

# Finite-image property of weighted tree automata over past-finite monotonic strong bimonoids

Manfred Droste

*Institute of Computer Science,  
University of Leipzig, Leipzig, Germany*

Zoltán Fülöp\* and Dávid Kószó†

*Department of Foundations of Computer Science,  
University of Szeged, Szeged, Hungary*

Heiko Vogler

*Faculty of Computer Science,  
Technische Universität Dresden, Dresden, Germany*

July 1, 2021

## Abstract

We consider weighted tree automata over strong bimonoids (for short: wta). A wta  $\mathcal{A}$  has the finite-image property if its recognized weighted tree language  $\llbracket \mathcal{A} \rrbracket$  has finite image; moreover,  $\mathcal{A}$  has the preimage property if the preimage under  $\llbracket \mathcal{A} \rrbracket$  of each element of the underlying strong bimonoid is a recognizable tree language. For each wta  $\mathcal{A}$  over a past-finite monotonic strong bimonoid we prove the following results. In terms of  $\mathcal{A}$ 's structural properties, we characterize whether it has the finite-image property. We characterize those past-finite monotonic strong bimonoids such that for each wta  $\mathcal{A}$  it is decidable whether  $\mathcal{A}$  has the finite-image property. In particular, the finite-image property is decidable for wta over past-finite monotonic semirings. Moreover, we prove that  $\mathcal{A}$  has the preimage property. All our results also hold for weighted string automata.

---

\*Research of this author was supported by grant TUDFO/47138-1/2019-ITM of the Ministry for Innovation and Technology, Hungary.

†Supported by the ÚNKP-20-3 - New National Excellence Program of the Ministry for Innovation and Technology from the source of the National Research, Development and Innovation Fund.

# 1 Introduction

Weighted string automata (wsa) were invented [Sch61] for the purpose of describing quantitative properties of recognizable languages, like degree of ambiguity or costs of acceptance. Essentially each wsa is a nondeterministic finite-state automaton in which each transition carries a weight (quantity). In order to calculate with weights, an algebraic structure is needed, called *weight algebra*, and wsa have been investigated over several different weight algebras: semirings [Sch61, Eil74, BR88, KS86, Sak09, DKV09], lattices [Wec78, Rah09], strong bimonoids [DSV10, CDIV10, DV12], valuation monoids [DGMM11, DM12], and multi-cost valuation structures [DP16]. The two operations of these weight algebras, usually called addition and multiplication, are used to calculate the weight of a run on a given input word (by means of multiplication) and to sum up the weights of several runs on the given word (by means of addition). In this way, a wsa  $\mathcal{A}$  recognizes a weighted language  $\llbracket \mathcal{A} \rrbracket$  (or: formal power series), i.e., a mapping from the set of input words to the carrier set of the weight algebra. For the theory of wsa we refer to [Sch61, Eil74, SS78, Wec78, BR82, KS86, Kui97, Sak09, DKV09].

In a similar way, finite-state tree automata have been extended to weighted tree automata (wta) over various weight algebras, e.g., complete distributive lattices [IF75, ÉL07], fields [BR82], commutative semirings [AB87], strong bimonoids [Rad10, DFKV20], multioperator monoids [Kui99, FMV09, FSV12], and tree-valuation monoids [DHV15]. In any case, a wta  $\mathcal{A}$  recognizes a weighted tree language  $\llbracket \mathcal{A} \rrbracket$  (or: formal tree series), i.e., a mapping from the set of input trees to the carrier set of the weight algebra. We note that each wsa is a particular wta (cf. [FV09, p. 324]). For surveys we refer to [ÉK03, FV09].

Very important weight algebras for wsa and wta are (a) the semiring of natural numbers  $\mathbb{N}$ , (b) the max-plus-semiring on  $\mathbb{N}$ , (c) the min-plus-semiring on  $\mathbb{N}$ , (d) the semiring of finite formal languages [DK09, Sec. 2], and (e) the semiring of matrices over the positive integers. Apart from (c), these algebras are *past-finite* with a suitable order, i.e., each element has only finitely many predecessors in this order. Moreover, the addition and multiplication of each of these algebras are *monotone* with respect to this partial order. In [BFGM05] wta over monotonic semirings were investigated.

Justified by these important examples of weight algebras, we want to advocate in this paper the class of *past-finite monotonic strong bimonoids* as a general model for weight algebras. These weight algebras share many properties with the semiring of natural numbers: they have a partial order on its carrier set (which is not necessarily total), they are zero-sum free and zero-divisor free, their two operations are monotone, and a strong kind of well-foundedness (called past-finiteness) holds. However, in general, distributivity is not required. We will show that the natural numbers  $\mathbb{N}$  with addition provide a natural example for past-finite monotonic strong bimonoids which are not semirings.

We will generalize classical results from the theory of wsa and wta over the mentioned specific semirings in this more general setting of weight algebras in a uniform way. We note that classical results for wsa and their proofs crucially

employ matrices and therefore need the distributivity of the underlying weight algebras. Our development uses an analysis of the structure of the wta, but also algebraic means like congruences and, ultimately, it is combined with a reduction to the classical results for the semiring of natural numbers.

There are two natural questions associated with a wta  $\mathcal{A}$ :

- Does  $\mathcal{A}$  have the finite-image property? A wta  $\mathcal{A}$  has the *finite-image property* if the weighted tree language  $\llbracket \mathcal{A} \rrbracket$  has finite image.
- Does  $\mathcal{A}$  have the preimage property? A wta  $\mathcal{A}$  has the *preimage property* if the preimage under  $\llbracket \mathcal{A} \rrbracket$  of each element of the weight algebra is a recognizable tree language (cf., e.g., [Eng75, GS84] for the theory of recognizable tree languages).

In the literature there are some answers to these questions. Each wsa over a finite semiring and over the semiring of natural numbers has the preimage property, and each wsa over a commutative ring which has the finite-image property also has the preimage property [BR88, Ch. III]. This has also been shown for wta [DV06, LB99]. Furthermore, for each wsa over any subsemiring of the rational numbers, the finite-image property is decidable [MS77]. This latter property is related to the classical Burnside property for semigroups [RR85]. Moreover, each wta over a locally finite semiring [DV06] and each wsa over a bi-locally finite strong bimonoid [DSV10] has the finite-image property and the preimage property. Thus, in particular, each wsa over a bounded (not necessarily distributive) lattice has the two properties.

Weighted tree languages which are recognized by wta that have both, the finite-image property and the preimage property, are called *recognizable step mappings* [DG05, DV06]. The class of such mappings is characterized by crisp-deterministic wta [FKV21] (cf. [DSV10] for the string case). Intuitively, such a wta can be considered as a usual (unweighted) deterministic finite-state tree automaton in which each final state carries a weight.

In this paper, we investigate the two mentioned questions for wta over past-finite monotonic strong bimonoids. It is an extended version of [DFKV20] and our main results are the following:

- For each wta  $\mathcal{A}$  over some arbitrary strong bimonoid, we give a sufficient criterion such that  $\mathcal{A}$  has the finite-image property and the preimage property (cf. Theorem 6.6).
- Each wta  $\mathcal{A}$  over some past-finite monotonic strong bimonoid has the preimage property (cf. Theorem 6.10).
- For each wta  $\mathcal{A}$  over some past-finite monotonic strong bimonoid, we characterize when  $\mathcal{A}$  has the finite-image property, in terms of structural properties of  $\mathcal{A}$  (cf. Theorem 7.1).
- We characterize the subclass  $\mathcal{C}$  of those past-finite monotonic strong bimonoids for which the following holds: for each wta over some weight algebra from  $\mathcal{C}$ , it is decidable whether it has the finite-image property (cf. Theorem 7.12). In particular,  $\mathcal{C}$  contains all past-finite monotonic semirings (cf. Theorem 7.14).

- Given a wta  $\mathcal{A}$  over a past-finite monotonic strong bimonoid and some  $k \in \mathbb{N}_+$ , it is decidable whether the cardinality of the image of  $\llbracket \mathcal{A} \rrbracket$  is bounded by  $k$  (cf. Theorem 8.2).

All the above results except Item 2 are new, i.e., do not appear in [DFKV20]. For our decidability results, we assume that the respective weight algebras are given in a computable way.

Since wsa [Sch61, Eil74] over semirings are a special case of wta over semirings (cf. [FV09, p. 324]), and this relationship also holds for wsa over strong bimonoids, all our results also hold for wsa. We will explain this in more detail in Section 10.

## 2 Preliminaries

### 2.1 General notions and notations

We denote by  $\mathbb{N}$  the set of natural numbers  $\{0, 1, 2, \dots\}$  and by  $\mathbb{N}_+$  the set  $\mathbb{N} \setminus \{0\}$ . For every  $m, n \in \mathbb{N}$ , we denote the set  $\{i \in \mathbb{N} \mid m \leq i \leq n\}$  by  $[m, n]$ . We abbreviate  $[1, n]$  by  $[n]$ . Hence,  $[0] = \emptyset$ .

Let  $A$  be a set. Then  $|A|$  denotes the cardinality of  $A$ ,  $\mathcal{P}_f(A)$  denotes the set of finite subsets of  $A$ ,  $A^*$  denotes the *set of all strings over  $A$* , and  $\varepsilon$  denotes the empty string. For every  $v, w \in A^*$ ,  $vw$  denotes the *concatenation of  $v$  and  $w$* ,  $|v|$  denotes the length of  $v$ , and  $\text{prefix}(v)$  denotes the set  $\{w \in A^* \mid (\exists u \in A^*) : v = wu\}$ .

Let  $B$  be a set and  $R$  a binary relation on  $B$ . As usual, for every  $a, b \in B$ , we write  $aRb$  instead of  $(a, b) \in R$ . We call  $R$  an *equivalence relation* if it is reflexive, symmetric, and transitive. If  $R$  is an equivalence relation, then for each  $b \in B$  we denote by  $[b]_R$  the equivalence class  $\{a \in B \mid aRb\}$  and by  $B/R$  the set  $\{[b]_R \mid b \in B\}$ . We say that  $R$  is a *partial ordering* if it is reflexive, antisymmetric, and transitive. For each  $b \in B$ , let  $\text{past}(b) = \{a \in B \mid aRb\}$ . We call  $(B, R)$  *past-finite* if  $\text{past}(b)$  is finite for each  $b \in B$ .

Let  $f : A \rightarrow B$  and  $g : B \rightarrow C$  be mappings, where  $C$  is a further set. The *image of  $f$*  is the set  $\text{im}(f) = \{f(a) \mid a \in A\}$ , and for each  $b \in B$ , the *preimage of  $b$  under  $f$*  is the set  $f^{-1}(b) = \{a \in A \mid f(a) = b\}$ . Moreover, the *composition of  $f$  and  $g$*  is the mapping  $g \circ f : A \rightarrow C$  defined by  $(g \circ f)(a) = g(f(a))$  for each  $a \in A$ .

### 2.2 Trees and contexts

We suppose that the reader is familiar with the fundamental concepts and results of the theory of finite-state tree automata and recognizable tree languages [Eng75, GS84, CDG<sup>+</sup>08]. Here we only recall some basic definitions.

A *ranked alphabet* is a tuple  $(\Sigma, \text{rk})$  which consists of an alphabet  $\Sigma$  and mapping  $\text{rk} : \Sigma \rightarrow \mathbb{N}$ , called *rank mapping*, such that  $\text{rk}^{-1}(0) \neq \emptyset$ . For each  $k \in \mathbb{N}$ , we define  $\Sigma^{(k)} = \{\sigma \in \Sigma \mid \text{rk}(\sigma) = k\}$ . Sometimes we write  $\sigma^{(k)}$  to indicate that  $\sigma \in \Sigma^{(k)}$ . As usual, we abbreviate  $(\Sigma, \text{rk})$  by  $\Sigma$  if  $\text{rk}$  is irrelevant

or it is clear from the context. If  $\Sigma = \Sigma^{(1)} \cup \Sigma^{(0)}$  such that  $|\Sigma^{(1)}| \geq 1$  and  $|\Sigma^{(0)}| = 1$ , then we call  $\Sigma$  a *string ranked alphabet*.

Let  $\Sigma$  be a ranked alphabet and  $H$  be a set disjoint from  $\Sigma$ . The *set of  $\Sigma$ -trees over  $H$* , denoted by  $T_\Sigma(H)$ , is the smallest set  $T$  such that (i)  $\Sigma^{(0)} \cup H \subseteq T$  and (ii) if  $k \in \mathbb{N}_+$ ,  $\sigma \in \Sigma^{(k)}$ , and  $\xi_1, \dots, \xi_k \in T$ , then  $\sigma(\xi_1, \dots, \xi_k) \in T$ . We write  $T_\Sigma$  for  $T_\Sigma(\emptyset)$ . For every  $\gamma \in \Sigma^{(1)}$  and  $\alpha \in \Sigma^{(0)}$ , we abbreviate the tree  $\gamma(\dots\gamma(\alpha)\dots)$  with  $n$  occurrences of  $\gamma$  by  $\gamma^n(\alpha)$  and write  $\gamma$  for  $\gamma^1$ . Any subset  $L$  of  $T_\Sigma$  is called  *$\Sigma$ -tree language*.

We define the *set of positions* of trees as a mapping  $\text{pos} : T_\Sigma(H) \rightarrow \mathcal{P}_f(\mathbb{N}_+^*)$  such that (i) for each  $\xi \in (\Sigma^{(0)} \cup H)$  let  $\text{pos}(\xi) = \{\varepsilon\}$  and (ii) for every  $\xi = \sigma(\xi_1, \dots, \xi_k)$  with  $k \in \mathbb{N}_+$ ,  $\sigma \in \Sigma^{(k)}$ , and  $\xi_1, \dots, \xi_k \in T_\Sigma(H)$ , let  $\text{pos}(\xi) = \{\varepsilon\} \cup \{iv \mid i \in [k], v \in \text{pos}(\xi_i)\}$ . The *height* and the *size* of a tree  $\xi \in T_\Sigma$  are  $\text{height}(\xi) = \max\{|v| \mid v \in \text{pos}(\xi)\}$  and  $\text{size}(\xi) = |\text{pos}(\xi)|$ , respectively.

Let  $\xi, \zeta \in T_\Sigma(H)$  and  $v \in \text{pos}(\xi)$ . Then the *label of  $\xi$  at  $v$* , denoted by  $\xi(v)$ , the *subtree of  $\xi$  at  $v$* , denoted by  $\xi|_v$ , and the *replacement of the subtree of  $\xi$  at  $v$  by  $\zeta$* , denoted by  $\xi[\zeta]_v$ , are defined as follows:

- (i) if  $\xi \in (\Sigma^{(0)} \cup H)$ , then we let  $\xi(\varepsilon) = \xi$ ,  $\xi|_\varepsilon = \xi$ , and  $\xi[\zeta]_\varepsilon = \zeta$  and
- (ii) for every  $\xi = \sigma(\xi_1, \dots, \xi_k)$  with  $k \in \mathbb{N}_+$ ,  $\sigma \in \Sigma^{(k)}$ , and  $\xi_1, \dots, \xi_k \in T_\Sigma(H)$ , we define  $\xi(\varepsilon) = \sigma$  and  $\xi|_\varepsilon = \xi$ , and  $\xi[\zeta]_\varepsilon = \zeta$ , and for every  $i \in [k]$  and  $v' \in \text{pos}(\xi_i)$ , we define
  - $\xi(iv') = \xi_i(v')$ ,
  - $\xi|_{iv'} = \xi_i|_{v'}$ , and
  - $\xi[\zeta]_{iv} = \sigma(\xi_1, \dots, \xi_{i-1}, \xi_i[\zeta]_v, \xi_{i+1}, \dots, \xi_k)$ .

Let  $\square$  be a symbol such that  $\square \notin \Sigma$ . For each  $\zeta \in T_\Sigma(\{\square\})$ , we define  $\text{pos}_\square(\zeta) = \{v \in \text{pos}(\zeta) \mid \zeta(v) = \square\}$ , and for each  $v \in \text{pos}(\zeta)$  we abbreviate by  $\zeta|_v$  the tree  $\zeta[\square]_v$ . We denote by  $C_\Sigma$  the set  $\{\zeta \in T_\Sigma(\{\square\}) \mid |\text{pos}_\square(\zeta)| = 1\}$ , and we call its elements *contexts over  $\Sigma$*  (for short:  *$\Sigma$ -contexts* or *contexts*). Thus a context is a tree over the ranked alphabet  $\Sigma$  and the set  $H = \{\square\}$  in which  $\square$  occurs precisely once, as a leaf.

Let  $c \in C_\Sigma$  with  $\{v\} = \text{pos}_\square(c)$  and  $\zeta \in (T_\Sigma \cup C_\Sigma)$ . Then we abbreviate  $c[\zeta]_v$  by  $c[\zeta]$ . Hence  $c[\zeta]$  is obtained from the context  $c$  by replacing the leaf  $\square$  by  $\zeta$ . Obviously, if  $\zeta \in C_\Sigma$ , then also  $c[\zeta] \in C_\Sigma$ . Moreover, for each  $n \in \mathbb{N}$ , we define the  *$n$ th power of  $c$* , denoted by  $c^n$ , by induction as follows:  $c^0 = \square$  and  $c^{n+1} = c[c^n]$ .

*In the rest of this paper,  $\Sigma$  will denote an arbitrary ranked alphabet if not specified otherwise. Moreover, if we write ' $\xi = \sigma(\xi_1, \dots, \xi_k)$ ', then we mean that there exist  $k \in \mathbb{N}_+$ ,  $\sigma \in \Sigma^{(k)}$ , and  $\xi_1, \dots, \xi_k \in T_\Sigma(H)$  such that  $\xi = \sigma(\xi_1, \dots, \xi_k)$ .*

### 2.3 Strong bimonoids

A *strong bimonoid* [DSV10, CDIV10, DV12] is an algebra  $(B, \oplus, \otimes, \mathbf{0}, \mathbf{1})$  such that  $(B, \oplus, \mathbf{0})$  is a commutative monoid,  $(B, \otimes, \mathbf{1})$  is a monoid,  $\mathbf{0} \neq \mathbf{1}$ , and  $\mathbf{0} \otimes b = b \otimes \mathbf{0} = \mathbf{0}$  for each  $b \in B$ .

We say that  $B$  is

- *commutative* if  $\otimes$  is commutative,
- *left distributive* (respectively, *right distributive*) if  $\otimes$  is distributive over  $\oplus$  from the left (respectively, the right), and
- a *semiring* if it is left and right distributive.

Moreover, we call  $B$

- *one-product free* if  $a \otimes b = \mathbb{1}$  implies  $a = \mathbb{1} = b$ ,
- *zero-divisor free* if  $a \otimes b = \mathbb{0}$  implies  $a = \mathbb{0}$  or  $b = \mathbb{0}$ ,
- *zero-sum free* if  $a \oplus b = \mathbb{0}$  implies  $a = \mathbb{0}$  and  $b = \mathbb{0}$ , and
- *(additively) idempotent* if  $a \oplus a = a$

for every  $a, b \in B$ .

In [BFGM05, Def. 12] the concept of *monotonic semiring* is defined. In the spirit of this definition, we define monotonic strong bimonoids as follows.

**Definition 2.1.** Let  $(B, \oplus, \otimes, \mathbb{0}, \mathbb{1})$  be a strong bimonoid and  $\preceq$  a partial order on  $B$ . We write  $a \prec b$  to denote that  $a \preceq b$  and  $a \neq b$ . We say that  $(B, \oplus, \otimes, \mathbb{0}, \mathbb{1}, \preceq)$  is *monotonic* if the following conditions hold:

- (i) for every  $a, b \in B : a \preceq a \oplus b$ , and
- (ii) for every  $a, b, c \in B \setminus \{\mathbb{0}\}$  with  $b \neq \mathbb{1}$  we have:  $a \otimes c \prec a \otimes b \otimes c$ .

