uplus Q \times Q_2$ ,  $I' = \{\iota\} \times I_1 \uplus \{\iota\} \times I_2$ ,  $F' = F \times F_1 \uplus G \times F_2$ ,
- $\Delta' = \{((p, p'), a, (q, q')) \mid (p, a, q) \in \Delta \text{ and } (p', a, q') \in \Delta_1 \cup \Delta_2\}$ , and  $\text{wt}'((p, p'), a, (q, q')) = \text{wt}_i(p', a, q')$  if  $(p', a, q') \in \Delta_i$  for  $i = 1, 2$ .

► **Lemma 12.** For each  $\bar{u} \in \Sigma_{\mathcal{V}}^+$ , we have

$$\llbracket \mathcal{A}' \rrbracket(\bar{u}) = \begin{cases} \llbracket \mathcal{A}_1 \rrbracket(\bar{u}), & \text{if } \bar{u} \text{ is valid and } \bar{u} \models \varphi, \\ \llbracket \mathcal{A}_2 \rrbracket(\bar{u}), & \text{if } \bar{u} \text{ is valid and } \bar{u} \not\models \varphi, \\ \emptyset, & \text{if } \bar{u} \text{ is not valid.} \end{cases}$$

Moreover, if  $\mathcal{A}_1$  and  $\mathcal{A}_2$  are aperiodic (resp. unambiguous, SCC-unambiguous) then so is  $\mathcal{A}'$ .

Let  $\mathcal{V}$  be a finite set of first-order variables and let  $\mathcal{V}' = \mathcal{V} \cup \{y\}$  where  $y \notin \mathcal{V}$ . Given a word  $\bar{w} \in \Sigma_{\mathcal{V}}^+$  and a position  $i \in \text{pos}(w)$ , we denote by  $(\bar{w}, y \mapsto i)$  the word over  $\Sigma_{\mathcal{V}'}$  whose projection on  $\Sigma_{\mathcal{V}}$  is  $\bar{w}$  and projection on the  $y$ -component is  $0^{i-1}10^{|\bar{w}|-i}$ , i.e., has a unique 1 on position  $i$ . Given a function  $A: \Sigma_{\mathcal{V}'}^+ \rightarrow \mathbb{N}\langle X \rangle$ , we define the function  $\sum_y A: \Sigma_{\mathcal{V}}^+ \rightarrow \mathbb{N}\langle X \rangle$  for  $\bar{w} \in \Sigma_{\mathcal{V}}^+$  by  $(\sum_y A)(\bar{w}) = \biguplus_{i \in \text{pos}(w)} A(\bar{w}, y \mapsto i)$ .

► **Lemma 13.** Let  $\mathcal{A}$  be a weighted automaton over  $\Sigma_{\mathcal{V}'}$ . We can construct a weighted automaton  $\mathcal{A}'$  over  $\Sigma_{\mathcal{V}}$  such that  $\llbracket \mathcal{A}' \rrbracket = \sum_y \llbracket \mathcal{A} \rrbracket$ . Moreover, if  $\mathcal{A}$  is aperiodic then  $\mathcal{A}'$  is also aperiodic, and if  $\mathcal{A}$  is SCC-unambiguous then  $\mathcal{A}'$  is also SCC-unambiguous.

We turn now to one of our main results: given a step-wFO formula  $\Psi$ , we can construct a weighted automaton for  $\prod_x \Psi$  which is both aperiodic and unambiguous.

When weights are uninterpreted, a weighted automaton  $\mathcal{A} = (Q, \Sigma, \Delta, \text{wt}, I, F)$  is a letter-to-letter transducer from its input alphabet  $\Sigma$  to the output alphabet  $\mathbb{R}$ . If in addition the input automaton is unambiguous, then we have a functional transducer. In the following, we will construct such functional transducers using the boolean output alphabet  $\mathbb{B} = \{0, 1\}$ .