We call  $(B, \oplus, \otimes, \mathbb{0}, \mathbb{1}, \preceq)$  *past-finite* if  $(B, \preceq)$  is past-finite.

If  $B$  is monotonic, then, as is easy to check,  $\mathbb{0} \prec \mathbb{1} \prec b$  for each  $b \in B \setminus \{\mathbb{0}, \mathbb{1}\}$ ; hence, clearly  $B$  is one-product free, zero-divisor free and zero-sum free. Moreover, if  $B$  has at least 3 elements, then  $B$  is infinite. The only monotonic strong bimonoid with 2 elements is the Boolean semiring  $(\mathbb{B}, \sup, \inf, \mathbb{0}, \mathbb{1})$  with its natural order, where  $\mathbb{B} = \{0, 1\}$ . Cf. [BFGM05, p. 122] for further results on monotonic semirings.

**Example 2.2.** We give six examples of past-finite monotonic semirings (cf. [BFGM05, p. 122]):

- (i) the semiring of natural numbers  $(\mathbb{N}, +, \cdot, \mathbb{0}, \mathbb{1}, \leq)$ ;
- (ii) the arctic semiring  $\text{ASR}_{\mathbb{N}} = (\mathbb{N}_{-\infty}, \max, +, -\infty, \mathbb{0}, \leq)$ , where  $\mathbb{N}_{-\infty} = \mathbb{N} \cup \{-\infty\}$ ;
- (iii) the semiring  $\text{Lcm} = (\mathbb{N}, \text{lcm}, \cdot, \mathbb{0}, \mathbb{1}, \leq)$ , where  $\text{lcm}(0, n) = n = \text{lcm}(n, 0)$  for each  $n \in \mathbb{N}$  and otherwise  $\text{lcm}$  is the usual least common multiple;
- (iv) the semiring  $\text{FSet}(\mathbb{N}) = (\mathcal{P}_f(\mathbb{N}), \cup, +, \emptyset, \{0\}, \preceq)$  where the addition on  $\mathbb{N}$  is extended to sets as usual, and  $\preceq$  is defined by  $N_1 \preceq N_2$  if there is an injective mapping  $f : N_1 \rightarrow N_2$  such that  $n \leq f(n)$  for each  $n \in N_1$ ;
- (v) for each  $n \in \mathbb{N}_+$ , the semiring  $\text{Mat}_n(\mathbb{N}_+) = (\mathbb{N}_+^{n \times n} \cup \{\underline{\mathbb{0}}, \underline{\mathbb{1}}\}, +, \cdot, \underline{\mathbb{0}}, \underline{\mathbb{1}}, \leq)$  of square matrices over  $\mathbb{N}_+$  with the common matrix addition and multiplication, where  $\underline{\mathbb{0}}$  is the  $n \times n$  zero matrix and  $\underline{\mathbb{1}}$  is the  $n \times n$  unit matrix; the partial order  $\leq$  is defined by  $M \leq M'$  if  $M_{ij} \leq M'_{ij}$  for each  $(i, j) \in [n] \times [n]$ ; and
- (vi) the semiring  $\text{FLang}_{\Sigma} = (\mathcal{P}_f(\Sigma^*), \cup, \cdot, \emptyset, \{\varepsilon\}, \preceq)$  over the alphabet  $\Sigma$  with the operations of union and concatenation, and  $\preceq$  is defined by  $L_1 \preceq L_2$  if there is an injective mapping  $f : L_1 \rightarrow L_2$  such that  $w$  is a subword of  $f(w)$  for each  $w \in L_1$ .

The semirings in (i)-(iv) are commutative, and the semirings in (ii)-(iv) and (vi) are idempotent.  $\square$

Next we wish to give a natural example of a past-finite monotonic strong bimonoid which is not a semiring.

**Example 2.3.** Take  $(\mathbb{N}_0, \oplus, +, 0, 0)$ , the natural numbers with plus and plus, with a new zero  $0$  added. That is, we have  $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$ , the bimonoid addition on  $\mathbb{N}$  is the usual one, denoted by  $\oplus$ , the bimonoid multiplication on  $\mathbb{N}$  is also the usual addition, denoted by  $+$ , in order to indicate that here the usual addition serves as bimonoid multiplication. Moreover,  $0 \oplus x = x$  and  $0 + x = 0$  for each  $x \in \mathbb{N}_0$ . Let  $\leq$  be the usual order on  $\mathbb{N}$  together with  $0 < x$  for each  $x \in \mathbb{N}$ . Then  $(\mathbb{N}_0, \oplus, +, 0, 0, \leq)$  is a past-finite strong bimonoid which is not a semiring. We might call this structure the *plus-plus-strong bimonoid of natural numbers*.

Now we give an example of an additively locally finite and past-finite monotonic strong bimonoid which is not a semiring.

**Example 2.4.** The strong bimonoid  $(\mathbb{N}, +', \cdot, 0, 1, \leq)$  with the operation  $+'$  defined, for each  $a, b \in \mathbb{N}$ , by

$$a +' b = \begin{cases} \min\{a + b, 100\} & \text{if } a, b \leq 100 \\ \max\{a, b\} & \text{otherwise} \end{cases}$$

is additively locally finite and past-finite monotonic. Moreover, it is neither left distributive nor right distributive.  $\square$

Next we show a general method for generating past-finite monotonic strong bimonoids.

**Example 2.5.** Let  $(B, \preceq)$  be a past-finite partially ordered set. Let  $(B, +)$  be a commutative semigroup such that, for every  $a, b \in B$ , we have  $a \preceq a + b$ . Moreover, let  $(B, \times)$  be a semigroup such that, for every  $a, b, c \in B$ , we have  $a \prec a \times b$ ,  $c \prec b \times c$ , and  $a \times c \prec a \times b \times c$ .

According to [DV12, Ex. 2.1(4)], we construct the *strong bimonoid induced by*  $(B, +)$  and  $(B, \times)$  to be the strong bimonoid  $(B', \oplus, \otimes, 0, 1)$  defined as follows:

- $B' = B \cup \{0, 1\}$  where  $0, 1 \notin B$ ;
- we define the operation  $\oplus : B' \times B' \rightarrow B'$  such that  $\oplus|_{B \times B} = +$  and for each  $b \in B'$  we let  $0 \oplus b = b = b \oplus 0$  and, if  $b \neq 0$ , then  $1 \oplus b = b = b \oplus 1$ ;
- we define the operation  $\otimes : B' \times B' \rightarrow B'$  such that  $\otimes|_{B \times B} = \times$  and for each  $b \in B'$  we let  $0 \otimes b = 0 = b \otimes 0$ , and  $1 \otimes b = b = b \otimes 1$ .

We define the partial ordering  $\preceq'$  on  $B'$  such that  $0 \prec' 1 \prec' b$  for each  $b \in B$  and  $\preceq' \cap (B \times B) = \preceq$ . Then  $(B', \oplus, \otimes, 0, 1, \preceq')$  is past-finite monotonic.

To verify this, we make the following observations. Clearly,  $(B', \preceq')$  is past-finite. By case analysis, it is easy to show that  $\oplus$  and  $\otimes$  satisfy properties (i) and (ii) of the definition of monotonic strong bimonoid, respectively. In particular, property (ii) can be seen as follows. Let  $a, b, c \in B' \setminus \{0\}$  and  $b \neq 1$  (i.e.,  $b \in B$ ): We claim that  $a \otimes c \prec' a \otimes b \otimes c$ .

If  $a = c = \mathbb{1}$ , then  $a \otimes c = \mathbb{1} \prec' b = a \otimes b \otimes c$ .

If  $a \neq \mathbb{1}$  and  $c = \mathbb{1}$ , then we have  $a \in B$  and  $a \prec a \times b$  by the assumption on  $(B, \times)$ , and hence  $a \prec' a \times b$  by the definition of  $\prec'$ . Then  $a \otimes c = a \prec' a \times b = a \otimes b = a \otimes b \otimes c$ .

If  $a = \mathbb{1}$  and  $c \neq \mathbb{1}$ , then  $c \in B$  and  $c \prec b \times c$  by the assumption on  $(B, \times)$ . Using the same arguments as in the previous case, we have  $a \otimes c = c \prec' b \times c = b \otimes c = a \otimes b \otimes c$ .

Finally, if  $a \neq \mathbb{1}$  and  $b \neq \mathbb{1}$ , then we have  $a, b, c \in B$  and therefore  $a \times c \prec a \times b \times c$  by the assumption on  $(B, \times)$ . Since  $\otimes$  on  $B$  and  $\prec'$  on  $B$  are equal to  $\times$  and  $\prec$ , respectively, we obtain  $a \otimes c \prec' a \otimes b \otimes c$ .

Clearly,  $(B', \oplus, \otimes, \mathbb{0}, \mathbb{1})$  is additively locally finite if, for each finite  $A \subseteq B$ , the subsemigroup of  $(B, +)$  generated by  $A$  is finite.  $\square$

As an application of the general method, we let  $(B, \preceq) = (\mathbb{N}_+, \leq)$ ,  $(B, +) = (\mathbb{N}_+, +)$  and  $(B, \times) = (\mathbb{N}_+, +)$ , both with the usual addition of natural numbers, and  $\mathbb{1} = 0$ . Then we obtain the plus-plus-strong bimonoid of natural numbers given in Example 2.3. As another application, we can choose  $(B, \preceq) = (\mathbb{N}_+, \leq)$ ,  $(B, +) = (\mathbb{N}_+, +)$  with the usual addition on natural numbers, and  $(B, \times) = (\mathbb{N}_+, \times)$  where  $a \times b = a + b + 2ab$  for every  $a, b \in \mathbb{N}_+$ . Then  $B'$  is a past-finite monotonic strong bimonoid. Moreover,  $B'$  is neither left distributive nor right distributive. As a third one, we can consider the semigroup  $(B, +) = (\mathbb{N}_+, \max)$  and the rest as above. Then  $B'$  is an idempotent and past-finite monotonic semiring.

For later use, we introduce the following notations and notions.

We extend  $\oplus$  to every finite set  $I$  and family  $(b_i \mid i \in I)$  of elements of  $B$  as usual, and denote the extended operation by  $\bigoplus$ . We always abbreviate  $\bigoplus(b_i \mid i \in I)$  by  $\bigoplus_{i \in I} b_i$ . Moreover, if  $I = [k]$  for some  $k \in \mathbb{N}$ , then we write  $\bigoplus_{i=1}^k b_i$ . For each  $b \in B$ , we abbreviate  $\bigoplus_{i=1}^k b$  by  $kb$ .

We also extend the operation  $\otimes$  to every  $k \in \mathbb{N}$  and family  $(b_i \mid i \in [k])$  of elements of  $B$  as usual, and denote the extended operation by  $\bigotimes$ . We always abbreviate  $\bigotimes(b_i \mid i \in [k])$  by  $\bigotimes_{i=1}^k b_i$ . For each  $b \in B$ , we abbreviate  $\bigotimes_{i=1}^k b$  by  $b^k$ . For each  $B_1, B_2 \subseteq B$ , we denote by  $B_1 \otimes B_2$  the set  $\{a \otimes b \mid a \in B_1, b \in B_2\}$ .

For each  $A \subseteq B$ , we denote by  $\langle A \rangle_{\oplus}$  the *submonoid of  $(B, \oplus, \mathbb{0})$  generated by  $A$*  and we denote by  $\langle A \rangle_{\otimes}$  the *submonoid of  $(B, \otimes, \mathbb{1})$  generated by  $A$* . We say that  $B$  is *additively locally finite* (respectively, *multiplicatively locally finite*) if, for each finite  $A \subseteq B$ , the submonoid  $\langle A \rangle_{\oplus}$  (respectively,  $\langle A \rangle_{\otimes}$ ) is finite. Observe that, if  $B$  is idempotent, then it is also additively locally finite. We say that  $B$  is *bi-locally finite* if it is both additively locally finite and multiplicatively locally finite. For each  $b \in B$ , we abbreviate  $\langle \{b\} \rangle_{\oplus}$  by  $\langle b \rangle_{\oplus}$  and if  $\langle b \rangle_{\oplus}$  is finite, then we say that  $b$  has *finite additive order*.

*In the rest of the paper,  $(B, \oplus, \otimes, \mathbb{0}, \mathbb{1})$  denotes an arbitrary strong bimonoid if not specified otherwise.*



## 2.4 Weighted tree languages

A  $(\Sigma, B)$ -weighted tree language (for short: weighted tree language) is a mapping  $r : T_\Sigma \rightarrow B$ . For every  $(\Sigma, B)$ -weighted tree language  $r$ , we denote by  $\text{supp}(r)$  the set  $\{\xi \in T_\Sigma \mid r(\xi) \neq 0\}$ .

Let  $L \subseteq T_\Sigma$  be a  $\Sigma$ -tree language. The *characteristic mapping of  $L$  with respect to  $B$*  is the mapping  $\mathbb{1}_{(B,L)} : T_\Sigma \rightarrow B$  defined, for each  $\xi \in T_\Sigma$ , by  $\mathbb{1}_{(B,L)}(\xi) = \mathbb{1}$  if  $\xi \in L$  and 0 otherwise.

Let  $r$  and  $r'$  be  $(\Sigma, B)$ -weighted tree languages and  $b \in B$ . We define the  $(\Sigma, B)$ -weighted tree languages  $r \oplus r'$  and  $b \otimes r$ , for each  $\xi \in T_\Sigma$ , by  $(r \oplus r')(\xi) = r(\xi) \oplus r'(\xi)$  and  $(b \otimes r)(\xi) = b \otimes r(\xi)$ , respectively.

We say that  $r$  is a  $(\Sigma, B)$ -recognizable one-step mapping (or just: recognizable one-step mapping) if there exist a recognizable  $\Sigma$ -tree language  $L \subseteq T_\Sigma$  and a  $b \in B$  such that  $r = b \otimes \mathbb{1}_{(B,L)}$ . The tree language  $L$  is called *step language*. Moreover,  $r$  is a  $(\Sigma, B)$ -recognizable step mapping (or just: recognizable step mapping) if there exist  $n \in \mathbb{N}_+$  and  $(\Sigma, B)$ -recognizable one-step mappings  $r_1, \dots, r_n$  such that  $r = \bigoplus_{i=1}^n r_i$  (where we extend the sum of two weighted tree languages in a straightforward way to the sum of finitely many weighted tree languages). Obviously, if  $r$  is a recognizable step-mapping, then  $\text{im}(r)$  is finite. We note that, in [Bor04], recognizable one-step mappings were called weighted tree languages which are constant on their supports.

For each  $b \in B$ , we define the weighted tree language  $\tilde{b}$  by  $\tilde{b}(\xi) = b$  for each  $\xi \in T_\Sigma$ . We note that  $\tilde{b}$  is the recognizable one-step mapping  $b \otimes \mathbb{1}_{(B,T_\Sigma)}$ .

## 3 Weighted tree automata with run semantics

We recall the concept of weighted tree automata over strong bimonoids from [Rad10] (also cf., e.g., [FV09, FKV21]). A *weighted tree automaton (over  $\Sigma$  and  $B$ )* (for short:  $(\Sigma, B)$ -wta or wta) is a tuple  $\mathcal{A} = (Q, \delta, F)$ , where  $Q$  is a finite nonempty set (*states*),  $\delta = (\delta_k \mid k \in \mathbb{N})$  is a family of mappings  $\delta_k : Q^k \times \Sigma^{(k)} \times Q \rightarrow B$  (*transition mappings*), and  $F : Q \rightarrow B$  (*root weight mapping*).

*From now on, for every  $k \in \mathbb{N}$ ,  $(q_1, \dots, q_k) \in Q^k$ ,  $\sigma \in \Sigma^{(k)}$ , and  $q \in Q$ , we abbreviate expressions of the form  $\delta_k((q_1, \dots, q_k), \sigma, q)$  by  $\delta_k(q_1 \cdots q_k, \sigma, q)$ . Moreover, we write  $F_q$  instead of  $F(q)$  for each  $q \in Q$ .*

We say that  $\mathcal{A}$  is *deterministic* (and *crisp-deterministic*) if, for every  $k \in \mathbb{N}$ ,  $w \in Q^k$ , and  $\sigma \in \Sigma^{(k)}$  there exists at most one  $q \in Q$  such that  $\delta_k(w, \sigma, q) \neq 0$  (respectively, there exists a  $q \in Q$  such that  $\delta_k(w, \sigma, q) = \mathbb{1}$ , and  $\delta_k(w, \sigma, q') = 0$  for each  $q' \in Q \setminus \{q\}$ ). Clearly, crisp-determinism implies determinism.

We define the run semantics for a  $(\Sigma, B)$ -wta as follows. Let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta,  $\zeta \in T_\Sigma(\{\square\})$ , and  $\rho : \text{pos}(\zeta) \rightarrow Q$ . We call  $\rho$  a *run of  $\mathcal{A}$  on  $\zeta$*  if, for every  $v \in \text{pos}(\zeta)$  with  $\zeta(v) \in \Sigma$ , we have  $\delta_k(\rho(v1) \cdots \rho(vk), \sigma, \rho(v)) \neq 0$  where  $\sigma = \zeta(v)$  and  $k = \text{rk}(\sigma)$ . If  $\rho(\varepsilon) = q$  for some  $q \in Q$ , then we say that  $\rho$  is a  *$q$ -run on  $\zeta$* . We denote by  $R_{\mathcal{A}}(q, \zeta)$  the set of all  $q$ -runs on  $\zeta$  and we let

$R_{\mathcal{A}}(\zeta) = \bigcup_{q \in Q} R_{\mathcal{A}}(q, \zeta)$ . If  $\mathcal{A}$  is deterministic, then  $|R_{\mathcal{A}}(\zeta)| \leq 1$ . Moreover, we let  $R_{\mathcal{A}}^{F \neq 0}(\zeta)$  denote the set of all  $\rho \in R_{\mathcal{A}}(\zeta)$  such that  $F_{\rho(\varepsilon)} \neq 0$ . In particular, for  $c \in C_{\Sigma}$  with  $\text{pos}_{\square}(c) = \{v\}$ , we call each  $\rho \in R_{\mathcal{A}}(q, c)$  a  $(q, \rho(v))$ -run on  $c$  and we denote the set of all  $(q, p)$ -runs on  $c$  by  $R_{\mathcal{A}}(q, c, p)$ . We note that  $R_{\mathcal{A}}(q, c) = \bigcup_{p \in Q} R_{\mathcal{A}}(q, c, p)$ . Each element of  $R_{\mathcal{A}}(q, c, q)$  is called *loop*.

Let  $\zeta \in T_{\Sigma}(\{\square\})$ ,  $\rho \in R_{\mathcal{A}}(\zeta)$ , and  $v \in \text{pos}(\zeta)$ . We define the mapping  $\rho|_v : \text{pos}(\zeta|_v) \rightarrow Q$  such that, for each  $v' \in \text{pos}(\zeta|_v)$ , we have  $\rho|_v(v') = \rho(vv')$ . Clearly,  $\rho|_v \in R_{\mathcal{A}}(\zeta|_v)$ , and hence we call it the *run induced by  $\rho$  at  $v$* .

We say that  $\mathcal{A}$  is *finitely ambiguous* if there exists  $K \in \mathbb{N}$  such that, for each  $\xi \in T_{\Sigma}$ , we have  $|R_{\mathcal{A}}^{F \neq 0}(\xi)| \leq K$ . Moreover, we call  $\mathcal{A}$  *unambiguous* if it is finitely ambiguous and  $K = 1$ . We note that each deterministic wta is unambiguous, and there exist easy examples of unambiguous wta for which there does not exist an equivalent deterministic wta [KLMP04].

Now we define the *weight of a run*  $\rho \in R_{\mathcal{A}}(\zeta)$  to be the element  $\text{wt}_{\mathcal{A}}(\zeta, \rho)$  of  $B$  by induction as follows: (i) if  $\zeta = \square$ , then  $\text{wt}_{\mathcal{A}}(\zeta, \rho) = \mathbb{1}$  and (ii) if  $\zeta = \sigma(\zeta_1, \dots, \zeta_k)$  then  $\text{wt}_{\mathcal{A}}(\zeta, \rho)$  is defined by

$$\text{wt}_{\mathcal{A}}(\zeta, \rho) = \left( \bigotimes_{i=1}^k \text{wt}_{\mathcal{A}}(\zeta_i, \rho|_i) \right) \otimes \delta_k(\rho(1) \cdots \rho(k), \sigma, \rho(\varepsilon)) . \quad (1)$$

If there is no confusion, then we drop the index  $\mathcal{A}$  from  $\text{wt}_{\mathcal{A}}$  and write just  $\text{wt}(\zeta, \rho)$  for the weight of  $\rho$ .

The *run semantics of  $\mathcal{A}$*  is the  $(\Sigma, B)$ -weighted tree language  $\llbracket \mathcal{A} \rrbracket : T_{\Sigma} \rightarrow B$  defined, for each  $\xi \in T_{\Sigma}$ , by

$$\llbracket \mathcal{A} \rrbracket(\xi) = \bigoplus_{\rho \in R_{\mathcal{A}}(\xi)} \text{wt}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = \bigoplus_{\rho \in R_{\mathcal{A}}^{F \neq 0}(\xi)} \text{wt}(\xi) \otimes F_{\rho(\varepsilon)} ,$$

where the second equality holds because  $\text{wt}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = \mathbb{0}$  for each  $\rho \in R_{\mathcal{A}}(\xi) \setminus R_{\mathcal{A}}^{F \neq 0}(\xi)$ . We will use the above equality without any reference. Let  $\mathcal{A}$  and  $\mathcal{B}$  be  $(\Sigma, B)$ -wta. We say that  $\mathcal{A}$  and  $\mathcal{B}$  are *equivalent* if  $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$ . A weighted tree language  $r : T_{\Sigma} \rightarrow B$  is *run-recognizable* if there exists a  $(\Sigma, B)$ -wta  $\mathcal{A}$  such that  $r = \llbracket \mathcal{A} \rrbracket$ . The class of all run-recognizable  $(\Sigma, B)$ -weighted tree languages is denoted by  $\text{Rec}(\Sigma, B)$ .

**Example 3.1.** For the weight structure of Example 2.3, the plus-plus-strong bimonoid of natural numbers, the weighted tree automaton along a run would sum up all weights (costs) of the transitions occurring in the run, but to determine the weight of a tree it would also execute all possible runs and sum up their weights (costs). This might be considered as the total sum of the weights of all transitions of all non-deterministic realizations (runs).

The following fact is well known and we will use it in the paper without any further reference (cf. e.g. [FV09, Sect. 3.4]). A  $\Sigma$ -tree language  $L \subseteq T_{\Sigma}$  is *recognizable*, i.e., recognizable by a finite-state  $\Sigma$ -tree automaton if and only if there exists a  $(\Sigma, \mathbb{B})$ -wta  $\mathcal{A}$  such that  $L = \text{supp}(\llbracket \mathcal{A} \rrbracket)$  (recall that  $\mathbb{B}$  is the

Boolean semiring). Moreover, for each  $(\Sigma, \mathbb{B})$ -wta  $\mathcal{A}$ , we can construct a finite-state  $\Sigma$ -tree automaton which recognizes  $\text{supp}(\llbracket \mathcal{A} \rrbracket)$ . Vice versa, for each finite-state  $\Sigma$ -tree automaton which recognizes  $L$ , we can construct a  $(\Sigma, \mathbb{B})$ -wta  $\mathcal{A}$  such that  $L = \text{supp}(\llbracket \mathcal{A} \rrbracket)$ . *Therefore, in order to avoid using several automata models, we identify finite state  $\Sigma$ -tree automata with  $(\Sigma, \mathbb{B})$ -wta.*

We note that also another semantics, called *initial algebra semantics*, can be defined for  $\mathcal{A}$  [Rad10, FV09, FKV21]. In general, the two kinds of semantics are different [DSV10], however, if  $B$  is a semiring or  $\mathcal{A}$  is deterministic, then they coincide [Bor05, Lm. 4.1.13], [Rad10, Thm. 4.1], and [FKV21, Thm. 3.10].

**Example 3.2.** Let  $\Sigma = \{\gamma^{(1)}, \alpha^{(0)}\}$ . We consider the  $(\Sigma, \text{ASR}_{\mathbb{N}})$ -wta  $\mathcal{A} = (\{q\}, \delta, F)$  with  $\delta_0(\varepsilon, \alpha, q) = F_q = 0$  and  $\delta_1(q, \gamma, q) = 1$ . Clearly,  $\mathcal{A}$  is deterministic and not crisp-deterministic (because 1 is not one of the unit elements of  $\text{ASR}_{\mathbb{N}}$ ). Moreover,  $\llbracket \mathcal{A} \rrbracket(\gamma^n(\alpha)) = n$  for each  $n \in \mathbb{N}$ . Hence,  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite.  $\square$

Next we recall three results which we will need in this paper. The first result is a straightforward generalization of [BMS<sup>+</sup>06, Lm. 3] from semirings to strong bimonoids, cf. also [FV09, Thm. 3.9].

**Lemma 3.3.** Let  $B$  and  $C$  be strong bimonoids,  $\mathcal{A}$  be  $(\Sigma, B)$ -wta, and  $h : B \rightarrow C$  a strong bimonoid homomorphism. We can construct a  $(\Sigma, C)$ -wta  $h(\mathcal{A})$  such that  $\llbracket h(\mathcal{A}) \rrbracket = h \circ \llbracket \mathcal{A} \rrbracket$ .

*Proof.* Let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta. We introduce the  $(\Sigma, C)$ -wta  $h(\mathcal{A}) = (Q, \delta', F')$  by defining  $\delta' = (\delta'_k \mid k \in \mathbb{N})$  with  $\delta'_k = h \circ \delta_k$  for each  $k \in \mathbb{N}$  and  $F' = h \circ F$ . Then it is easy to show that  $\llbracket h(\mathcal{A}) \rrbracket = h \circ \llbracket \mathcal{A} \rrbracket$ .  $\square$

The second result characterizes the class of weighted tree languages which can be run-recognized by crisp-deterministic wta.

**Lemma 3.4.** cf. [FKV21, Lm. 5.3] Let  $\mathcal{A}$  be a  $(\Sigma, B)$ -wta. Then the following statements are equivalent.

- (i) There exists a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{B} \rrbracket$ .
- (ii)  $\llbracket \mathcal{A} \rrbracket$  is a  $(\Sigma, B)$ -recognizable step mapping.
- (iii)  $\mathcal{A}$  has the finite-image property and the preimage property.

We remark that if  $\mathcal{B} = (Q, \delta, F)$  is a crisp-deterministic  $(\Sigma, B)$ -wta, then for each  $b \in B$  we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $\llbracket \mathcal{B} \rrbracket^{-1}(b)$ . Indeed, if  $b \notin \text{im}(F)$ , then  $\llbracket \mathcal{B} \rrbracket^{-1}(b) = \emptyset$ . Therefore let now  $b \in \text{im}(F)$ . From  $\mathcal{B}$  we immediately obtain a crisp-deterministic  $(\Sigma, \mathbb{B})$ -wta  $\mathcal{B}_b$  with the same state set and same transitions with non-zero weight such that the states of  $\mathcal{B}_b$  have final weight  $1 \in \mathbb{B}$  iff they have final weight  $b$  in  $\mathcal{B}$ . Then  $\mathcal{B}_b$  recognizes the tree language  $\llbracket \mathcal{B} \rrbracket^{-1}(b)$ . This proves our remark.

The third result shows that each  $(\Sigma, B)$ -wta is crisp-determinizable if  $B$  is bi-locally finite. Formally, a  $(\Sigma, B)$ -wta  $\mathcal{A}$  is *crisp-determinizable (with respect to the run semantics)* if there exists a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{C}$  such that  $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{C} \rrbracket$ . Thus,  $\mathcal{A}$  is crisp-determinizable if  $\mathcal{A}$  satisfies the conditions of Lemma 3.4.

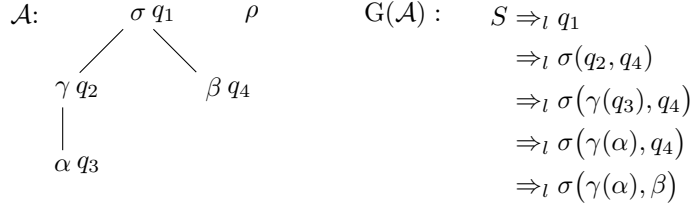


Figure 1: A run  $\rho \in R_{\mathcal{A}}(q_1, \xi)$ , where  $F_{q_1} \neq \emptyset$  and  $\xi = \sigma(\gamma(\alpha), \beta)$  and the leftmost derivation of  $G(\mathcal{A})$  for  $\xi$  which corresponds to  $\rho$ .

**Lemma 3.5.** [FKV21, Cor. 7.5] Let  $\mathcal{A}$  be a  $(\Sigma, B)$ -wta. If  $B$  is bi-locally finite, then  $\mathcal{A}$  is crisp-determinizable.

*In the rest of this paper, we let  $\mathcal{A} = (Q, \delta, F)$  be an arbitrary  $(\Sigma, B)$ -wta.*

## 4 Trim wta

In this section we define the concept of trim wta. Moreover, we show that, for each wta which satisfies certain simple properties, an equivalent trim wta can be constructed effectively. For this, to each  $(\Sigma, B)$ -wta  $\mathcal{A}$ , we associate a context-free grammar  $G(\mathcal{A})$  and show that  $\mathcal{A}$  is trim if and only if  $G(\mathcal{A})$  is reduced. Then we exploit the fact that for each context-free grammar one can construct effectively an equivalent reduced context-free grammar [Har78, Thm. 3.2.3].

A state  $p \in Q$  is *useful (in  $\mathcal{A}$ )* if there exist  $\xi \in T_{\Sigma}$  and  $\rho \in R_{\mathcal{A}}(\xi)$  such that  $F_{\rho(\varepsilon)} \neq \emptyset$  and  $p \in \text{im}(\rho)$ . The  $(\Sigma, B)$ -wta  $\mathcal{A}$  is *trim* if each of its states is useful.

Let  $G = (N, \Delta, P, S)$  be a context-free grammar [Har78, HMU07], with nonterminal set  $N$ , terminal set  $\Delta$ , set  $P$  of rules, and initial nonterminal  $S \in N$ . We denote by  $L(G)$  the language generated by  $G$ . A nonterminal  $A \in N$  is *useful (in  $G$ )* if there exist  $\alpha, \beta \in (N \cup \Delta)^*$  and  $w \in \Delta^*$  such that  $S \Rightarrow^* \alpha A \beta \Rightarrow^* w$ . Then  $G$  is *reduced* if each of its nonterminals is useful [Har78, p. 78].

To each  $(\Sigma, B)$ -wta  $\mathcal{A}$ , we associate the context-free grammar  $G(\mathcal{A}) = (N, \Delta, P, S)$  where  $S$  is a new symbol,  $N = Q \cup \{S\}$ ,  $\Delta = \Sigma \cup \Xi$  and  $\Xi$  consists of the two parentheses ( and ) and the comma, and  $P$  is defined as follows:

- for each  $q \in Q$ , if  $F_q \neq \emptyset$ , then  $S \rightarrow q$  is in  $P$  and
- for every  $k \in \mathbb{N}$ ,  $\sigma \in \Sigma^{(k)}$ ,  $q_1, \dots, q_k, q \in Q$ : if  $\delta_k(q_1 \cdots q_k, \sigma, q) \neq \emptyset$ , then  $q \rightarrow \sigma(q_1, \dots, q_k)$  is in  $P$ .

Then it can be shown that, for each  $\xi \in T_{\Sigma}$ , there exists a bijection between the set of runs  $\rho \in R_{\mathcal{A}}(\xi)$  with  $F_{\rho(\varepsilon)} \neq \emptyset$  and the set of leftmost derivations of  $G(\mathcal{A})$  for  $\xi$ , cf. Figure 1. Hence a state  $p \in Q$  is useful in  $\mathcal{A}$  if and only if it is useful in  $G(\mathcal{A})$ . This implies that  $\mathcal{A}$  is trim if and only if  $G(\mathcal{A})$  is reduced.

We say that the strong bimonoid  $B$  has an *effective test for  $\emptyset$*  if for each  $b \in B$  we can decide whether  $b = \emptyset$ .

**Theorem 4.1.** [DFKV20, Lm. 5] *Let  $B$  have an effective test for  $\emptyset$  and  $\mathcal{A}$  be a  $(\Sigma, B)$ -wta. If  $\mathcal{A}$  is given effectively and has a useful state, then we can construct effectively a  $(\Sigma, B)$ -wta  $\mathcal{A}'$  such that  $\mathcal{A}'$  is trim and  $\llbracket \mathcal{A}' \rrbracket = \llbracket \mathcal{A} \rrbracket$ . If  $\mathcal{A}$  is finitely ambiguous, then  $\mathcal{A}'$  is so.*

*Proof.* Using the effective test for  $\emptyset$ , we can construct effectively the context-free grammar  $G(\mathcal{A}) = (N, \Delta, P, S)$  (recall that  $N = Q \cup \{S\}$ ). Due to our assumption on  $\mathcal{A}$  we have  $L(G(\mathcal{A})) \neq \emptyset$ . Thus, by [Har78, Thm. 3.2.3], a context-free grammar  $G' = (N', \Delta, P', S)$  can be constructed effectively such that  $G'$  is reduced and  $L(G') = L(G(\mathcal{A}))$ . By the proof of that theorem, we know that  $N' = Q' \cup \{S\}$ , where  $Q'$  is the set of all useful nonterminals in  $Q$ . Hence  $Q'$  is the set of all useful states of  $\mathcal{A}$ . Moreover,  $Q' \neq \emptyset$  by our assumption on  $\mathcal{A}$ .

Now let  $\mathcal{A}' = (Q', \delta', F')$  be the  $(\Sigma, B)$ -wta such that, for each  $k \in \mathbb{N}$ ,  $\delta'_k = \delta_k|_{(Q')^k \times \Sigma^{(k)} \times Q'}$ , and  $F' = F|_{Q'}$ . It is obvious that  $\mathcal{A}'$  is trim and  $\mathcal{A}'$  is finitely ambiguous if  $\mathcal{A}$  is so.

Lastly we prove that  $\llbracket \mathcal{A} \rrbracket = \llbracket \mathcal{A}' \rrbracket$ . Let  $\xi \in T_\Sigma$ . Obviously,  $R_{\mathcal{A}'}(\xi) \subseteq R_{\mathcal{A}}(\xi)$  and for each  $\rho \in R_{\mathcal{A}'}(\xi)$  we have  $\text{wt}_{\mathcal{A}'}(\xi, \rho) = \text{wt}_{\mathcal{A}}(\xi, \rho)$ . If  $\rho \in R_{\mathcal{A}}(\xi) \setminus R_{\mathcal{A}'}(\xi)$ , then there exist  $p \in \text{im}(\rho)$  such that  $p$  is not useful. Then  $F_{\rho(\varepsilon)} = \emptyset$  and hence  $\text{wt}_{\mathcal{A}}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = \emptyset$ . Thus we can compute

$$\llbracket \mathcal{A} \rrbracket(\xi) = \bigoplus_{\rho \in R_{\mathcal{A}}(\xi)} \text{wt}_{\mathcal{A}}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = \bigoplus_{\rho \in R_{\mathcal{A}'}(\xi)} \text{wt}_{\mathcal{A}'}(\xi, \rho) \otimes F'_{\rho(\varepsilon)} = \llbracket \mathcal{A}' \rrbracket(\xi).$$

□

## 5 Pumping lemma

In this section, we wish to prove a pumping lemma for runs of weighted tree automata. Pumping lemmas are used in order to achieve structural implications on small or particular large trees (cf. [GS84, Lm. 2.10.1] and [Bor04, Lm. 5.5]). Essentially, we follow the classical approach for unweighted tree automata combined with an analysis of Equality (1). Assume we are given a wta  $\mathcal{A}$  with state set  $Q$ , a tree  $\xi \in T_\Sigma$  with height greater than  $|Q|$  and a run  $\kappa$  of  $\mathcal{A}$  on  $\xi$ . As for unweighted tree automata, choose a path, i.e., a linearly ordered subset of positions, in  $\xi$  whose length equals  $\text{height}(\xi)$ . Clearly, there are two positions  $u, v$  in this path with  $\kappa(u) = \kappa(v)$  in  $Q$ ; say,  $u$  is above  $v$ , i.e., there exists  $w \in \mathbb{N}_+^*$  such that  $v = uw$  (cf. Figure 2). Now we consider the subtrees  $\xi|_u$  (respectively,  $\xi|_v$ ) comprising all positions of  $\xi$  which are equal to or below  $u$  (respectively,  $v$ ). Clearly, we can cut out the context  $(\xi|_u)^w$ , thus replacing the subtree  $\xi|_u$  by  $\xi|_v$  and obtaining a smaller tree for which a restriction of the run  $\kappa$  leads to the same state as  $\kappa$ . But, we can also substitute a copy of the context  $(\xi|_u)^w$  at position  $v$ . We obtain a tree  $\xi'$ , and we can copy the corresponding part of the mapping  $\kappa$  to obtain a run  $\kappa'$  on  $\xi'$  leading again to the same final state as  $\kappa$ .

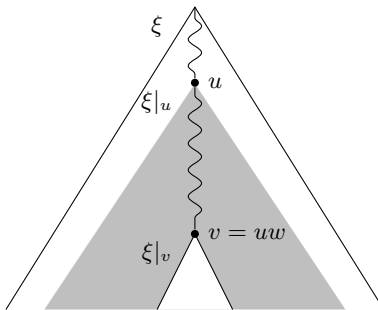


Figure 2: Illustration of the tree  $\xi$ , the positions  $u$  and  $v$ , and the subtrees  $\xi|_u$  and  $\xi|_v$ . Moreover, the context  $(\xi|_u)^w$  is shaded.

Now when we calculate the weight of  $\kappa'$  on  $\xi'$  according to Equality (1), a careful analysis shows that after an insertion process two factors of the product of weights originating from transitions at positions of the context  $(\xi|_u)^w$  get repeated. Hence, if we repeat the insertion process, then the two factors are replaced by their powers.

To make this outline formally exact, we show that the product of weights of the run  $\kappa$  on the context  $(\xi|_u)^w$  splits into two factors, a 'left one' and 'right one' (see Observation 5.1). This product of weights with the insertion process performed once is analyzed in Lemma 5.2; it turns out that splitting described before leads to an additional 'left factor' and an additional 'right factor'. The consequence of the cutting respectively replacing process in general for the product of weights is described in Lemma 5.3; the multiple insertion process leads to an additional 'left power' and an additional 'right power'. The whole pumping lemma is given in Theorem 5.4.

The question may arise why we present another pumping lemma and do not use an existing one (cf. [Bor04, Lm. 5.5]). To answer this we note that Borchardt's setting deals with deterministic wta over semirings and employs initial algebra semantics, whereas in Theorem 5.4 we deal with (arbitrary) wta over strong bimonoids and employ run semantics. Nevertheless, if we consider the class of all deterministic wta over semirings, then the two settings coincide.