► **Lemma 14.** Let  $\mathcal{V} = \{y_1, \dots, y_m\}$ . Given an FO formula  $\varphi$  with free variables contained in  $\mathcal{V}' = \mathcal{V} \cup \{x\}$ , we can construct a transducer  $\mathcal{B}_{\varphi, \mathcal{V}}$  from  $\Sigma_{\mathcal{V}}$  to  $\mathbb{B}$  which is aperiodic and unambiguous and such that for all words  $\bar{w} \in \Sigma_{\mathcal{V}}^+$

1. there is a (unique) accepting run of  $\mathcal{B}_{\varphi, \mathcal{V}}$  on the input word  $\bar{w}$  iff it is a valid encoding of a pair  $(w, \sigma)$  where  $w \in \Sigma^+$  and  $\sigma: \mathcal{V} \rightarrow \text{pos}(w)$  is a valuation,
2. and in this case, for all  $1 \leq i \leq |w|$ , the  $i$ th bit of the output is 1 iff  $w, \sigma[x \mapsto i] \models \varphi$ .

**Proof (sketch).** Notice that  $\Sigma_{\mathcal{V}'} = \Sigma_{\mathcal{V}} \times \mathbb{B}$  so letters in  $\Sigma_{\mathcal{V}'}$  are of the form  $(\bar{a}, 0)$  or  $(\bar{a}, 1)$  where  $\bar{a} \in \Sigma_{\mathcal{V}}$ . Abusing the notations, when  $\bar{v} \in \Sigma_{\mathcal{V}}^*$ , we write  $(\bar{v}, 0)$  to denote the word over  $\Sigma_{\mathcal{V}'}$  whose projection on  $\Sigma_{\mathcal{V}}$  is  $\bar{v}$  and projection on the  $x$ -component consists of 0's only.

Consider the deterministic, complete and aperiodic automaton  $\mathcal{A}_{\varphi, \mathcal{V}'} = (Q, \Sigma_{\mathcal{V}'}, \Delta, \iota, F, G)$  associated with  $\varphi$  by Theorem 3. We also denote by  $\Delta$  the extension of the transition function to subsets of  $Q$ . So we see the deterministic and complete transition relation both as a total function  $\Delta: Q \times \Sigma_{\mathcal{V}'} \rightarrow Q$  and  $\Delta: 2^Q \times \Sigma_{\mathcal{V}'} \rightarrow 2^Q$ .

We construct now the transducer  $\mathcal{B}_{\varphi, \mathcal{V}} = (Q', \Sigma_{\mathcal{V}}, \Delta', \text{wt}, I', F')$ . The set of states is  $Q' = Q \times 2^Q \times 2^Q \times \mathbb{B}$ . The unique initial state is  $\iota' = (\iota, \emptyset, \emptyset, 0)$ . The set of final states is  $F' = (Q \times 2^Q \times 2^Q \times \mathbb{B}) \setminus \{\iota'\}$ . Then, we define the following transitions:

- $\delta = ((p, X, Y, b), \bar{a}, (p', X', Y', 1)) \in \Delta'$  is a transition with weight  $\text{wt}(\delta) = 1$  if  $p' = \Delta(p, (\bar{a}, 0))$ ,  $X' = \Delta(X, (\bar{a}, 0)) \cup \{\Delta(p, (\bar{a}, 1))\}$  and  $Y' = \Delta(Y, (\bar{a}, 0))$ ,
- $\delta = ((p, X, Y, b), \bar{a}, (p', X', Y', 0)) \in \Delta'$  is a transition with weight  $\text{wt}(\delta) = 0$  if  $p' = \Delta(p, (\bar{a}, 0))$ ,  $X' = \Delta(X, (\bar{a}, 0))$  and  $Y' = \Delta(Y, (\bar{a}, 0)) \cup \{\Delta(p, (\bar{a}, 1))\}$ .