Now we introduce some notations.

Let  $c \in C_\Sigma$ ,  $\zeta \in T_\Sigma$ ,  $\{v\} = \text{pos}_\square(c)$ ,  $q', q \in Q$ ,  $\rho \in R_{\mathcal{A}}(q', c, q)$ , and  $\theta \in R_{\mathcal{A}}(q, \zeta)$ . The *combination of  $\rho$  and  $\theta$* , denoted by  $\rho[\theta]$ , is the mapping  $\rho[\theta] : \text{pos}(c[\zeta]) \rightarrow Q$  defined for every  $u \in \text{pos}(c[\zeta])$  as follows: if  $u = vw$  for some  $w$ , then we define  $\rho[\theta](u) = \theta(w)$ , otherwise we define  $\rho[\theta](u) = \rho(u)$ . Clearly,  $\rho[\theta] \in R_{\mathcal{A}}(q', c[\zeta])$ . For every  $\xi \in T_\Sigma$ ,  $\rho \in R_{\mathcal{A}}(\xi)$ , and  $v \in \text{pos}(\xi)$ , we define the run  $\rho|_v$  on the context  $\xi|_v$  such that for every  $w \in \text{pos}(\xi|_v)$  we set  $\rho|_v(w) = \rho(w)$ . If  $\rho \in R_{\mathcal{A}}(\square)$  with  $\rho(\varepsilon) = q$  for some  $q \in Q$ , then sometimes we write  $\tilde{q}$  for  $\rho$ .

Let  $c \in C_\Sigma$ ,  $\{v\} = \text{pos}_\square(c)$ , and  $\rho \in R_{\mathcal{A}}(c)$ . We define two mappings  $l_{c,\rho} : \text{prefix}(v) \rightarrow B$  and  $r_{c,\rho} : \text{prefix}(v) \rightarrow B$  inductively on the length of their arguments (cf. [Bor04, p. 526] for deterministic wta). Intuitively, the product

(1), which yields the element  $\text{wt}(c, \rho) \in B$ , can be split into a left subproduct  $l_{c, \rho}(\varepsilon)$  and a right subproduct  $r_{c, \rho}(\varepsilon)$ , where the border is given by the factor  $\mathbb{1}$  coming from the weight of  $\square$ . Figure 3 shows the illustration of mappings  $l_{c, \rho}$  and  $r_{c, \rho}$ . Formally, let  $w \in \text{prefix}(v)$ . Then, assuming that  $c(w) = \sigma$  and  $\text{rk}(\sigma) = k$ , we let

$$l_{c, \rho}(w) = \begin{cases} \mathbb{1} & \text{if } w = v \\ \bigotimes_{j=1}^{i-1} \text{wt}(c|_{wj}, \rho|_{wj}) \otimes l_{c, \rho}(wi) & \text{if } wi \in \text{prefix}(v) \text{ for some } i \in \mathbb{N}_+ \end{cases}$$

$$r_{c, \rho}(w) = \begin{cases} \mathbb{1} & \text{if } w = v \\ r_{c, \rho}(wi) \otimes \bigotimes_{j=i+1}^k \text{wt}(c|_{wj}, \rho|_{wj}) \otimes \delta_k(\rho(w1) \cdots \rho(wk), \sigma, \rho(w)) & \text{if } wi \in \text{prefix}(v) \text{ for some } i \in \mathbb{N}_+ . \end{cases}$$

In the sequel, we abbreviate  $l_{c, \rho}(\varepsilon)$  and  $r_{c, \rho}(\varepsilon)$  by  $l_{c, \rho}$  and  $r_{c, \rho}$ , respectively.

**Observation 5.1.** Let  $c \in C_\Sigma$  and  $\rho \in R_{\mathcal{A}}(c)$ . Then  $\text{wt}(c, \rho) = l_{c, \rho} \otimes r_{c, \rho}$ .

The next lemma can be proved by an easy and straightforward induction on  $c$ .

**Lemma 5.2.** (cf. [Bor04, Lm. 5.1]) Let  $c \in C_\Sigma$ ,  $\zeta \in T_\Sigma$ ,  $q', q \in Q$ ,  $\rho \in R_{\mathcal{A}}(q', c, q)$ , and  $\theta \in R_{\mathcal{A}}(q, \zeta)$ . Then  $\text{wt}(c[\zeta], \rho[\theta]) = l_{c, \rho} \otimes \text{wt}(\zeta, \theta) \otimes r_{c, \rho}$ .

Let  $c \in C_\Sigma$ ,  $q \in Q$ , and  $\rho \in R_{\mathcal{A}}(q, c, q)$  be a loop. For each  $n \in \mathbb{N}$ , the  $n$ th power of  $\rho$ , denoted by  $\rho^n$ , is the run on  $c^n$  defined by induction as follows:  $\rho^0 = \tilde{q}$  (note that  $c^0 = \square$ ) and  $\rho^{n+1} = \rho[\rho^n]$ . Next we apply the previous results to the weights of powers of loops.

**Lemma 5.3.** (cf. [Bor04, Lm. 5.3]) Let  $c', c \in C_\Sigma$  and  $\zeta \in T_\Sigma$ ,  $q', q \in Q$ ,  $\rho' \in R_{\mathcal{A}}(q', c', q)$ ,  $\rho \in R_{\mathcal{A}}(q, c, q)$ , and  $\theta \in R_{\mathcal{A}}(q, \zeta)$ . Then, for each  $n \in \mathbb{N}$ ,

$$\text{wt}(c' [c^n[\zeta]], \rho' [\rho^n[\theta]]) = l_{c', \rho'} \otimes (l_{c, \rho})^n \otimes \text{wt}(\zeta, \theta) \otimes (r_{c, \rho})^n \otimes r_{c', \rho'} .$$

*Proof.* We can prove easily by induction on  $n$  that

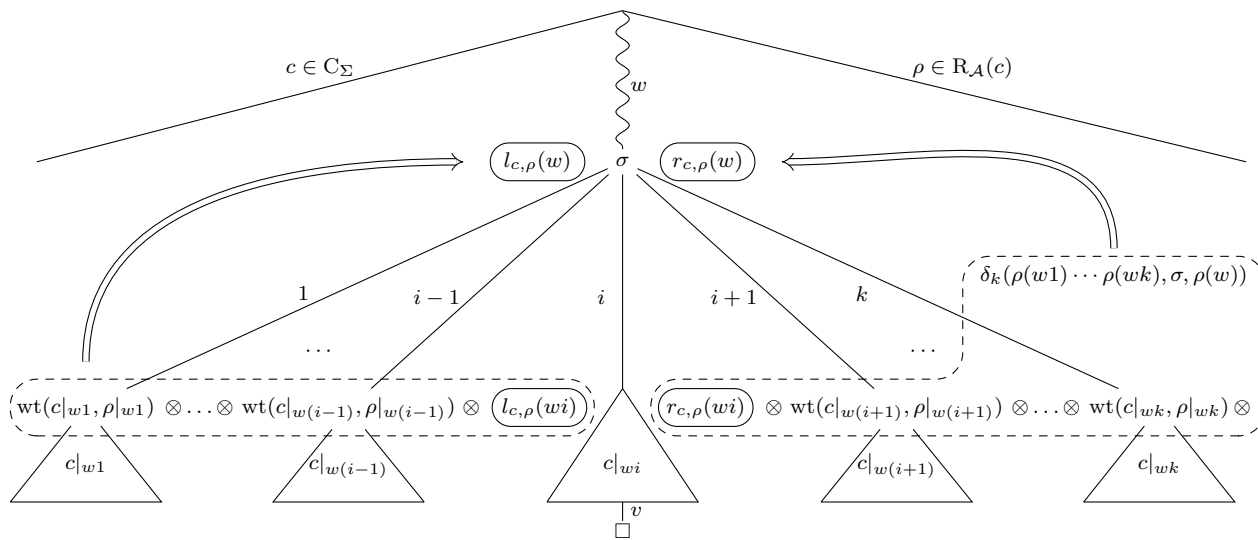
$$\text{wt}(c^n[\zeta], \rho^n[\theta]) = (l_{c, \rho})^n \otimes \text{wt}(\zeta, \theta) \otimes (r_{c, \rho})^n \text{ for each } n \in \mathbb{N} . \quad (2)$$

Then for each  $n \in \mathbb{N}$  we have

$$\begin{aligned} \text{wt}(c' [c^n[\zeta]], \rho' [\rho^n[\theta]]) &= l_{c', \rho'} \otimes \text{wt}(c^n[\zeta], \rho^n[\theta]) \otimes r_{c', \rho'} && \text{(by Lemma 5.2)} \\ &= l_{c', \rho'} \otimes (l_{c, \rho})^n \otimes \text{wt}(\zeta, \theta) \otimes (r_{c, \rho})^n \otimes r_{c', \rho'} && \text{(by (2))} \end{aligned}$$

□

Finally, we recall from [Bor04] the pumping lemma for runs of  $\mathcal{A}$  on trees in  $T_\Sigma$  which are large enough. We note that  $B$  need not be commutative.

Figure 3: Illustration of mappings  $l_{c,\rho}$  and  $r_{c,\rho}$



**Theorem 5.4.** (*pumping lemma, cf. [Bor04, Lm. 5.5]*) Let  $\xi \in T_\Sigma$ ,  $q' \in Q$ ,  $\kappa \in R_{\mathcal{A}}(q', \xi)$ . If  $\text{height}(\xi) \geq |Q|$ , then there exist  $c', c \in C_\Sigma$ ,  $\zeta \in T_\Sigma$ ,  $q \in Q$ ,  $\rho' \in R_{\mathcal{A}}(q', c', q)$ ,  $\rho \in R_{\mathcal{A}}(q, c, q)$ , and  $\theta \in R_{\mathcal{A}}(q, \zeta)$  such that  $\xi = c'[c[\zeta]]$ ,  $\kappa = \rho'[\rho[\theta]]$ ,  $\text{height}(c) > 0$ ,  $\text{height}(c[\zeta]) < |Q|$ , and, for each  $n \in \mathbb{N}$ ,

$$\text{wt}(c'[c^n[\zeta]], \rho'[\rho^n[\theta]]) = l_{c', \rho'} \otimes (l_{c, \rho})^n \otimes \text{wt}(\zeta, \theta) \otimes (r_{c, \rho})^n \otimes r_{c', \rho'} .$$

*Proof.* Since  $\text{height}(\xi) \geq |Q|$  there exist  $u, w \in \mathbb{N}_+^*$  such that  $uw \in \text{pos}(\xi)$ ,  $|w| > 0$ ,  $\text{height}(\xi|_u) < |Q|$ , and  $\kappa(u) = \kappa(uw)$ . Then we let  $c' = \xi|_u$ ,  $c = (\xi|_u)^w$ ,  $\zeta = \xi|_{uw}$ . Clearly,  $\xi = c'[c[\zeta]]$ . Moreover, we set  $\rho' = \kappa|_u$ ,  $\rho = (\kappa|_u)^w$  and  $\theta = \kappa|_{uw}$ . Then the statement follows from Lemma 5.3.  $\square$

We say that *small loops of  $\mathcal{A}$  have weight 1* if, for every  $q \in Q$ ,  $c \in C_\Sigma$ , and loop  $\rho \in R_{\mathcal{A}}(q, c, q)$ , if  $\text{height}(c) < |Q|$ , then  $\text{wt}(c, \rho) = 1$ .

**Lemma 5.5.** Let  $B$  be commutative or one-product free. If small loops of  $\mathcal{A}$  have weight 1, then, for every  $\xi \in T_\Sigma$ ,  $q' \in Q$ , and  $\kappa \in R_{\mathcal{A}}(q', \xi)$ , there exist  $\xi' \in T_\Sigma$  and  $\kappa' \in R_{\mathcal{A}}(q', \xi')$  such that  $\text{height}(\xi') < |Q|$  and  $\text{wt}(\xi, \kappa) = \text{wt}(\xi', \kappa')$ .

*Proof.* Let  $\xi \in T_\Sigma$ ,  $q' \in Q$ , and  $\kappa \in R_{\mathcal{A}}(q', \xi)$ . We may assume that  $\text{height}(\xi) \geq |Q|$ . Applying Theorem 5.4 (for  $n = 1$  and  $n = 0$ ), there exist  $c, c' \in C_\Sigma$ ,  $\zeta \in T_\Sigma$ ,  $q \in Q$ ,  $\rho' \in R_{\mathcal{A}}(q', c', q)$ ,  $\rho \in R_{\mathcal{A}}(q, c, q)$ , and  $\theta \in R_{\mathcal{A}}(q, \zeta)$  such that  $\xi = c'[c[\zeta]]$ ,  $\kappa = \rho'[\rho[\theta]]$ ,  $\text{height}(c) > 0$ ,  $\text{height}(c[\zeta]) < |Q|$ , and

$$\begin{aligned} \text{wt}(\xi, \kappa) &= \text{wt}(c'[c[\zeta]], \rho'[\rho[\theta]]) = l_{c', \rho'} \otimes l_{c, \rho} \otimes \text{wt}(\zeta, \theta) \otimes r_{c, \rho} \otimes r_{c', \rho'} , \\ \text{wt}(c'[\zeta], \rho'[\theta]) &= l_{c', \rho'} \otimes \text{wt}(\zeta, \theta) \otimes r_{c', \rho'} . \end{aligned}$$

By our assumption  $\text{wt}(c, \rho) = 1$ , and by Observation 5.1 we have  $\text{wt}(c, \rho) = l_{c, \rho} \otimes r_{c, \rho}$ . If  $B$  is commutative, then  $\text{wt}(\xi, \kappa) = \text{wt}(c'[\zeta], \rho'[\theta]) \otimes l_{c, \rho} \otimes r_{c, \rho}$ . If  $B$  is one-product free, then  $l_{c, \rho} = r_{c, \rho} = 1$ . Hence in both cases we have  $\text{wt}(\xi, \kappa) = \text{wt}(c'[\zeta], \rho'[\theta])$ .

Note that  $\rho'[\theta] \in R_{\mathcal{A}}(q', c'[\zeta])$  and  $\text{size}(c'[\zeta]) < \text{size}(\xi)$ . If  $\text{height}(c'[\zeta]) < |Q|$ , then we are ready. Otherwise we continue with  $c'[\zeta]$ ,  $q'$ , and  $\rho'[\theta]$  as before. After finitely many steps, we obtain  $\xi' \in T_\Sigma$  and  $\kappa' \in R_{\mathcal{A}}(q', \xi')$  with  $\text{height}(\xi') < |Q|$  as required.  $\square$

## 6 The preimage property and a sufficient condition for a wta to be crisp-determinizable

An important result for recognizable weighted string languages  $r$  over the semiring  $\mathbb{N}$  is that, for each  $n \in \mathbb{N}$ , the string language  $r^{-1}(n)$  is recognizable (preimage property, [BR88, III. Cor. 2.5]). In this section we show some variants of the preimage property. Then, as a main result of this section, we give a sufficient condition for a wta  $\mathcal{A}$  over an arbitrary strong bimonoid  $B$  which guarantees that  $\llbracket \mathcal{A} \rrbracket$  is a recognizable step mapping (cf. Theorem 6.6) and thus, in particular,  $\llbracket \mathcal{A} \rrbracket$  satisfies the preimage property and  $\mathcal{A}$  is crisp-determinizable

by Lemma 3.4. We also show that if the strong bimonoid is computable and  $\mathcal{A}$  is given effectively, then the crisp-deterministic wta equivalent to  $\mathcal{A}$  can be constructed effectively. Finally, as an application of Theorem 6.6, we extend the mentioned preimage property [BR88, III. Cor. 2.5] to wta over past-finite monotonic strong bimonoids (cf. Theorem 6.10).

We say that  $(B, \oplus, \otimes, 0, 1)$  is *computable* if  $B$  is a recursively enumerable set with tests for equality and the operations  $\oplus$  and  $\otimes$  are computable (e.g., by a Turing machine).

Subsequently, we will need the following concepts. We define the sets

$$\begin{aligned} \mathsf{H}(\mathcal{A}) &= \{\text{wt}(\xi, \rho) \mid \xi \in \mathsf{T}_\Sigma \text{ and } \rho \in \mathsf{R}_{\mathcal{A}}(\xi)\} \text{ and} \\ \mathsf{C}(\mathcal{A}) &= \{\text{wt}(\xi, \rho) \otimes F_{\rho(\varepsilon)} \mid \xi \in \mathsf{T}_\Sigma, \rho \in \mathsf{R}_{\mathcal{A}}(\xi)\} . \end{aligned}$$

We call elements of  $\mathsf{C}(\mathcal{A})$  *complete run weights of  $\mathcal{A}$* . Observe that if  $\mathsf{H}(\mathcal{A})$  is a finite set, then  $\mathsf{C}(\mathcal{A})$  is also finite because  $\mathsf{C}(\mathcal{A}) \subseteq \mathsf{H}(\mathcal{A}) \otimes \text{im}(F)$ . The following notions will be needed for our main result of this section.

Let  $b \in B$ . If  $b$  has finite additive order, then there exists a least number  $i \in \mathbb{N}_+$  such that  $ib = (i+k)b$  for some  $k \in \mathbb{N}_+$ , and there exists a least number  $p \in \mathbb{N}_+$  such that  $ib = (i+p)b$ . We call  $i$  the *index (of  $b$ )* and  $p$  the *period (of  $b$ )*, and denote them by  $i(b)$  and  $p(b)$ , respectively. Moreover, we call  $i+p-1$ , i.e., the number of elements of  $\langle b \rangle_{\oplus}$ , the *order of  $b$* .

Then for each  $b \in \mathsf{C}(\mathcal{A})$ , we define the mapping  $f_{\mathcal{A},b} : \mathsf{T}_\Sigma \rightarrow \mathbb{N}$ , called the *complete run number mapping of  $b$* , by

$$f_{\mathcal{A},b}(\xi) = |\{\rho \in \mathsf{R}_{\mathcal{A}}(\xi) \mid \text{wt}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = b\}|$$

for each  $\xi \in \mathsf{T}_\Sigma$ . The mapping  $f_{\mathcal{A},b}$  is *bounded*, if there exists  $K \in \mathbb{N}$  such that  $f_{\mathcal{A},b}(\xi) \leq K$  for each  $\xi \in \mathsf{T}_\Sigma$ . Clearly, if  $\mathcal{A}$  is finitely ambiguous, then there exists  $K \in \mathbb{N}$  such that, for each  $\xi \in \mathsf{T}_\Sigma$ , we have  $|\mathsf{R}_{\mathcal{A}}^{F \neq 0}(\xi)| \leq K$ , and thus,  $f_{\mathcal{A},b}$  is bounded by  $K$  for each  $b \in \mathsf{C}(\mathcal{A}) \setminus \{0\}$ .

**Theorem 6.1.** *Let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta such that  $\mathsf{H}(\mathcal{A})$  is finite. If, for each  $b \in \mathsf{C}(\mathcal{A})$ , the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order, then  $\llbracket \mathcal{A} \rrbracket$  has the finite-image property.*

*Proof.* We note that  $\mathsf{C}(\mathcal{A})$  is finite because  $\mathsf{H}(\mathcal{A})$  is finite. Then for each  $\xi \in \mathsf{T}_\Sigma$ , we have

$$\llbracket \mathcal{A} \rrbracket(\xi) = \bigoplus_{b \in \mathsf{C}(\mathcal{A})} (f_{\mathcal{A},b}(\xi))b . \quad (3)$$

Let  $b \in \mathsf{C}(\mathcal{A})$ . If the mapping  $f_{\mathcal{A},b}$  is bounded by  $K$  for some  $K \in \mathbb{N}$ , then  $(f_{\mathcal{A},b}(\xi))b \in \{jb \mid j \in [0, K]\}$ . Otherwise  $b$  has finite additive order. Thus we have  $(f_{\mathcal{A},b}(\xi))b \in \{0, b, 2b, \dots, (i(b) + p(b) - 1)b\}$ .  $\square$

Next, we wish to show that under the assumptions of Theorem 6.1,  $\mathcal{A}$  also has the preimage property. For the main result of this section, we will need the following preparation.

**Lemma 6.2.** Let  $B$  be a finite semiring and  $\mathcal{A}$  be a  $(\Sigma, B)$ -wta. Then the following statements hold.

1.  $\mathcal{A}$  has the preimage property.
2. If  $B$  is computable and  $\mathcal{A}$  is given effectively, then, for each  $b \in B$ , we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $\llbracket \mathcal{A} \rrbracket^{-1}(b)$ .

*Proof.* If, in addition,  $B$  is commutative, then Statement 1 immediately follows from [DV06, Lm. 6.1] and Statement 2 is also clear by the proof of that lemma. However, we can drop the condition that  $B$  is commutative because of the following. In [DV06], for each  $\xi \in \mathbb{T}_\Sigma$ , the weight of a run  $\rho \in \mathbb{R}_\mathcal{A}(\xi)$  is defined as the product of the weight of the transitions determined by the run:  $\text{wt}(\xi, \rho) = \bigotimes_{w \in \text{pos}(\xi)} \delta_k(\rho(w1) \cdots \rho(wk), \xi(w), \rho(w))$ , where  $\xi(w) \in \Sigma^{(k)}$  and the factors are multiplied in an arbitrary order. However, in the proof of [DV06, Lm. 6.1], we observe that the order of the factors in such products does not change. Hence that proof is also valid for an arbitrary, fixed order of elements in the products. In particular, it is valid for the depth-first left-to-right order with which we have defined  $\text{wt}(\xi, \rho)$  (cf. (1)). Hence the proof of [DV06, Lm. 6.1] is also valid for our setting.  $\square$

**Lemma 6.3.** Let  $\mathcal{A}$  be a  $(\Sigma, \mathbb{N})$ -wta. Then the following statements hold.

1.  $\mathcal{A}$  has the preimage property.
2. If  $\mathcal{A}$  is given effectively, then, for each  $n \in \mathbb{N}$ , we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $\llbracket \mathcal{A} \rrbracket^{-1}(n)$ .

*Proof.* Statement 1 was proved in [DV06, Lm. 6.3(2)]. Now we prove Statement 2. It was stated already in [DV06], but we include the proof for the sake of completeness. Let  $n \in \mathbb{N}$  and  $M = \{k \in \mathbb{N} \mid n < k\}$ . Moreover, let  $\sim$  be the equivalence relation on the set  $\mathbb{N}$  defined such that its classes are the singleton sets  $\{k\}$  for each  $k \in [0, n]$  and the set  $M$ . As is well known,  $\sim$  is a congruence, which can be seen as follows. Let  $k, k' \in M$  and  $m \in \mathbb{N}$ . Obviously,  $k \leq k + m$ , and hence  $k + m \in M$  and similarly  $k' + m \in M$ . Moreover, if  $m \neq 0$ , then we have  $k \cdot m \in M$  and  $k' \cdot m \in M$ . Thus  $M$  is a congruence class and the relation  $\sim$  is a congruence on  $\mathbb{N}$ . Then the quotient semiring  $\mathbb{N}/\sim$  is finite. Let  $h : \mathbb{N} \rightarrow \mathbb{N}/\sim$  be the canonical semiring homomorphism. Clearly, we can give effectively the congruence classes of  $\sim$ , i.e., the elements of  $\mathbb{N}/\sim$ , by choosing only one representative for each congruence class. Due to this fact and that  $\mathbb{N}$  is computable, the semiring  $\mathbb{N}/\sim$  is computable. By Lemma 3.3, we can construct effectively the  $(\Sigma, \mathbb{N}/\sim)$ -wta  $h(\mathcal{A})$  such that  $\llbracket h(\mathcal{A}) \rrbracket = h \circ \llbracket \mathcal{A} \rrbracket$ . Since  $\mathbb{N}/\sim$  is finite and computable, by Lemma 6.2(2), we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $(h \circ \llbracket \mathcal{A} \rrbracket)^{-1}(\{n\}) = \llbracket \mathcal{A} \rrbracket^{-1}(n)$ .  $\square$

For every  $m \in \mathbb{N}$  and  $n \in \mathbb{N}_+$ , we define  $m + n \cdot \mathbb{N} = \{m + n \cdot j \mid j \in \mathbb{N}\}$ . Let us denote by  $\mathbb{N}/n\mathbb{N}$  the semiring of nonnegative integers modulo  $n$ . Moreover, for  $m \in \mathbb{N}$ , we let  $\overline{m} = m + n\mathbb{N}$ , the residue class of  $m$  modulo  $n$ .

**Lemma 6.4.** (cf. [BR88, III. Cor. 2.4]) Let  $\mathcal{A}$  be a  $(\Sigma, \mathbb{N})$ -wta. Then the following statements hold.

1. For every  $m \in \mathbb{N}$  and  $n \in \mathbb{N}_+$ , the  $\Sigma$ -tree language  $[[\mathcal{A}]]^{-1}(m + n \cdot \mathbb{N})$  is recognizable.
2. If  $\mathcal{A}$  is given effectively, then, for every  $m \in \mathbb{N}$  and  $n \in \mathbb{N}_+$ , we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $[[\mathcal{A}]]^{-1}(m + n \cdot \mathbb{N})$ .

*Proof.* Let us abbreviate  $[[\mathcal{A}]]$  by  $r$ .

Proof of 1: Let  $m \in \mathbb{N}$ . If  $m < n$ , then, by Lemma 3.3,  $(h \circ r) \in \text{Rec}(\Sigma, \mathbb{N}/n\mathbb{N})$ , where  $h : \mathbb{N} \rightarrow \mathbb{N}/n\mathbb{N}$  is the canonical semiring homomorphism. Moreover,  $r^{-1}(m + n \cdot \mathbb{N}) = r^{-1}(h^{-1}(\overline{m})) = (h \circ r)^{-1}(\overline{m})$ . Since  $\mathbb{N}/n\mathbb{N}$  is a finite semiring, by Lemma 6.2(1), the  $\Sigma$ -tree language  $(h \circ r)^{-1}(\overline{m})$  is recognizable. Now assume that  $m \geq n$ . Then there exist  $m' \in [0, n-1]$  and  $k \in \mathbb{N}_+$  such that  $m = m' + n \cdot k$ . Then

$$r^{-1}(m + n \cdot \mathbb{N}) = r^{-1}(m' + n \cdot \mathbb{N}) \setminus \bigcup_{j=0}^{k-1} r^{-1}(m' + n \cdot j) .$$

As we saw, the  $\Sigma$ -tree language  $r^{-1}(m' + n \cdot \mathbb{N})$  is recognizable because  $m' < n$ . Moreover, by Lemma 6.3(1), for each  $j \in [0, k-1]$ , the  $\Sigma$ -tree language  $r^{-1}(m' + n \cdot j)$  is also recognizable. Finally,  $\Sigma$ -tree languages are closed under union and subtraction. Thus, also in this case,  $r^{-1}(m + n \cdot \mathbb{N})$  is recognizable.

Proof of 2: We follow the proof of Statement 1. Let  $m \in \mathbb{N}$ . Assume that  $m < n$ . Obviously, we can give effectively the residue classes modulo  $n$ , i.e., the elements of  $\mathbb{N}/n\mathbb{N}$ , by choosing only one representative for each residue class. Because of this fact and that  $\mathbb{N}$  is computable, the semiring  $\mathbb{N}/n\mathbb{N}$  is also computable. Since  $\mathcal{A}$  is given effectively, by Lemma 3.3, we can construct effectively the  $(\Sigma, \mathbb{N}/n\mathbb{N})$ -wta  $h(\mathcal{A})$  such that  $[[h(\mathcal{A})]] = h \circ r$ . Since  $\mathbb{N}/n\mathbb{N}$  is a computable finite semiring, by Lemma 6.2(2), we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $(h \circ r)^{-1}(\overline{m}) = r^{-1}(m + n \cdot \mathbb{N})$ . Now assume that  $m \geq n$ . Since  $m' < n$ , by the above, we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $r^{-1}(m' + n \cdot \mathbb{N})$ . Moreover, by Lemma 6.3(2), for each  $j \in [0, k-1]$ , we can also construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $r^{-1}(m' + n \cdot j)$ . Lastly,  $\Sigma$ -tree languages are closed effectively under union and subtraction, and thus, we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $r^{-1}(m + n \cdot \mathbb{N})$ .  $\square$

**Lemma 6.5.** Let  $B$  be computable. If a  $(\Sigma, B)$ -wta  $\mathcal{A} = (Q, \delta, F)$  is given effectively and  $H(\mathcal{A})$  is finite, then we can compute the sets  $H(\mathcal{A})$  and  $C(\mathcal{A})$ .

*Proof.* First we prove that the set  $H(\mathcal{A})$  can be computed. For every  $i \in \mathbb{N}$  and  $q \in Q$  let

$$H_{i,q} = \{\text{wt}(\xi, \rho) \mid \xi \in T_\Sigma, \text{height}(\xi) \leq i, \rho \in R_{\mathcal{A}}(q, \xi)\} .$$

Clearly, we have  $H_{0,q} \subseteq H_{1,q} \subseteq \dots \subseteq H(\mathcal{A})$  for each  $q \in Q$ . We claim that, for each  $i \in \mathbb{N}$ ,

$$\text{if } \forall q \in Q: H_{i,q} = H_{i+1,q}, \text{ then } \forall q \in Q: H_{i+1,q} = H_{i+2,q} .$$

To show this, let  $i \in \mathbb{N}$ ,  $q \in Q$ , and  $b \in H_{i+2,q}$ . There exist  $\xi \in T_\Sigma$  and  $\rho \in R_{\mathcal{A}}(q, \xi)$  such that  $\text{height}(\xi) \leq i + 2$  and  $\text{wt}(\xi, \rho) = b$ . We may assume that  $\text{height}(\xi) = i + 2$ . Hence  $\xi = \sigma(\xi_1, \dots, \xi_k)$  such that  $\text{height}(\xi_j) \leq i + 1$  for each  $j \in [k]$ . Clearly, for each  $j \in [k]$ , we have  $\text{wt}(\xi_j, \rho|_j) \in H_{i+1, \rho(j)}$ , so by our assumption there exist  $\zeta_j \in T_\Sigma$  with  $\text{height}(\zeta_j) \leq i$  and run  $\theta_j \in R_{\mathcal{A}}(\rho(j), \zeta_j)$  such that  $\text{wt}(\xi_j, \rho|_j) = \text{wt}(\zeta_j, \theta_j)$ .

Now let  $\zeta = \sigma(\zeta_1, \dots, \zeta_k)$ . Obviously,  $\text{height}(\zeta) \leq i + 1$ . Moreover, let  $\theta \in R_{\mathcal{A}}(q, \zeta)$  such that  $\theta|_j = \theta_j$  for each  $j \in [k]$ . Clearly,  $\text{wt}(\zeta, \theta) \in H_{i+1,q}$ , and we calculate

$$\begin{aligned} \text{wt}(\zeta, \theta) &= \left( \bigotimes_{j=1}^k \text{wt}(\zeta_j, \theta|_j) \right) \otimes \delta_k(\theta(1) \cdots \theta(j), \sigma, q) \\ &= \left( \bigotimes_{j=1}^k \text{wt}(\xi_j, \rho|_j) \right) \otimes \delta_k(\rho(1) \cdots \rho(j), \sigma, q) = \text{wt}(\xi, \rho) = b . \end{aligned}$$

This shows that  $b \in H_{i+1,q}$ , proving our claim.

We recall that  $H_{0,q} \subseteq H_{1,q} \subseteq \dots \subseteq H(\mathcal{A})$  for each  $q \in Q$ . Since  $B$  is computable, we can compute  $H_{i,q}$  for every  $i \in \mathbb{N}$  and  $q \in Q$ . Then, since  $H(\mathcal{A})$  is finite, by computing  $H_{0,q}$  for each  $q \in Q$ ,  $H_{1,q}$  for each  $q \in Q$ , and so on, we can find the least number  $i_m \in \mathbb{N}$  such that  $H_{i_m,q} = H_{i_m+1,q}$  for each  $q \in Q$  and thus by the implication shown above  $H_{i_m,q} = H_{j,q}$  for every  $q \in Q$  and  $j \in \mathbb{N}$  with  $j \geq i_m$ . We show that  $H(\mathcal{A}) = \bigcup_{q \in Q} H_{i_m,q}$ . For this, let  $b \in H(\mathcal{A})$ , i.e.,  $b = \text{wt}(\xi, \rho)$  for some  $\xi \in T_\Sigma$  with  $\text{height}(\xi) = j$ ,  $q \in Q$  and  $\rho \in R_{\mathcal{A}}(q, \xi)$ . Then  $b \in H_{j,q} = H_{i_m,q}$ . The other inclusion is obvious. Since we can compute the set  $\bigcup_{q \in Q} H_{i_m,q}$ , the set  $H(\mathcal{A})$  can be computed.

Now we prove that the set  $C(\mathcal{A})$  can be computed. Let  $i_m$  be the number as before. It suffices to show that

$$C(\mathcal{A}) = \{ \text{wt}(\xi, \rho) \otimes F_{\rho(\varepsilon)} \mid \xi \in T_\Sigma, \text{height}(\xi) \leq i_m, \rho \in R_{\mathcal{A}}(\xi) \} ,$$

because we can compute the set on the right-hand side of the above equality. Let us denote this set by  $C$ . It is obvious that  $C \subseteq C(\mathcal{A})$ . For the proof of the other inclusion, let  $b \in C(\mathcal{A})$ , i.e.,  $b = \text{wt}(\xi, \rho) \otimes F_q$  for some  $\xi \in T_\Sigma$ ,  $q \in Q$ , and  $\rho \in R_{\mathcal{A}}(q, \xi)$ . Since  $\text{wt}(\xi, \rho) \in H(\mathcal{A})$ , by the proof of computing the set  $H(\mathcal{A})$ , we have  $\text{wt}(\xi, \rho) \in H_{i_m,q}$ , i.e., there exist  $\xi' \in T_\Sigma$  with  $\text{height}(\xi') \leq i_m$ , and  $\rho' \in R_{\mathcal{A}}(q, \xi')$  such that  $\text{wt}(\xi, \rho) = \text{wt}(\xi', \rho')$ . Hence  $b \in C$ .  $\square$

Next we present the main result of this section. It gives a structural condition on a  $(\Sigma, B)$ -wta  $\mathcal{A}$ , for arbitrary strong bimonoid  $B$ , which is sufficient to imply that  $\mathcal{A}$  has the finite-image property. Our result generalizes [DGMM11, Thm. 6.2(a)] and [DSV10, Thm. 11] from bi-locally finite strong bimonoids to arbitrary strong bimonoids, in case of [DSV10, Thm. 11] even from strings to trees.

**Theorem 6.6.** *Let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta such that  $H(\mathcal{A})$  is finite. If, for each  $b \in C(\mathcal{A})$ , the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order, then the following statements hold.*

1.  $\mathcal{A}$  has the finite-image property and the preimage property.
2. If  $B$  is computable and  $\mathcal{A}$  is given effectively, then we can construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ .

*Proof. Proof of 1:* We note that  $C(\mathcal{A})$  is finite because  $H(\mathcal{A})$  is finite. For each  $b \in C(\mathcal{A})$  we define the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}'_b = (Q', \delta', F'_b)$  as follows:  $Q' = Q \times H(\mathcal{A})$  and for every  $k \in \mathbb{N}$ ,  $\sigma \in \Sigma^{(k)}$  and  $(q_1, y_1), \dots, (q_k, y_k), (q, y) \in Q'$ , let

$$\delta'_k((q_1, y_1) \cdots (q_k, y_k), \sigma, (q, y)) = \begin{cases} 1 & \text{if } (\bigotimes_{i=1}^k y_i) \otimes \delta_k(q_1 \cdots q_k, \sigma, q) = y \\ 0 & \text{otherwise,} \end{cases}$$

and let

$$(F'_b)_{(q,y)} = \begin{cases} 1 & \text{if } y \otimes F_q = b \\ 0 & \text{otherwise.} \end{cases}$$

Let  $\xi \in T_\Sigma$  and  $b \in C(\mathcal{A})$ . We observe that there exists a bijection between the two sets

$$\{\rho \in R_{\mathcal{A}}(\xi) \mid \text{wt}_{\mathcal{A}}(\xi, \rho) \otimes F_{\rho(\varepsilon)} = b\} \text{ and } \{\rho' \in R_{\mathcal{A}'_b}(\xi) \mid \text{wt}_{\mathcal{A}'_b}(\xi, \rho') \cdot (F'_b)_{\rho'(\varepsilon)} = 1\}.$$

It follows that  $\llbracket \mathcal{A}'_b \rrbracket(\xi) = f_{\mathcal{A},b}(\xi)$ , and thus, by Equality (3), we have

$$\llbracket \mathcal{A} \rrbracket(\xi) = \bigoplus_{b \in C(\mathcal{A})} (\llbracket \mathcal{A}'_b \rrbracket(\xi))b.$$

Let us define the mapping  $r_b : T_\Sigma \rightarrow B$  by  $r_b(\xi) = (\llbracket \mathcal{A}'_b \rrbracket(\xi))b$  for each  $\xi \in T_\Sigma$ . Then  $\llbracket \mathcal{A} \rrbracket = \bigoplus_{b \in C(\mathcal{A})} r_b$  and it suffices to show that  $r_b$  is a recognizable step mapping for each  $b \in C(\mathcal{A})$  because, obviously, recognizable step mappings are closed under the operation  $\oplus$ .

To prove this latter, let  $b \in C(\mathcal{A})$ . We distinguish the following two cases.

Case 1: The mapping  $f_{\mathcal{A},b}$  is bounded, i.e., there exists  $K \in \mathbb{N}$  such that  $f_{\mathcal{A},b}(\xi) \leq K$  for each  $\xi \in T_\Sigma$ . Clearly,  $\text{im}(r_b) \subseteq \{jb \mid j \in [0, K]\}$ . For each  $j \in [0, K]$ , let  $L_{b,j} = \llbracket \mathcal{A}'_b \rrbracket^{-1}(j)$ . By Lemma 6.3(1),  $L_{b,j}$  is a recognizable  $\Sigma$ -tree language. By our assumption, we have  $\bigcup_{j \in [0, K]} L_{b,j} = T_\Sigma$ . Hence  $r_b = \bigoplus_{j \in [0, K]} (jb) \otimes \mathbb{1}_{(B, L_{b,j})}$ , i.e., it is a recognizable step mapping.

Case 2:  $b$  has finite additive order. Then we have  $\langle b \rangle_{\oplus} = \{0, b, 2b, \dots, (i(b) + p(b) - 1)b\}$ . So

$$(\forall n \in \mathbb{N})(\exists \text{ exactly one } j \in [0, i(b) + p(b) - 1] : nb = jb).$$

Now let  $L_{b,j} = \{\xi \in T_\Sigma \mid (\llbracket \mathcal{A}'_b \rrbracket(\xi))b = jb\}$  for each  $j \in [0, i(b) + p(b) - 1]$ . Observe that  $L_{b,0} = T_\Sigma$ . We claim that  $L_{b,j}$  is recognizable. We have

- $L_{b,j} = \llbracket \mathcal{A}'_b \rrbracket^{-1}(j)$  if  $0 \leq j < i(b)$ , and
- $L_{b,j} = \llbracket \mathcal{A}'_b \rrbracket^{-1}(j + p(b) \cdot \mathbb{N})$  if  $i(b) \leq j \leq i(b) + p(b) - 1$ .