Notice that, whenever we read a new input letter  $\bar{a} \in \Sigma_{\mathcal{V}}$ , there is a non-deterministic choice. In the first case above, we guess that formula  $\varphi$  will hold on the input word when the valuation is extended by assigning  $x$  to the current position, whereas in the second case we guess that  $\varphi$  will not hold. The guess corresponds to the output of the transition, as required by the second condition of Lemma 14. Now, we have to check that the guess is correct. For this, the first component of  $\mathcal{B}_{\varphi, \mathcal{V}}$  computes the state  $p = \Delta(\iota, (\bar{u}, 0))$  reached by  $\mathcal{A}_{\varphi, \mathcal{V}}$  after reading  $(\bar{u}, 0)$  where  $\bar{u} \in \Sigma_{\mathcal{V}}^*$  is the current prefix of the input word. When reading the current letter  $\bar{a} \in \Sigma_{\mathcal{V}}$ , the transducer adds the state  $\Delta(p, (\bar{a}, 1)) = \Delta(\iota, (\bar{u}, 0)(\bar{a}, 1))$  either to the “positive”  $X$ -component or to the “negative”  $Y$ -component of its state, depending on its guess as explained above. Then, the transducer continues reading the suffix  $\bar{v} \in \Sigma_{\mathcal{V}}^*$  of the input word. It updates the  $X$  (resp.  $Y$ )-component so that it contains the state  $q = \Delta(\iota, (\bar{u}, 0)(\bar{a}, 1)(\bar{v}, 0))$  at the end of the run. Now, the acceptance condition allows us to check that the guess was correct.

1. If  $\bar{w} = \bar{u}\bar{a}\bar{v}$  is not a valid encoding of a pair  $(w, \sigma)$  with  $w \in \Sigma^+$  and  $\sigma: \mathcal{V} \rightarrow \text{pos}(w)$  then  $q \notin F \cup G$  and the run of the transducer is not accepting. Otherwise, let  $i \in \text{pos}(w)$  be the position where the guess was made.
2. If the guess was positive then  $q$  belongs to the  $X$ -component and the accepting condition implies  $q \in F$ , which means by definition of  $\mathcal{A}_{\varphi, \mathcal{V}}$  that  $w, \sigma[x \mapsto i] \models \varphi$ .
3. If the guess was negative then  $q$  belongs to the  $Y$ -component and the accepting condition implies  $q \in G$ , which means by definition of  $\mathcal{A}_{\varphi, \mathcal{V}}$  that  $w, \sigma[x \mapsto i] \not\models \varphi$ .  $\blacktriangleleft$

► **Theorem 15.** *Let  $\mathcal{V} = \{y_1, \dots, y_m\}$ . Given a step-wFO formula  $\Psi$  with free variables contained in  $\mathcal{V}' = \mathcal{V} \cup \{x\}$ , we can construct a weighted automaton  $\mathcal{A}_{\Psi, \mathcal{V}}$  over  $\Sigma_{\mathcal{V}}$  which is aperiodic, unambiguous and equivalent to  $\prod_x \Psi$ , i.e.,  $\{\mathcal{A}_{\Psi, \mathcal{V}}\}(\bar{w}) = \{\prod_x \Psi\}_{\mathcal{V}}(\bar{w})$  for all words  $\bar{w} \in \Sigma_{\mathcal{V}}^+$ .*

**Proof.** In case  $\Psi = r$  is an atomic step-wFO formula, we replace it with the equivalent  $\top ? r : r$  step-wFO formula. Let  $\varphi_1, \dots, \varphi_k$  be the FO formulas occurring in  $\Psi$ . By the above remark, we have  $k \geq 1$ . Consider the aperiodic and unambiguous transducers  $\mathcal{B}_1, \dots, \mathcal{B}_k$  given by Lemma 14. For  $1 \leq i \leq k$ , we let  $\mathcal{B}_i = (Q_i, \Sigma_{\mathcal{V}}, \Delta_i, \text{wt}_i, I_i, F_i)$ . The weighted automaton  $\mathcal{A}_{\Psi, \mathcal{V}} = (Q, \Sigma_{\mathcal{V}}, \Delta, \text{wt}, I, F)$  is essentially a cartesian product of the transducers  $\mathcal{B}_i$ . More precisely, we let  $Q = \prod_{i=1}^k Q_i$ ,  $I = \prod_{i=1}^k I_i$ ,  $F = \prod_{i=1}^k F_i$ , and

$$\Delta = \{((p_1, \dots, p_k), \bar{a}, (q_1, \dots, q_k)) \mid (p_i, \bar{a}, q_i) \in \Delta_i \text{ for all } 1 \leq i \leq k\}.$$

Since the transducers  $\mathcal{B}_i$  are all aperiodic and unambiguous, we deduce by Lemma 11 that  $\mathcal{A}_{\Psi, \mathcal{V}}$  is also aperiodic and unambiguous. It remains to define the weight function  $\text{wt}$ .