Then  $L_{b,j}$  is recognizable in both cases, by Lemmas 6.3(1) and 6.4(1), respectively.

Let  $\xi \in T_\Sigma$ . By the above, there exists a unique number  $j \in [0, i(b)+p(b)-1]$  such that  $r_b(\xi) = (\llbracket \mathcal{A}'_b \rrbracket(\xi))b = jb$ , and so  $\xi \in L_{b,j}$ .

Hence,

$$r_b = \bigoplus_{0 \leq j \leq i(b)+p(b)-1} (jb) \otimes \mathbb{1}_{(B, L_{b,j})}.$$

i.e., it is a recognizable step mapping.

Proof of 2: By Lemma 6.5, we can compute the set  $H(\mathcal{A})$ , and thus the set  $C(\mathcal{A})$ . Moreover, for each  $b \in C(\mathcal{A})$ , we can construct effectively the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}'_b$ .

Next we decide, for each  $b \in C(\mathcal{A})$ , whether the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order. Note that one of these conditions holds by our assumption.

For this, we run the following two algorithms in parallel for  $i = 0, 1, 2, \dots$ . In the first algorithm, we construct effectively the finite  $\Sigma$ -tree automaton which recognizes the  $\Sigma$ -tree language  $L_{b,i} = \llbracket \mathcal{A}'_b \rrbracket^{-1}(i)$  (cf. Lemma 6.3(2)) and check whether  $\bigcup_{j \in [0,i]} L_{b,j} = T_\Sigma$  (cf. [GS84, Thm. 2.10.3]). If this is the case, then it means that  $f_{\mathcal{A},b}$  is bounded by  $i$ . We let  $K_b = i$  and stop.

In the second algorithm, we compute the sum  $ib$  and check whether  $ib = jb$  for some  $j < i$ . If this is the case, then  $\langle b \rangle_\oplus = \{0, b, 2b, \dots, (i-1)b\}$  is a finite set. We let  $K_b = i-1$  and stop.

By our note above, the decision algorithm will stop for some  $i \in \mathbb{N}$ .

If the first algorithm stops, then we can describe  $r_b$  as in Case 1 of Statement 1. If the second algorithm stops, then we can compute  $i(b)$  and  $p(b)$  and can describe  $r_b$  as in Case 2 of Statement 1. By Lemmas 6.3(2) and 6.4(2), we can construct effectively a finite  $\Sigma$ -tree automaton which recognizes  $L_{b,j}$ . If both algorithms stop, then we can proceed in either way.

Now we have

$$\llbracket \mathcal{A} \rrbracket = \bigoplus_{b \in C(\mathcal{A})} \bigoplus_{0 \leq j \leq K_b} (jb) \otimes \mathbb{1}_{(B, L_{b,j})}.$$

By applying the direct product construction in the proof of (iv)  $\Rightarrow$  (i) of [FKV21, Lm. 5.3] we can construct effectively the crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ .  $\square$

Now we can give a simple condition on the strong bimonoid  $B$  and a structural condition on the wta  $\mathcal{A}$  ensuring that  $\mathcal{A}$  has the finite image property and the preimage property.

**Corollary 6.7.** Let  $B$  be commutative or one-product free and let small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ .

1. Then  $H(\mathcal{A})$  is finite.
2. If, in addition, for each  $b \in C(\mathcal{A})$ , the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order, then  $\mathcal{A}$  has the finite-image property and the preimage property.

*Proof.* First we prove Statement 1. If small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ , then by Lemma 5.5 we have

$$H(\mathcal{A}) = \{\text{wt}(\xi, \rho) \mid \xi \in T_\Sigma, \text{height}(\xi) < |Q| \text{ and } \rho \in R_{\mathcal{A}}(\xi)\} .$$

Hence  $H(\mathcal{A})$  is finite. Then Statement 2 follows from Theorem 6.6(1).  $\square$

Next, we compare Theorem 6.6 and [FKV21, Thm. 7.3]. This makes sense because both results show sufficient conditions for a wta  $\mathcal{A}$  such that  $\llbracket \mathcal{A} \rrbracket$  is a recognizable step mapping. Let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta. Theorem 7.3 of [FKV21] requires that  $\mathcal{A}$  has finite order property, i.e., (a) the set  $\langle \text{im}(\delta) \rangle_\otimes$  is finite and (b) each element  $b \in \langle \text{im}(\delta) \rangle_\otimes \otimes \text{im}(F)$  has finite additive order. Condition (a) implies that  $H(\mathcal{A})$  is finite. And Condition (b) implies that each element  $b \in C(\mathcal{A})$  has finite additive order. Hence Theorem 6.6 is at least as strong as [FKV21, Thm. 7.3]. The next example shows a scenario in which Theorem 6.6 is applicable but not [FKV21, Thm. 7.3].

**Example 6.8.** We consider the ranked alphabet  $\Sigma = \{\gamma^{(1)}, \nu^{(1)}, \alpha^{(0)}\}$  and the arctic semiring  $\text{ASR}_{\mathbb{N}} = (\mathbb{N}_{-\infty}, \max, +, -\infty, 0)$ . Moreover, we let  $\mathcal{A} = (Q, \delta, F)$  be the trim  $(\Sigma, \text{ASR}_{\mathbb{N}})$ -wta where  $Q = \{q_1, q_2\}$ ,  $\delta_0(\varepsilon, \alpha, q_1) = \delta_1(q_1, \gamma, q_1) = 0$ , and  $\delta_1(q_1, \nu, q_2) = 1$ ; and  $F(q_1) = F(q_2) = 0$ .

Since  $\langle \text{im}(\delta) \rangle_+ = \langle \{0, 1\} \rangle_+ = \mathbb{N}$  is infinite, we cannot apply [FKV21, Thm. 7.3]. Moreover, since  $\text{ASR}_{\mathbb{N}}$  is one-product free, small loops of  $\mathcal{A}$  have weight 0 and each  $n \in C(\mathcal{A})$  has finite additive order, by Corollary 6.7(2) we obtain that  $\llbracket \mathcal{A} \rrbracket$  is a recognizable step mapping.  $\square$

As a consequence of Theorem 6.6, we can extend Lemma 6.2 from finite semirings to finite strong bimonoids.

**Corollary 6.9.** Let  $B$  be finite and  $\mathcal{A}$  be a  $(\Sigma, B)$ -wta. Then the following statements hold.

1.  $\mathcal{A}$  has the preimage property.
2. If  $B$  is computable and  $\mathcal{A}$  is given effectively, then, for each  $b \in B$ , we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $\llbracket \mathcal{A} \rrbracket^{-1}(b)$ .

*Proof.* Let  $\mathcal{A} = (Q, \delta, F)$  and we abbreviate  $\llbracket \mathcal{A} \rrbracket$  by  $r$ . Since  $B$  is finite, so is the set  $H(\mathcal{A})$ . Moreover,  $b$  has finite additive order for each  $b \in C(\mathcal{A})$ . Then Statement 1 follows from Theorem 6.6(1).

Now we prove Statement 2. By Theorem 6.6(2), we can construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B} = (Q', \delta', F')$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ . Note that  $\text{im}(\llbracket \mathcal{A} \rrbracket) \subseteq \text{im}(F')$  because  $\mathcal{B}$  is crisp-deterministic. Then Statement 2 follows from the remark after Lemma 3.4.  $\square$

Finally, as an application of Corollary 6.9, we show that for arbitrary past-finite monotonic strong bimonoid  $B$ , every  $(\Sigma, B)$ -wta has the finite-image property. This generalizes the preimage property [BR88, III. Cor. 2.5] from strings to trees and from the semiring  $\mathbb{N}$  to past-finite monotonic strong bimonoids.



**Theorem 6.10.** *Let  $B$  be past-finite monotonic and  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta. Then the following statements hold.*

1.  $\mathcal{A}$  has the preimage property.
2. If  $B$  is computable,  $\mathcal{A}$  is given effectively, and  $b \in B$  such that the set  $\text{past}(b)$  is computable, then we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $\llbracket \mathcal{A} \rrbracket^{-1}(b)$ .

*Proof. Proof of 1:* Let  $b \in B$  and put  $C = B \setminus \text{past}(b) = \{a \in B \mid a \not\leq b\}$ . Moreover, let  $\sim$  be the equivalence relation on the set  $B$  defined such that its classes are the singleton sets  $\{a\}$  for each  $a \in \text{past}(b)$  and the set  $C$ . We claim that  $\sim$  is a congruence. To show that  $C$  is a congruence class, let  $c, c' \in C$  and  $d \in B$ . Since  $B$  is monotonic, we have  $c \preceq c \oplus d$ , hence  $c \oplus d \in C$  and similarly  $c' \oplus d \in C$ . Also, if  $d \neq 0$ , again we obtain  $c \preceq c \otimes d$  and  $c \preceq d \otimes c$ , showing  $c \otimes d, d \otimes c \in C$  and similarly  $c' \otimes d, d \otimes c' \in C$ . Hence  $C$  is a congruence class and the relation  $\sim$  is a congruence on the strong bimonoid  $B$ . Then the quotient strong bimonoid  $B/\sim$  is finite. Let  $h : B \rightarrow B/\sim$  be the canonical strong bimonoid homomorphism. Let us abbreviate  $\llbracket \mathcal{A} \rrbracket$  by  $r$ . Then, by Lemma 3.3,  $(h \circ r) \in \text{Rec}(\Sigma, B/\sim)$ . Moreover  $r^{-1}(b) = (h \circ r)^{-1}(\{b\})$ . Since  $B/\sim$  is finite, by Corollary 6.9(1), the  $\Sigma$ -tree language  $(h \circ r)^{-1}(\{b\})$  is recognizable.

*Proof of 2:* Let  $\sim$  be the congruence defined as in the proof of Statement 1 and  $h : B \rightarrow B/\sim$  be the canonical strong bimonoid homomorphism. Since  $B$  is computable and also  $\text{past}(b)$  is computable, we can give effectively the congruence classes of  $\sim$ , i.e., the elements of  $B/\sim$ , by choosing only one representative for each congruence class. By Lemma 3.3, we can construct effectively the  $(\Sigma, B/\sim)$ -wta  $h(\mathcal{A})$  such that  $\llbracket h(\mathcal{A}) \rrbracket = h \circ r$ . Since  $B/\sim$  is computable and finite, and  $\mathcal{A}$  is given effectively, by Corollary 6.9(2), we can construct effectively a finite-state  $\Sigma$ -tree automaton which recognizes  $(h \circ r)^{-1}(\{b\})$ .  $\square$

**Corollary 6.11.** *Let  $B$  be past-finite monotonic and  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta. Then  $\text{supp}(\llbracket \mathcal{A} \rrbracket)$  is a recognizable  $\Sigma$ -tree language.*

*Proof.* By Theorem 6.10, the  $\Sigma$ -tree language  $\llbracket \mathcal{A} \rrbracket^{-1}(0)$  is recognizable. Since the class of recognizable  $\Sigma$ -tree languages is closed under complement and  $\text{supp}(\llbracket \mathcal{A} \rrbracket) = T_\Sigma \setminus \llbracket \mathcal{A} \rrbracket^{-1}(0)$ , we obtain the statement of the corollary.  $\square$

## 7 Characterization and decidability of the finite-image property

In most of this section,  $B$  will be a past-finite strong bimonoid. Our main results of this section will describe when a  $(\Sigma, B)$ -wta  $\mathcal{A}$  has the finite-image property and when it can be decided whether  $\mathcal{A}$  has the finite-image property. First we characterize when an arbitrary  $(\Sigma, B)$ -wta  $\mathcal{A}$  has the finite-image property by structural properties of the wta  $\mathcal{A}$ .

**Theorem 7.1.** *Let  $B$  be a past-finite monotonic strong bimonoid and  $\mathcal{A}$  be a trim  $(\Sigma, B)$ -wta. Then the following statements are equivalent.*

1.  $\mathcal{A}$  has the finite-image property.
2. Small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$  and, for each  $b \in C(\mathcal{A})$ , the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order.

*Proof.* (1)  $\Rightarrow$  (2): First we show that small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ . We proceed by contraposition. Suppose there exist  $q \in Q$ ,  $c \in C_\Sigma$ , and  $\rho \in R_{\mathcal{A}}(q, c, q)$  such that  $\text{height}(c) < |Q|$  and  $\mathbb{1} \prec \text{wt}(c, \rho)$ . Since  $\mathcal{A}$  is trim, the state  $q$  is useful and thus there exist  $\xi \in T_\Sigma, \theta \in R_{\mathcal{A}}(q, \xi)$  and  $c' \in C_\Sigma, q' \in Q$  with  $F_{q'} \neq 0$ , and  $\rho' \in R_{\mathcal{A}}(q', c', q)$ . By Lemma 5.3, for each  $n \in \mathbb{N}$ , we have

$$\text{wt}(c' [c^n[\xi]], \rho' [\rho^n[\theta]]) = l_{c', \rho'} \otimes (l_{c, \rho})^n \otimes \text{wt}(\xi, \theta) \otimes (r_{c, \rho})^n \otimes r_{c', \rho'} .$$

Since  $\mathbb{1} \prec \text{wt}(c, \rho) = l_{c, \rho} \otimes r_{c, \rho}$ , we have  $\mathbb{1} \prec l_{c, \rho}$  or  $\mathbb{1} \prec r_{c, \rho}$  and thus by monotonicity we obtain

$$\text{wt}(c' [c^0[\xi]], \rho' [\rho^0[\theta]]) \prec \text{wt}(c' [c^1[\xi]], \rho' [\rho^1[\theta]]) \prec \dots . \quad (4)$$

We define a sequence  $\xi_1, \xi_2, \xi_3, \dots$  of trees in  $T_\Sigma$  such that the elements  $\llbracket \mathcal{A} \rrbracket(\xi_1), \llbracket \mathcal{A} \rrbracket(\xi_2), \llbracket \mathcal{A} \rrbracket(\xi_3), \dots$  are pairwise different as follows. We let  $\xi_1 = c' [c[\xi]]$ . Then  $P_1 = \text{past}(\llbracket \mathcal{A} \rrbracket(\xi_1))$  is finite. By (4) we choose  $n_2$  such that  $\text{wt}(c' [c^{n_2}[\xi]], \rho' [\rho^{n_2}[\theta]]) \notin P_1$  and let  $\xi_2 = c' [c^{n_2}[\xi]]$ . Since  $\rho' [\rho^{n_2}[\theta]] \in R_{\mathcal{A}}(q', \xi_2)$  and  $B$  is monotonic, we have

$$\text{wt}(\xi_2, \rho' [\rho^{n_2}[\theta]]) \preceq \text{wt}(\xi_2, \rho' [\rho^{n_2}[\theta]]) \otimes F_{q'} \preceq \llbracket \mathcal{A} \rrbracket(\xi_2).$$

(Note that  $F_{q'}$  may be  $\mathbb{1}$ .) Hence  $\llbracket \mathcal{A} \rrbracket(\xi_2) \notin P_1$ . Put  $P_2 = \text{past}(\llbracket \mathcal{A} \rrbracket(\xi_2))$ . Then we choose  $n_3 \in \mathbb{N}$  such that  $\text{wt}(c' [c^{n_3}[\xi]], \rho' [\rho^{n_3}[\theta]]) \notin P_1 \cup P_2$  and let  $\xi_3 = c' [c^{n_3}[\xi]]$ . As before, we have  $\llbracket \mathcal{A} \rrbracket(\xi_3) \notin P_1 \cup P_2$ . Continuing this process, we obtain the desired sequence of trees. It means that  $\mathcal{A}$  does not have the finite-image property.

Now let  $b \in C(\mathcal{A})$ . If the mapping  $f_{\mathcal{A},b}$  is not bounded, then there exists an infinite sequence  $\xi_1, \xi_2, \dots$  of trees in  $T_\Sigma$  such that  $f_{\mathcal{A},b}(\xi_1) < f_{\mathcal{A},b}(\xi_2) < \dots$ . By Equality (3), we have  $(f_{\mathcal{A},b}(\xi_i))b \preceq \llbracket \mathcal{A} \rrbracket(\xi_i)$  for each  $i \in \mathbb{N}$ . Thus  $(f_{\mathcal{A},b}(\xi_i))b \in P$ , where  $P = \bigcup_{a \in \text{im}(\llbracket \mathcal{A} \rrbracket)} \text{past}(a)$ . Since  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is finite and  $B$  is past-finite, the set  $P$  is also finite. Hence  $(f_{\mathcal{A},b}(\xi_i))b = (f_{\mathcal{A},b}(\xi_j))b$  for some  $i, j \in \mathbb{N}$  with  $i < j$ , which implies that  $b$  has finite additive order.

(2)  $\Rightarrow$  (1): It follows from Corollary 6.7(2).  $\square$

The following example shows that in Theorem 7.1 even for commutative semirings we cannot replace the assumption that  $B$  is past-finite by being idempotent. The wta  $\mathcal{A}$  given below can actually be considered as a weighted string automaton (cf. Section 10).

**Example 7.2.** In this example we give an idempotent and monotonic semiring  $B$  which is not past-finite, and a  $(\Sigma, B)$ -wta  $\mathcal{A}$  such that  $\mathcal{A}$  has the finite-image property.

For this, let  $B = \mathbb{N}_+ \cup \mathbb{N}'_+ \cup \{0, \mathbb{1}\}$  where  $\mathbb{N}'_+$  is an isomorphic copy of  $\mathbb{N}_+$ , and furthermore,  $0$  and  $\mathbb{1}$  are new elements such that  $\{0, \mathbb{1}\} \cap (\mathbb{N}_+ \cup \mathbb{N}'_+) = \emptyset$ .

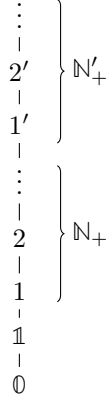


Figure 4: The Hasse diagram of the ordering  $(B, \leq_B)$  in Example 7.2.

For each  $C \in \{\mathbb{N}_+, \mathbb{N}'_+\}$  we denote by  $\leq_C$  the usual linear order of  $C$  and  $+_C$  the usual binary addition operation on  $C$ .

We define a partial ordering  $\leq_B$  on  $B$  such that  $\leq_B$  extends each of the linear orders of  $\mathbb{N}_+$  and  $\mathbb{N}'_+$  (i.e.,  $\leq_B \cap (C \times C) = \leq_C$  for each  $C \in \{\mathbb{N}_+, \mathbb{N}'_+\}$ ) and such that  $0 <_B \mathbb{1} <_B a <_B b'$  for every  $a \in \mathbb{N}_+$  and  $b' \in \mathbb{N}'_+$ . Note that  $(B, \leq_B)$  is a partial order. Figure 4 shows the Hasse diagram of the ordering  $(B, \leq_B)$ . Moreover, letting  $\vee$  be the usual binary supremum operation,  $(B, \vee)$  is a join-semilattice. Note that we have  $a \vee b' = b'$  for each  $a \in \mathbb{N}_+$  and  $b' \in \mathbb{N}'_+$ .