Given a bit vector  $\bar{b} = (b_1, \dots, b_k) \in \mathbb{B}^k$  of size  $k$ , we define  $\Psi(\bar{b})$  as the weight from  $\mathbb{R}$  resulting from the step-wFO formula  $\Psi$  when the FO conditions  $\varphi_1, \dots, \varphi_k$  evaluate to  $\bar{b}$ . Formally, the definition is by structural induction on the step-wFO formula:

$$r(\bar{b}) = r \qquad (\varphi_i ? \Psi_1 : \Psi_2)(\bar{b}) = \begin{cases} \Psi_1(\bar{b}) & \text{if } b_i = 1 \\ \Psi_2(\bar{b}) & \text{if } b_i = 0. \end{cases}$$

Consider a transition  $\delta = ((p_1, \dots, p_k), \bar{a}, (q_1, \dots, q_k)) \in \Delta$  and let  $\delta_i = (p_i, \bar{a}, q_i)$  for  $1 \leq i \leq k$ . Let  $\bar{b} = (b_1, \dots, b_k) \in \mathbb{B}^k$  where  $b_i = \text{wt}(\delta_i) \in \mathbb{B}$  for all  $1 \leq i \leq k$ . We define  $\text{wt}(\delta) = \Psi(\bar{b})$ .

Let  $\bar{w} \in \Sigma_{\mathcal{V}}^+$ . If  $\bar{w}$  is not a valid encoding of a pair  $(w, \sigma)$  then  $\{\prod_x \Psi\}_{\mathcal{V}}(\bar{w}) = \emptyset$  by definition. Moreover,  $\{\mathcal{A}_{\Psi, \mathcal{V}}\}(\bar{w}) = \emptyset$  since by Lemma 14,  $\bar{w}$  is not in the support of  $\mathcal{B}_1$ . We assume below that  $\bar{w}$  is a valid encoding of a pair  $(w, \sigma)$  where  $w \in \Sigma^+$  and  $\sigma: \mathcal{V} \rightarrow \text{pos}(w)$  is a valuation. Then, each transducer  $\mathcal{B}_i$  admits a unique accepting run  $\rho_i$  reading the input

word  $\bar{w}$ . These result in the unique accepting run  $\rho$  of  $\mathcal{A}_{\Psi, \mathcal{V}}$  reading  $\bar{w}$ . The projections of  $\rho$  on  $\mathcal{B}_1, \dots, \mathcal{B}_k$  are  $\rho_1, \dots, \rho_k$ . Let  $j \in \text{pos}(w) = \{1, \dots, |w|\}$  be a position in  $\bar{w}$  and let  $\delta^j$  be the  $j$ -th transition of  $\rho$ . For  $1 \leq i \leq k$ , we denote by  $\delta_i^j$  the projection of  $\delta^j$  on  $\mathcal{B}_i$  and we let  $b_i^j = \text{wt}(\delta_i^j)$ . By Lemma 14, we get  $b_i^j = 1$  iff  $w, \sigma[x \mapsto j] \models \varphi_i$ . Finally, let  $\bar{b}^j = (b_1^j, \dots, b_k^j)$ . From the above, we deduce that  $\llbracket \Psi \rrbracket_{\mathcal{V} \cup \{x\}}(w, \sigma[x \mapsto j]) = \Psi(\bar{b}^j) = \text{wt}(\delta^j)$ . Putting things together, we have  $\llbracket \mathcal{A}_{\Psi, \mathcal{V}} \rrbracket(w, \sigma) = \{\{\text{wt}(\rho)\}\} = \{\{\text{wt}(\delta^1) \cdots \text{wt}(\delta^{|w|})\}\} = \{\{\prod_x \Psi\}_{\mathcal{V}}(w, \sigma)\}$ . ◀

► **Theorem 16.** *Let  $\Phi$  be a wFO sentence. We can construct an aperiodic SCC-unambiguous weighted automaton  $\mathcal{A}$  such that  $\llbracket \mathcal{A} \rrbracket = \llbracket \Phi \rrbracket$ . Moreover, if  $\Phi$  does not contain the sum operations  $+$  and  $\sum_x$ , then  $\mathcal{A}$  can be chosen to be unambiguous. If  $\Phi$  does not contain the sum operation  $\sum_x$ , we can construct  $\mathcal{A}$  as a finite union of unambiguous weighted automata.*

**Proof.** We proceed by structural induction on  $\Phi$ . For  $\Phi = \mathbf{0}$  this is trivial. For  $\Phi = \prod_x \Psi$  with a step-wFO formula  $\Psi$ , we obtain an aperiodic unambiguous weighted automaton  $\mathcal{A}$  by Theorem 15. For formulas  $\varphi ? \Phi_1 : \Phi_2$ ,  $\Phi_1 + \Phi_2$  and  $\sum_x \Phi$ , we apply Lemmas 12, 11, 13. ◀

In the proof of Theorem 16, we may obtain the final statement also as a consequence of the preceding one by the following observations which could be of independent interest. Let  $\varphi$  be an FO-formula and  $\Phi_1, \Phi_2$  two wFO formulas, with free variables contained in  $\mathcal{V}$ . Then,

$$\begin{aligned} \llbracket \varphi ? \Phi_1 : \Phi_2 \rrbracket_{\mathcal{V}} &= \llbracket \varphi ? \Phi_1 : \mathbf{0} + \neg \varphi ? \Phi_2 : \mathbf{0} \rrbracket_{\mathcal{V}}, \\ \llbracket \varphi ? \Phi_1 + \Phi_2 : \mathbf{0} \rrbracket_{\mathcal{V}} &= \llbracket \varphi ? \Phi_1 : \mathbf{0} + \varphi ? \Phi_2 : \mathbf{0} \rrbracket_{\mathcal{V}}. \end{aligned}$$

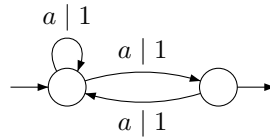
Hence, given a wFO sentence  $\Phi$  not containing the sum operation  $\sum_x$ , we can rewrite  $\Phi$  as a sum of  $\mathbf{0}$ ,  $\prod_x \Psi$  and if-then-else sentences of the form  $\varphi ? \Phi' : \mathbf{0}$  where  $\Phi'$  does not contain the sum operations  $+$  or  $\sum_x$ .

**Proof of Thm 1.** Immediate by Theorem 10, Theorem 4, Corollary 8 and Theorem 16. ◀

## 7 Concluding remarks

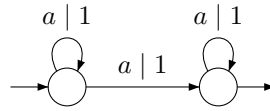
We introduced a model of aperiodic weighted automata and showed that a suitable concept of weighted first order logic and two natural sublogics have the same expressive power as polynomially ambiguous, finitely ambiguous, resp. unambiguous aperiodic weighted automata. For the three semirings  $\mathbb{N}_{+, \times}$ ,  $\mathbb{N}_{\max, +}$  and  $\mathbb{N}_{\min, +}$  the hierarchies of these automata classes and thereby of the corresponding logics are strict. Some separating examples are given below. Proofs and other separating examples can be found in [14].

► **Example 17.** Let  $\Sigma = \{a\}$  and consider the automaton  $\mathcal{A}$  below over the semiring  $\mathbb{N}_{+, \times}$  of natural numbers. Note that the weighted automaton computes the sequence  $(F_n)_{n \geq 0}$  of Fibonacci numbers  $0, 1, 1, 2, 3, 5, \dots$ . More precisely, for any  $n \in \mathbb{N}$ , we have  $\llbracket \mathcal{A} \rrbracket(a^n) = F_n$ .