Now we define a commutative multiplication  $\otimes$  on  $B$  as follows. For each  $C \in \{\mathbb{N}_+, \mathbb{N}'_+\}$  we let  $\otimes|_{C \times C} = +_C$ . Furthermore, for every  $a \in \mathbb{N}_+$  and  $b' \in \mathbb{N}'_+$  we let  $a \otimes b' = a' +_{\mathbb{N}'_+} b'$ . Finally, let  $0 \otimes b = 0$  and  $\mathbb{1} \otimes b = b$  for each  $b \in B$ . Then  $(B, \vee, \otimes, 0, \mathbb{1})$  is a monotonic strong bimonoid. Clearly,  $B$  is not past-finite, e.g.,  $\text{past}(\mathbb{1}') = (\{0, \mathbb{1}\} \cup \mathbb{N}_+)$ , which is infinite. However,  $B$  is idempotent because it is a join-semilattice. Obviously,  $B$  is monotonic.

Let  $\Sigma = \{\gamma^{(1)}, e^{(0)}\}$ . Next we consider the  $(\Sigma, B)$ -wta  $\mathcal{A} = (\{p, q\}, \delta, F)$  with  $\delta_0(\varepsilon, e, p) = \mathbb{1}$ ,  $\delta_0(\varepsilon, e, q) = 1'$  (in  $\mathbb{N}'_+$ ),  $\delta_1(p, \gamma, p) = 1$  (in  $\mathbb{N}_+$ ),  $\delta_1(q, \gamma, q) = \mathbb{1}$ ,  $\delta_1(p, \gamma, q) = 0 = \delta_1(q, \gamma, p)$ , and  $F_p = \mathbb{1} = F_q$ . Let  $n \in \mathbb{N}$  and  $\xi = \gamma^n(e) \in \text{T}_\Sigma$ . Clearly, there exist two runs on  $\xi$ : Let us denote them by  $\rho^{\xi, p}$  and  $\rho^{\xi, q}$ . Obviously,  $\text{wt}(\xi, \rho^{\xi, p}) = n$  if  $n \geq 1$ , and otherwise  $\text{wt}(\xi, \rho^{\xi, p}) = \mathbb{1}$ , and furthermore,  $\text{wt}(\xi, \rho^{\xi, q}) = 1'$ . Thus, we have  $[\mathcal{A}](\xi) = \text{wt}(\xi, \rho^{\xi, p}) \vee \text{wt}(\xi, \rho^{\xi, q}) = 1'$ . Hence,  $\mathcal{A}$  has the finite-image property.

Observe that  $\mathcal{A}$  has two small loops with weight 1 (in  $\mathbb{N}_+$ ) respectively  $\mathbb{1}$ .  $\square$

We say that the strong bimonoid  $B$  has *effective tests for 0 and 1* if for each  $b \in B$  we can decide whether  $b = 0$  and whether  $b = \mathbb{1}$ .

**Theorem 7.3.** [DFKV20, Cor. 14] *Let  $B$  be past-finite monotonic and have effective tests for 0 and 1. Moreover, let  $\mathcal{A}$  be given effectively. If  $B$  is additively locally finite or  $\mathcal{A}$  is finitely ambiguous, then it is decidable whether  $\mathcal{A}$  has the finite-image property.*

*Proof.* By Theorem 4.1, we may assume that  $\mathcal{A}$  is trim. By Theorem 7.1,  $\mathcal{A}$  has the finite-image property if and only if small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ . The latter property is decidable because (a) there exist only finitely many  $c \in C_\Sigma$  such that  $\text{height}(c) < |Q|$ , and (b) since  $B$  is monotonic, for all  $c \in C_\Sigma$ ,  $q \in Q$ , and  $\rho \in R_{\mathcal{A}}(q, c, q)$  we have  $\text{wt}(c, \rho) = \mathbb{1}$  if and only if for each  $v \in \text{pos}(c)$  we have  $\delta_k(\rho(v1) \cdots \rho(vk), \sigma, \rho(v)) = \mathbb{1}$  where  $\sigma = c(v)$  and  $k = \text{rk}(\sigma)$ , and (c) this is decidable because  $B$  has an effective test for  $\mathbb{1}$ .  $\square$

The decidability problem addressed in Theorem 7.3 is meaningful, because in Example 3.2 we considered the additively locally finite and past-finite monotonic semiring  $\text{ASR}_{\mathbb{N}}$  and a deterministic  $(\Sigma, \text{ASR}_{\mathbb{N}})$ -wta  $\mathcal{A}$  for which  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite.

As an immediate consequence of Theorems 6.10(1) and 7.3 and Lemma 3.4, under the assumptions of Theorem 7.3, it is decidable whether  $\mathcal{A}$  is crisp-determinizable.

Next, we compare Theorem 6.1 with Theorem 7.3 in the following sense: we show an example of a wta  $\mathcal{A}$  such that (a) by applying Theorem 6.1 we know that  $\mathcal{A}$  has the finite-image property and (b) we cannot apply Theorem 7.3 to decide whether  $\mathcal{A}$  has the finite-image property.

**Example 7.4.** We consider the ranked alphabet  $\Sigma = \{\gamma^{(1)}, \nu^{(1)}, \alpha^{(0)}\}$  and the tropical semiring  $\text{TSR}_{\mathbb{N}} = (\mathbb{N}_{\infty}, \min, +, \infty, 0)$ . Moreover, we let  $\mathcal{A} = (Q, \delta, F)$  be the trim  $(\Sigma, \text{TSR}_{\mathbb{N}})$ -wta (as in Example 6.8) where  $Q = \{q_1, q_2\}$ ,  $\delta_0(\varepsilon, \alpha, q_1) = \delta_1(q_1, \gamma, q_1) = 0$ , and  $\delta_1(q_1, \nu, q_2) = 1$ ; and  $F_{q_1} = F_{q_2} = 0$ .

Then  $\mathcal{A}$  satisfies the assumptions of Theorem 6.1. In fact,  $\text{im}(\llbracket \mathcal{A} \rrbracket) = \{0, 1\}$ .

The tropical semiring  $\text{TSR}_{\mathbb{N}}$  cannot be extended into a past-finite monotonic semiring  $(\mathbb{N}_{\infty}, \min, +, \infty, 0, \preceq)$ . To see this, assume that  $\preceq$  is a monotonic partial order. Then  $a \preceq \min\{a, b\}$  for every  $a, b \in \mathbb{N}_{\infty}$  (by Condition (i) of monotonicity). Thus  $a \geq b$  (i.e.,  $b = \min\{a, b\}$ ) implies  $a \preceq b$ . However,  $\geq$  is not past-finite. Hence, we cannot use Theorem 7.3 to decide whether  $\mathcal{A}$  has the finite-image property.  $\square$

**Theorem 7.5.** *Let  $B$  be monotonic and have effective tests for  $\mathbb{0}$  and  $\mathbb{1}$ . It is decidable, for arbitrary unambiguous  $(\Sigma, B)$ -wta  $\mathcal{A}$ , whether  $\mathcal{A}$  has the finite-image property.*

*Proof.* By Theorem 4.1, we may assume that  $\mathcal{A}$  is trim. Then, by the proof of Theorem 7.3, we can decide whether small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ .

If this is not the case, then we follow the proof (1  $\Rightarrow$  2) of Theorem 7.1 up to (4), and as there, we can produce an infinite set of weights of runs on trees. Due to unambiguity, these are also the weights of the corresponding trees. Thus,  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite, i.e.,  $\mathcal{A}$  does not have the finite-image property.

Otherwise, by Corollary 6.7,  $\mathcal{A}$  has the finite-image property.  $\square$

As a side-result, we show that the well-known decidability of finiteness of context-free languages [Har78, Thm. 8.2.2] can be formally derived from Theorem 7.3.

**Corollary 7.6.** The finiteness of context-free languages is decidable.

*Proof.* Let  $G = (N, \Delta, P, S)$  be a context-free grammar. By [Har78, Thm. 3.2.3] we may assume that  $G$  is reduced. Then we have

$$\text{for each set } U \subseteq \Delta^*: U \text{ is finite if and only if } \{|w| \mid w \in U\} \text{ is finite.} \quad (5)$$

We construct effectively the ranked alphabet  $(P, \text{rk})$  such that if  $p \in P$  has the form

$$A \rightarrow w_0 A_1 w_1 \cdots A_k w_k$$

for some  $k \in \mathbb{N}$ ,  $w_0, w_1, \dots, w_k \in \Delta^*$ , and  $A, A_1, \dots, A_k \in N$ , then  $\text{rk}(p) = k$ .

Finally, we consider the arctic semiring  $\text{ASR}_{\mathbb{N}} = (\mathbb{N}_{-\infty}, \max, +, -\infty, 0)$  and construct effectively the  $(P, \text{ASR}_{\mathbb{N}})$ -wta  $A(G) = (N, \delta, F)$  by  $F_S = 0$  and  $F_A = -\infty$  for each  $A \in N \setminus \{S\}$  and if  $p \in P$  has the form as above, then

$$\delta_k(A_1 \cdots A_k, p, A) = |w_0 w_1 \cdots w_k| .$$

Using (5), it is easy to see that

$$\text{im}(\llbracket A(G) \rrbracket) \text{ is finite if and only if } L(G) \text{ is finite.} \quad (6)$$

Since  $\text{ASR}_{\mathbb{N}}$  is past-finite monotonic and idempotent, and  $A(G)$  is unambiguous and trim, we obtain from (6) and Theorem 7.3 that it is decidable whether  $L(G)$  is finite or not.  $\square$

To prove that we can decide whether an arbitrary  $(\Sigma, \mathbb{N})$ -wta has the finite-image property (cf. Theorem 7.11), we need the following preparation.

To each  $(\Sigma, B)$ -wta  $\mathcal{A} = (Q, \delta, F)$  we associate the  $(\Sigma, \mathbb{B})$ -wta  $\text{fta}(\mathcal{A}) = (Q, \delta_{\mathbb{B}}, F_{\mathbb{B}})$  defined as follows:

- for every  $k \in \mathbb{N}$ ,  $w \in Q^k$ ,  $\sigma \in \Sigma^{(k)}$ , and  $q \in Q$ , let  $(\delta_{\mathbb{B}})_k(w, \sigma, q) = 1$  if and only if  $\delta_k(w, \sigma, q) \neq \emptyset$ , and
- for each  $q \in Q$ , let  $(F_{\mathbb{B}})_q = 1$  if and only if  $F_q \neq \emptyset$ .

**Observation 7.7.** For each  $\xi \in T_{\Sigma}$ , we have  $|\mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)| = |\mathbb{R}_{\text{fta}(\mathcal{A})}^{F_{\mathbb{B}} \neq 0}(\xi)|$ .

**Lemma 7.8.** Let  $\mathcal{A}$  be a trim  $(\Sigma, \mathbb{N})$ -wta such that all small loops of  $\mathcal{A}$  have weight 1. Then  $\mathcal{A}$  has the finite-image property if and only if  $\text{fta}(\mathcal{A})$  is finitely ambiguous.

*Proof.* By Corollary 6.7(1), the set  $H(\mathcal{A})$  is finite, and thus  $C(\mathcal{A})$  is also finite.

Let  $C = \max\{b \mid b \in C(\mathcal{A})\}$ , i.e.,  $C$  is the maximum of all possible complete run weights of  $\mathcal{A}$ .

First assume that  $\text{fta}(\mathcal{A})$  is finitely ambiguous under the uniform bound  $K \in \mathbb{N}$  and let  $\xi \in T_{\Sigma}$ . Then, by Observation 7.7,  $|\mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)| \leq K$ , and thus, we have

$$\llbracket \mathcal{A} \rrbracket(\xi) = \sum_{\rho \in \mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)} \text{wt}(\xi) \cdot F_{\rho}(\varepsilon) \leq \sum_{\rho \in \mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)} C \leq K \cdot C.$$

Hence,  $\mathcal{A}$  has the finite-image property.

Now we prove the other direction. For this, assume that  $\mathcal{A}$  has the finite-image property. Let  $K = \max\{\llbracket \mathcal{A} \rrbracket(\xi) \mid \xi \in \mathbb{T}_\Sigma\}$ . Moreover, let  $\xi \in \mathbb{T}_\Sigma$ . Then

$$K \geq \llbracket \mathcal{A} \rrbracket(\xi) = \sum_{\rho \in \mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)} \text{wt}(\xi) \cdot F_\rho(\varepsilon) \geq \sum_{\rho \in \mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)} 1 = |\mathbb{R}_{\mathcal{A}}^{F \neq 0}(\xi)| ,$$

and Observation 7.7 shows that  $\text{fta}(\mathcal{A})$  is finitely ambiguous.  $\square$

**Lemma 7.9.** Let  $B$  have an effective test for  $\emptyset$ . It is decidable, for arbitrary  $(\Sigma, B)$ -wta  $\mathcal{A} = (Q, \delta, F)$  given effectively, whether  $\mathcal{A}$  has a useful state.

*Proof.* In fact,  $\mathcal{A}$  has a useful state if and only if there exists a state  $q \in Q$  with  $F_q \neq \emptyset$  for which there exist a tree  $\xi \in \mathbb{T}_\Sigma$  and a run  $\rho \in \mathbb{R}_{\mathcal{A}}(q, \xi)$ . By standard pumping arguments, the latter is the case if and only if there exist a tree  $\xi'$  with  $\text{height}(\xi') < |Q|$  and run  $\rho' \in \mathbb{R}_{\mathcal{A}}(q, \xi')$ . This property is easily decidable.  $\square$

The following result will be crucial for Theorem 7.11.

**Theorem 7.10.** [Sei89, Thm 2.5(2)] Let  $\mathcal{A}$  be a  $(\Sigma, \mathbb{B})$ -wta. We can decide whether  $\mathcal{A}$  is finitely ambiguous.

**Theorem 7.11.** It is decidable, for arbitrary  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}$ , whether  $\mathcal{A}$  has the finite-image property.

*Proof.* First, by Lemma 7.9, we decide whether  $\mathcal{A}$  has a useful state. If  $\mathcal{A}$  does not have a useful state, then  $\llbracket \mathcal{A} \rrbracket = \tilde{\emptyset}$  and thus  $\llbracket \mathcal{A} \rrbracket$  has the finite-image property. Otherwise, by Theorem 4.1 we may assume that  $\mathcal{A}$  is trim.

By the proof of Theorem 7.3 we can decide whether small loops of  $\mathcal{A}$  have weight 1. If the answer is no, then  $\mathcal{A}$  has a loop  $\rho$  on a context  $c$  with  $1 < \text{wt}(c, \rho)$ . Then, by Theorem 7.1,  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite, i.e.,  $\mathcal{A}$  does not have the finite-image property. (Note that  $\mathbb{N}$  is past-finite monotonic.)

Otherwise, all small loops of  $\mathcal{A}$  have weight 1. Then by Lemma 7.8,  $\mathcal{A}$  has the finite-image property if and only if  $\text{fta}(\mathcal{A})$  is finitely ambiguous. Finally, by Theorem 7.10, we can decide whether  $\text{fta}(\mathcal{A})$  is finitely ambiguous. This completes the proof.  $\square$

Next we can characterize those past-finite monotonic computable strong bi-monoids  $B$  for which for each  $(\Sigma, B)$ -wta  $\mathcal{A}$  it is decidable whether  $\mathcal{A}$  has the finite-image property.

**Theorem 7.12.** Let  $B$  be past-finite monotonic and computable. Then the following statements are equivalent:

1. It is decidable, for each ranked alphabet  $\Sigma$ , whether an arbitrary  $(\Sigma, B)$ -wta given effectively has the finite-image property.
2. It is decidable, for each  $b \in B$ , whether  $b$  has finite additive order.

*Proof.* (1  $\Rightarrow$  2). Let  $b \in B$ . We construct the  $(\{\gamma^{(1)}, \alpha^{(0)}\}, B)$ -wta  $\mathcal{A} = (\{p, q\}, \delta, F)$  such that

- $\delta_1(p, \gamma, q) = b$ ,  $\delta_0(\varepsilon, \alpha, p) = \delta_1(p, \gamma, p) = \delta_1(q, \gamma, q) = F_q = 1$ , and

- $\delta_0(\varepsilon, \alpha, q) = \delta_1(q, \gamma, p) = F_p = 0$ .

Clearly, for each  $n \in \mathbb{N}$ , we have  $\llbracket \mathcal{A} \rrbracket(\gamma^n \alpha) = nb$  and thus we have  $\text{im}(\llbracket \mathcal{A} \rrbracket) = \langle b \rangle_{\oplus}$ .

Therefore  $\mathcal{A}$  has the finite-image property if and only if  $b$  has finite additive order. By Statement 1, the former is decidable. Hence, it is decidable whether  $b$  has finite additive order.

(2  $\Rightarrow$  1). Let  $\mathcal{A}$  be an arbitrary  $(\Sigma, B)$ -wta given effectively. By Theorem 4.1, we may assume that  $\mathcal{A}$  is trim. By the proof of Theorem 7.3 we can decide whether small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ . If the answer is no, then by Theorem 7.1,  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite, i.e.,  $\mathcal{A}$  does not have the finite-image property. Otherwise, all small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$ . Then, by Corollary 6.7(1), the set  $H(\mathcal{A})$  is finite, which implies that the set  $C(\mathcal{A})$  is also finite. In addition, by Lemma 6.5 and the fact that  $B$  is computable, we can compute the set  $C(\mathcal{A})$ . Due to our assumption on small loops of  $\mathcal{A}$  and by Theorem 7.1,  $\mathcal{A}$  has the finite-image property if and only if, for each  $b \in C(\mathcal{A})$ , the mapping  $f_{\mathcal{A}, b}$  is bounded or  $b$  has finite additive order. Since  $B$  is computable, by the proof of Theorem 6.6, we can construct effectively, for each  $b \in C(\mathcal{A})$ , the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}'_b$  described in the proof of Theorem 6.6. By Theorem 7.11, it is decidable, for each  $b \in C(\mathcal{A})$ , whether  $\llbracket \mathcal{A}'_b \rrbracket$  has finite image. Moreover, by Statement 2, it is decidable, for each  $b \in C(\mathcal{A})$ , whether  $b$  has finite additive order. Therefore, it is decidable, whether  $\mathcal{A}$  has the finite-image property.  $\square$

**Lemma 7.13.** Let  $B$  be a monotonic strong bimonoid such that it is left or right distributive. Then either  $B$  is idempotent or else each  $b \in B \setminus \{0\}$  has infinite additive order.

*Proof.* We may assume that  $B$  is left distributive. Clearly, if  $\mathbb{1} = \mathbb{1} \oplus \mathbb{1}$ , then  $B$  is idempotent. Therefore, assume that  $\mathbb{1} \neq \mathbb{1} \oplus \mathbb{1}$ , and thus,  $\mathbb{1} \prec \mathbb{1} \oplus \mathbb{1}$ . Since  $B$  is monotonic and left distributive, for each  $b \in B \setminus \{0\}$ , we have  $b \prec b \otimes (\mathbb{1} \oplus \mathbb{1}) = b \oplus b$ . Consequently,  $b \oplus b \prec (b \oplus b) \oplus (b \oplus b) = 4b$ . Thus, for each  $n \in \mathbb{N}_+$ , we have  $b \prec b \oplus b \prec \dots \prec (2^n)b$ . Hence,  $\langle b \rangle_{\oplus}$  is infinite. (Observe that the proof is similar if we assume right distributivity instead of left distributivity.)  $\square$

**Theorem 7.14.** Let  $B$  be a past-finite monotonic strong bimonoid and computable such that it is left or right distributive. Then it is decidable, for each  $(\Sigma, B)$ -wta  $\mathcal{A}$  given effectively, whether  $\mathcal{A}$  has the finite-image property. In particular, this is decidable if  $B$  is a past-finite monotonic and computable semiring.