Clearly,  $\mathcal{A}$  is exponentially ambiguous and aperiodic with index 2. But  $\llbracket \mathcal{A} \rrbracket$  cannot be realized by an aperiodic polynomially ambiguous weighted automaton. In [31], it was shown that the Fibonacci numbers cannot be computed by copyless cost-register automata. ◻

► **Example 18.** Consider the automaton  $\mathcal{A}$  below over  $\Sigma = \{a\}$  and the semiring  $\mathbb{N}_{+, \times}$ . Clearly,  $\llbracket \mathcal{A} \rrbracket(a^n) = n$  for each  $n > 0$ , and  $\mathcal{A}$  is aperiodic and polynomially (even linearly) ambiguous. But  $\mathcal{A}$  is not equivalent to any finitely ambiguous weighted automaton.



┘

► **Example 19.** Let  $\Sigma = \{a\}$  and consider the function  $f: \Sigma^* \rightarrow \mathbb{N}$  defined by  $f(a^n) = 2^n + 1$ . We have  $f = \llbracket \mathcal{A} \rrbracket$  for some aperiodic and 2-ambiguous weighted automaton: one self-loop computes  $2^n$  and another self-loop computes 1. But  $f$  cannot be realized by an unambiguous weighted automaton over  $\mathbb{N}_{+, \times}$ .

Our main theorem generalizes to the weighted setting a classical result of automata theory. A challenging open problem is to obtain similar results for suitable weighted linear temporal logics. Another interesting problem is to characterize wFO with unrestricted weighted products, possibly using *aperiodic* restrictions of the pebble weighted automata studied in [4, 27, 5].

Decidability problems for wFO or equivalently for weighted aperiodic automata are also open and very interesting. For instance, given a wMSO sentence, is there an equivalent wFO sentence? Decidability may indeed depend on the specific semiring.

---

## References

- 1 Parvaneh Babari, Manfred Droste, and Vitaly Perevoshchikov. Weighted register automata and weighted logic on data words. *Theor. Comput. Sci.*, 744:3–21, 2018.
- 2 Jean Berstel and Christophe Reutenauer. *Rational Series and their Languages*. Springer, 1988.
- 3 Benedikt Bollig and Paul Gastin. Weighted versus Probabilistic Logics. In Volker Diekert and Dirk Nowotka, editors, *International Conference on Developments in Language Theory (DLT'09)*, volume 5583 of *Lecture Notes in Computer Science*, pages 18–38. Springer, 2009.
- 4 Benedikt Bollig, Paul Gastin, Benjamin Monmege, and Marc Zeitoun. Pebble Weighted Automata and Transitive Closure Logics. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming*, pages 587–598, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
- 5 Benedikt Bollig, Paul Gastin, Benjamin Monmege, and Marc Zeitoun. Pebble Weighted Automata and Weighted Logics. *ACM Transactions on Computational Logic*, 15(2):1–35, 2014.
- 6 J. Richard Büchi. Weak Second-Order Arithmetic and Finite Automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:66–92, 1960.
- 7 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Expressiveness and Closure Properties for Quantitative Languages. *Logical Methods in Computer Science*, 6(3), 2010.
- 8 Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4):23:1–23:38, 2010.
- 9 Volker Diekert and Paul Gastin. First-order definable languages. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives*, volume 2 of *Texts in Logic and Games*, pages 261–306. Amsterdam University Press, 2008.
- 10 Manfred Droste and Stefan Dück. Weighted Automata and Logics on Graphs. In *Mathematical Foundations of Computer Science (MFCS'15)*, volume 9234 of *Lecture Notes in Computer Science*, pages 192–204. Springer, 2015.
- 11 Manfred Droste and Stefan Dück. Weighted automata and logics for infinite nested words. *Inf. Comput.*, 253:448–466, 2017.

- 12 Manfred Droste and Paul Gastin. Weighted Automata and Weighted Logics. In *International Colloquium on Automata, Languages and Programming (ICALP'05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 513–525. Springer, 2005.
- 13 Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theor. Comput. Sci.*, 380(1-2):69–86, 2007.
- 14 Manfred Droste and Paul Gastin. Aperiodic Weighted Automata and Weighted First-Order Logic. *CoRR*, abs/1902.08149, 2019.
- 15 Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. Springer Berlin Heidelberg, 2009.
- 16 Manfred Droste and Ingmar Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Inf. Comput.*, 220:44–59, 2012.
- 17 Manfred Droste and Vitaly Perevoshchikov. Multi-weighted Automata and MSO Logic. *Theory Comput. Syst.*, 59(2):231–261, 2016.
- 18 Manfred Droste and George Rahonis. Weighted automata and weighted logics on infinite words. *Izvestiya VUZ. Matematika*, 54:26–45, 2010.
- 19 Manfred Droste and Heiko Vogler. Weighted tree automata and weighted logics. *Theor. Comput. Sci.*, 366(3):228–247, 2006.
- 20 Manfred Droste and Heiko Vogler. Weighted automata and multi-valued logics over arbitrary bounded lattices. *Theor. Comput. Sci.*, 418:14–36, 2012.
- 21 Calvin C. Elgot. Decision Problems of Finite Automata Design and Related Arithmetics. *Transactions of the American Mathematical Society*, 98:21–52, 1961.
- 22 Ina Fichtner. Weighted Picture Automata and Weighted Logics. *Theory Comput. Syst.*, 48(1):48–78, 2011.
- 23 Paul Gastin and Benjamin Monmege. A unifying survey on weighted logics and weighted automata. *Soft Computing*, 22(4):1047–1065, December 2018.
- 24 Oscar H Ibarra and Bala Ravikumar. On sparseness, ambiguity and other decision problems for acceptors and transducers. In *Symposium on Theoretical Aspects of Computer Science (STACS'86)*, volume 210 of *Lecture Notes in Computer Science*, pages 171–179. Springer, 1986.
- 25 Daniel Kirsten. A Burnside Approach to the Termination of Mohri's Algorithm for Polynomially Ambiguous Min-Plus-Automata. *RAIRO - Theoretical Informatics and Applications*, 42(3):553–581, June 2008.
- 26 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, November 2004.
- 27 Stephan Kreutzer and Cristian Riveros. Quantitative Monadic Second-Order Logic. In *Symposium on Logic in Computer Science (LICS'13)*. IEEE, June 2013.
- 28 Werner Kuich and Arto Salomaa. *Semirings, Automata, Languages*. Springer Berlin Heidelberg, 1986.
- 29 Eleni Mandrali and George Rahonis. On weighted first-order logics with discounting. *Acta Informatica*, 51(2):61–106, January 2014.
- 30 Filip Mazowiecki and Cristian Riveros. Pumping Lemmas for Weighted Automata. In *Symposium on Theoretical Aspects of Computer Science (STACS'18)*, volume 96 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 50:1–50:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018.
- 31 Filip Mazowiecki and Cristian Riveros. Copyless cost-register automata: Structure, expressiveness, and closure properties. *Journal of Computer and System Sciences*, 100:1–29, March 2019.
- 32 Robert McNaughton and Seymour Papert. *Counter-Free Automata*. The MIT Press, Cambridge, Mass., 1971.
- 33 Erik Paul. On Finite and Polynomial Ambiguity of Weighted Tree Automata. In *International Conference on Developments in Language Theory (DLT'16)*, volume 9840 of *Lecture Notes in Computer Science*, pages 368–379. Springer, 2016.

- 34 Karin Quaas. MSO logics for weighted timed automata. *Formal Methods in System Design*, 38(3):193–222, 2011.
- 35 Christophe Reutenauer. *Propriétés arithmétiques et topologiques de séries rationnelles en variables non commutatives*. PhD thesis, Université Paris VI, 1977.
- 36 Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.
- 37 Jacques Sakarovitch and Rodrigo de Souza. Lexicographic Decomposition of k-Valued Transducers. *Theory of Computing Systems*, 47(3):758–785, April 2009.
- 38 Arto Salomaa and Matti Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Springer, 1978.
- 39 Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270, September 1961.
- 40 Marcel Paul Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.
- 41 Boris A. Trakhtenbrot. Finite Automata and Logic of Monadic Predicates. *Doklady Akademii Nauk SSSR*, 149:326–329, 1961.
- 42 Andreas Weber and Helmut Seidl. On the degree of ambiguity of finite automata. *Theoretical Computer Science*, 88(2):325–349, October 1991.