*Proof.* Let  $\mathcal{A}$  be an arbitrary  $(\Sigma, B)$ -wta given effectively. We first check whether  $\mathbb{1} = \mathbb{1} \oplus \mathbb{1}$ . If this is the case, then  $B$  is idempotent, and thus, by Theorem 7.12, it is decidable whether  $\mathcal{A}$  has the finite-image property. Otherwise, by Lemma 7.13, each  $b \in B \setminus \{0\}$  has infinite additive order. Again, by Theorem 7.12, it is decidable whether  $\mathcal{A}$  has the finite-image property.  $\square$

We give examples illustrating applications of Theorem 7.12.

**Example 7.15.**

1. Let  $B$  be an additively locally finite strong bimonoid. In this case Theorem 7.12(2) is satisfied. This is the situation of Theorem 7.3. Note, however, Theorem 7.3 does not require  $B$  to be computable.
2. Let  $B$  be a strong bimonoid such that, for each  $b \in (B \setminus \{0\})$ , the set  $\langle b \rangle_{\oplus}$  is not finite, e.g., the semiring  $\mathbb{N}$  of natural numbers. In this case Theorem 7.12(2) is also satisfied. This situation can be viewed as orthogonal to the situation of Theorem 7.3.
3. Each semiring  $B$  belongs to either Item 1 or Item 2, because by Lemma 7.13, it suffices to decide whether  $\mathbb{1} = \mathbb{1} \oplus \mathbb{1}$ .
4. The following strong bimonoid combines features of Items 1 and 2, i.e., it contains elements both which have, respectively, which do not have finite additive order, but this is decidable for each element. Let  $B = (\mathbb{N}, \oplus, \cdot, 0, 1, \leq)$ , where, for each  $x, y \in \mathbb{N}$ , we let

$$x \oplus y = \begin{cases} \max\{x, y\} & \text{if } x \in \{0, 1, 2\} \text{ or } y \in \{0, 1, 2\} \\ x + y & \text{otherwise ;} \end{cases}$$

in addition,  $\cdot$  is the usual multiplication, and  $\leq$  is the usual order on  $\mathbb{N}$ . Then, for each  $b \in B$ , we can test whether  $b \in \{0, 1, 2\}$  to decide whether  $b$  has finite additive order. Thus, Theorem 7.12(2) is satisfied.  $\square$

## 8 Further decidability results for wta over past-finite monotonic strong bimonoids

In all of this section,  $B$  will be a past-finite monotonic and computable strong bimonoid. We recall that, by Theorem 6.10(1), a  $(\Sigma, B)$ -wta  $\mathcal{A}$  always has the preimage property. Hence, by Lemma 3.4, the  $(\Sigma, B)$ -wta  $\mathcal{A}$  has the finite-image property iff  $\llbracket \mathcal{A} \rrbracket$  is a recognizable step mapping iff  $\mathcal{A}$  is crisp-determinizable. Hence, the question arises whether in case  $\mathcal{A}$  has the finite image property, and thus, is crisp-determinizable, we can actually construct an equivalent crisp-deterministic wta. This will be shown to be true in the following result. Note that then, from the equivalent crisp-deterministic wta, we also know the elements of the finite set  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  as well as tree automata for the recognizable tree languages  $\llbracket \mathcal{A} \rrbracket^{-1}(b)$  (for all  $b \in \text{im}(\llbracket \mathcal{A} \rrbracket)$ ).

**Theorem 8.1.** *Let  $B$  be past-finite monotonic and computable. Moreover, let  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta given effectively. If  $\mathcal{A}$  has the finite-image property, then we can construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ .*

*Proof.* By Lemma 7.9, it is decidable whether  $\mathcal{A}$  has a useful state. If  $\mathcal{A}$  does not have a useful state, then  $\llbracket \mathcal{A} \rrbracket = \tilde{\emptyset}$  and we can construct effectively the desired crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  in an easy way.

Otherwise, by Theorem 4.1 we may assume that  $\mathcal{A}$  is trim. Then, by Theorem 7.1 and Corollary 6.7(1),  $\mathcal{A}$  satisfies the assumption of Theorem 6.6. Thus



we can use Theorem 6.6(2) to construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ .  $\square$

In the rest of this section we will show that if  $B$  is past-finite monotonic and computable, then we can decide, for every wta  $\mathcal{A}$  given effectively and positive integer  $k$ , whether the cardinality of the image of  $\llbracket \mathcal{A} \rrbracket$  is bounded by  $k$ , and in this case we can construct effectively a crisp-deterministic wta which is equivalent to  $\mathcal{A}$ .

**Theorem 8.2.** *Let  $B$  be past-finite monotonic and computable. It is decidable, for every  $(\Sigma, B)$ -wta  $\mathcal{A}$  given effectively and  $k \in \mathbb{N}_+$ , whether we have  $|\text{im}(\llbracket \mathcal{A} \rrbracket)| \leq k$ . Moreover, in this case we can construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ .*

*Proof.* First, by the proof of Lemma 7.9, we check whether  $\mathcal{A}$  has a useful state. If  $\mathcal{A}$  has no useful state, we have  $\llbracket \mathcal{A} \rrbracket = \bar{0}$ , hence our statement holds obviously. Therefore we may assume that  $\mathcal{A}$  has a useful state. By Theorem 4.1, we may assume that  $\mathcal{A}$  is trim.

Next we check whether all small loops of  $\mathcal{A}$  have weight  $\mathbb{1}$  (cf. proof of Theorem 7.3). If this is not the case, then by Theorem 7.1,  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite and hence  $\mathcal{A}$  is not crisp-determinizable. Otherwise,  $H(\mathcal{A})$  is finite by Corollary 6.7(1) and thus  $C(\mathcal{A})$  is also finite. (We recall that each monotonic strong bimonoid is one-product free.) Using Lemma 6.5, we compute  $C(\mathcal{A})$ .

We choose an effective enumeration  $\xi_0, \xi_1, \dots$  of  $T_\Sigma$ . Then we run the following Algorithms A and B in parallel.

Algorithm A: Compute  $\llbracket \mathcal{A} \rrbracket(\xi_0), \llbracket \mathcal{A} \rrbracket(\xi_1), \dots$ . We let this algorithm terminate if we have obtained more than  $k$  different values.

Algorithm B: For each  $b \in C(\mathcal{A})$ , we run the following subalgorithm  $\text{Alg}(b)$ .  $\text{Alg}(b)$ : We compute the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}'_b$  as in the proof of Theorem 6.6. Now we proceed as in the algorithm described in the proof of Theorem 6.6(2). Successively for  $i = 0, 1, \dots$ ,

- (a) we compute and store a finite-state  $\Sigma$ -tree automaton for the language  $L_{b,i} = \llbracket \mathcal{A}'_b \rrbracket^{-1}(i)$  (cf. Lemma 6.3(2)), and
- (b) we compute and store the element  $ib$ .

We let  $\text{Alg}(b)$  terminate, if for some  $i \in \mathbb{N}$ , we have  $\bigcup_{j \in [0,i]} L_{b,j} = T_\Sigma$  or  $ib = jb$  for some  $j < i$ . Clearly, both equalities just given are decidable. We let Algorithm B terminate if, for each  $b \in C(\mathcal{A})$ , the algorithm  $\text{Alg}(b)$  terminates.

Next we show that this decision procedure terminates and thus we can decide whether  $|\text{im}(\llbracket \mathcal{A} \rrbracket)| \leq k$ . Clearly, if Algorithm A terminates, we have  $k < |\text{im}(\llbracket \mathcal{A} \rrbracket)|$ .

Therefore let us assume that Algorithm A does not terminate. In this case, we have  $|\text{im}(\llbracket \mathcal{A} \rrbracket)| \leq k$ , and we have to show that Algorithm B terminates. By Theorem 7.1, for each  $b \in C(\mathcal{A})$ , the mapping  $f_{\mathcal{A},b}$  is bounded or  $b$  has finite additive order. Hence, the assumptions of Theorem 6.6 are satisfied. As

shown in the proof of Theorem 6.6(2), for each  $b \in C(\mathcal{A})$ , the algorithm  $\text{Alg}(b)$  terminates. Hence, Algorithm B terminates.

It follows Algorithm A or Algorithm B terminates. If Algorithm A terminates while Algorithm B is running, we know that  $k < |\text{im}(\llbracket \mathcal{A} \rrbracket)|$ . Now assume Algorithm B terminates while Algorithm A is still running. Then by the proof of Theorem 6.6(2), we can compute effectively a crisp-deterministic wta  $\mathcal{B}$  equivalent to  $\mathcal{A}$ . Then we decide whether  $|\text{im}(\llbracket \mathcal{B} \rrbracket)| \leq k$ .  $\square$

Besides the proof of Theorem 8.2 we also show Algorithm 1 which implements the decision procedure in that proof as a pseudo code. We explain the implementation details as follows.

We assume that we have an enumeration  $\xi_0, \xi_1, \dots$  of  $T_\Sigma$ .

As first step (lines 5–8), Algorithm 1 tests whether  $\mathcal{A}$  has a useful state as in the proof of Lemma 7.9. If the answer is no, then we have  $\llbracket \mathcal{A} \rrbracket = \tilde{\emptyset}$ . Hence Algorithm 1 constructs effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  with  $\llbracket \mathcal{B} \rrbracket = \tilde{\emptyset}$  and terminates with output “yes”. Otherwise, Algorithm 1 trims  $\mathcal{A}$  according to Theorem 4.1.

As second step (lines 10–12), Algorithm 1 tests whether every small loop of  $\mathcal{A}$  has weight  $\mathbb{1}$ . This can be done by the same argument as in the proof of Theorem 7.3). If this is not true, then  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite (by Theorem 7.1) and Algorithm 1 terminates with output “no”. Otherwise, Algorithm 1 continues and, by Corollary 6.7(1), the set  $H(\mathcal{A})$  is finite, and hence also  $C(\mathcal{A})$  is finite.

As third step (lines 13–17), Algorithm 1 computes the set  $C(\mathcal{A}) \subseteq B$ ; by Lemma 6.5 this is possible. Moreover, for each  $b \in C(\mathcal{A})$ , a Boolean value  $\text{flag}(b)$ , a set  $S_b \subseteq B$ , and a set  $U_b \subseteq T_\Sigma$  are initialized. The intuition for  $\text{flag}$ ,  $S_b$ , and  $U_b$  are as follows.

$$\text{flag}(b) = \begin{cases} \text{true} & \text{if } f_{\mathcal{A},b} \text{ is bounded or } \langle b \rangle_\oplus \text{ is finite} \\ \text{false} & \text{otherwise} \end{cases}$$

In the set  $S_b$ , Algorithm 1 collects multiples of  $b$ , i.e.,  $S_b \subseteq \{ib \mid i \in \mathbb{N}\}$ . Moreover, in the set  $U_b$ , Algorithm 1 collects all those trees  $\xi$  for which the multiplicity  $f_{\mathcal{A},b}(\xi)$  is  $i$  for some  $i \in \mathbb{N}$ , i.e.,

$$U_b \subseteq \bigcup_{i \in \mathbb{N}} L_{b,i} \quad \text{where } L_{b,i} = \{\xi \in T_\Sigma \mid f_{\mathcal{A},b}(\xi) = i\} .$$

We note that Algorithm 1 does not compute the set  $U_b$  itself, but a finite representation of  $U_b$  in the form of a finite-state  $\Sigma$ -tree automaton (cf. the proof of Theorem 6.6).

As fourth step (18–41), Algorithm 1 runs the following two algorithms in parallel. Here, however, we explain these two algorithms separately.

- **Algorithm A:** Algorithm A computes  $\llbracket \mathcal{A} \rrbracket(\xi_0), \llbracket \mathcal{A} \rrbracket(\xi_1), \dots$  and collects these values into a set  $W$  (line 21). To compute these values, Algorithm A uses the given enumeration  $\xi_0, \xi_1, \dots$  of  $T_\Sigma$ . Algorithm A terminates if more than  $k$  different values were obtained (line 22). Then the answer to the decision problem is “no”.

- Algorithm B: For each  $b \in C(\mathcal{A})$ , Algorithm B computes simultaneously the sequences  $L_{b,0}, L_{b,1}, \dots$  (line 25) and  $0b, 1b, \dots$  (line 32). Algorithm B terminates, if  $\text{flag}(b) = \text{true}$  for each  $b \in C(\mathcal{A})$  (lines 27 and 30), i.e., for each  $b \in C(\mathcal{A})$  there exists an  $i \in \mathbb{N}$  such that  $\bigcup_{j \in [0,i]} L_{b,j} = T_\Sigma$  or  $ib = jb$  for some  $j < i$ . Clearly, both tests ( $U_b = T_\Sigma$  and  $ib \in S_b$ ) are decidable.

The parallel running of Algorithms A and B terminates, and after termination the assumptions of Theorem 6.6 are satisfied (cf. the proof of Theorem 8.2). Then, by Theorem 6.6(2), Algorithm 1 can compute effectively a crisp-deterministic wta  $\mathcal{B}$  equivalent to  $\mathcal{A}$  and it decides whether  $|\text{im}(\llbracket \mathcal{B} \rrbracket)| \leq k$ . This finished the explanation of Algorithm 1.

To draw conclusions from Theorem 8.2, we need the following notions.

Let  $r$  be a  $(\Sigma, B)$ -weighted tree language,  $E \subseteq B$  be a finite set, and  $b \in B$ . We say that  $r$  is a  $(\Sigma, B, E)$ -recognizable step mapping (for short:  $E$ -recognizable step mapping) if  $r$  is a recognizable step mapping and  $\text{im}(r) = E$ . Moreover, we say that  $r$  is *constant* if  $r = \tilde{b}$  for some  $b \in B$ .

Now we show that, for arbitrary past-finite monotonic and computable strong bimonoid  $B$ ,  $(\Sigma, B)$ -wta  $\mathcal{A}$  given effectively, finite subset  $E \subseteq B$ , and  $b \in B$ , it is decidable whether  $\llbracket \mathcal{A} \rrbracket$  is an  $E$ -recognizable step mapping, and whether  $\llbracket \mathcal{A} \rrbracket = \tilde{b}$  (for the definition of  $\tilde{b}$ , see Section 2.4). Moreover, we revisit the decidability results of Borchardt ([Bor04, Sect. 6]) concerning the constant problem and the constant-on-its-support problem and prove them for a larger class of algebras.

**Corollary 8.3.** Let  $B$  be past-finite monotonic and computable. Then, for each  $(\Sigma, B)$ -wta  $\mathcal{A}$  given effectively, the following questions are decidable:

- Given a finite subset  $E \subseteq B$ , is the weighted tree language  $\llbracket \mathcal{A} \rrbracket$  an  $E$ -recognizable step mapping?
- Given  $b \in B$ , is  $\llbracket \mathcal{A} \rrbracket = \tilde{b}$ ?
- Is  $\llbracket \mathcal{A} \rrbracket$  constant?
- Is  $\llbracket \mathcal{A} \rrbracket$  a recognizable one-step mapping?

*Proof.* First, for each case (a)-(d), we define a number  $k$ . In case (a), let  $k = |E|$ . In cases (b) and (c), we put  $k = 1$ . In case (d), we set  $k = 2$ . Let us decide whether  $|\text{im}(\llbracket \mathcal{A} \rrbracket)| \leq k$  (cf. Theorem 8.2). If this is not the case, the answer to the respective question is "no". Otherwise, a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  is given such that  $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ . Now for this  $\mathcal{B}$  we can decide the questions of (a) – (d).  $\square$

## 9 A comparison of the concepts cost-finiteness and having the finite-image property

Let  $B$  be a strong bimonoid. A  $(\Sigma, B)$ -wta  $\mathcal{A}$  is called *cost-finite* if the set

$$H(\mathcal{A})^{F \neq 0} = \{\text{wt}(\xi, \rho) \mid \xi \in T_\Sigma, \rho \in R_{\mathcal{A}}(\xi), \text{ and } F_{\rho(\varepsilon)} \neq 0\} ,$$

---

**Algorithm 1:** Deciding whether  $\text{im}(\llbracket \mathcal{A} \rrbracket) \leq k$ 

---

**Input:** a  $(\Sigma, B)$ -wta  $\mathcal{A} = (Q, \delta, F)$  given effectively,  
an effective enumeration  $\xi_0, \xi_1, \dots$  of  $T_\Sigma$ , and  $k \in \mathbb{N}_+$

**Output:** "yes" if  $|\text{im}(\llbracket \mathcal{A} \rrbracket)| \leq k$  and "no" otherwise

```
1 Variables  $i \in \mathbb{N}, b \in B, W \subseteq B, \xi \in T_\Sigma$ 
2   flag :  $C(\mathcal{A}) \rightarrow \{\text{true}, \text{false}\}$ 
3   family  $(U_b \subseteq T_\Sigma \mid b \in C(\mathcal{A}))$ 
4   family  $(S_b \subseteq B \mid b \in C(\mathcal{A}))$ 
5 if  $\neg(\mathcal{A}$  has a useful state) then                                % cf. proof of Lemma 7.9
6   |   construct effectively a crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{B} \rrbracket = 0$ 
7   |   return "yes"
8 end
9 trim  $\mathcal{A}$                                                             % cf. Theorem 4.1
10 if  $\neg(\text{all small loops of } \mathcal{A} \text{ have weight } 1)$  then           % cf. proof of Theorem 7.3
11 |   return "no"                                                  %  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is infinite, cf. Theorem 7.1
12 end
                                                                    %  $H(\mathcal{A})$  is finite, cf. Corollary 6.7(1)
13 compute  $C(\mathcal{A})$                                                 % cf. Lemma 6.5
14 foreach  $b \in C(\mathcal{A})$  do
15 |   flag( $b$ )  $\leftarrow$  false,  $U_b \leftarrow \emptyset$ , and  $S_b \leftarrow \emptyset$ 
16 |   construct effectively the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}'_b$            % cf. proof of Theorem 6.6(2)
17 end
18  $i \leftarrow 0$  and  $W \leftarrow \emptyset$ 
19 while true do
20 |    $\xi \leftarrow \xi_i$                                            % query the next tree
21 |    $W \leftarrow W \cup \{\llbracket \mathcal{A} \rrbracket(\xi)\}$ 
22 |   if  $|W| > k$  then return "no"
23 |   if flag $^{-1}$ (false)  $\neq \emptyset$  then
24 |   |   foreach  $b \in \text{flag}^{-1}(\text{false})$  do
25 |   |   |    $U_b \leftarrow U_b \cup \llbracket \mathcal{A}'_b \rrbracket^{-1}(i)$            % cf. Lemma 6.3(2)
26 |   |   |   if  $U_b = T_\Sigma$  then                                % check whether  $f_{\mathcal{A}, b}$  is bounded
27 |   |   |   |   flag( $b$ )  $\leftarrow$  true
28 |   |   |   else
29 |   |   |   |   if  $ib \in S_b$  then                                % check whether  $b$  has finite additive order
30 |   |   |   |   |   flag( $b$ )  $\leftarrow$  true
31 |   |   |   |   else
32 |   |   |   |   |    $S_b \leftarrow S_b \cup \{ib\}$ 
33 |   |   |   |   end
34 |   |   |   end
35 |   |   end
36 |   |    $i \leftarrow i + 1$ 
37 |   else                                                        % conditions of Theorem 6.6(2) hold
38 |   |   construct effectively the crisp-deterministic  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that
39 |   |   |    $\llbracket \mathcal{B} \rrbracket = \llbracket \mathcal{A} \rrbracket$ 
40 |   |   |   if  $|\text{im}(\llbracket \mathcal{B} \rrbracket)| \leq k$  then return "yes" else return "no"
41 end
```

---

is finite (cf. [BFGM05]). We note that in [BFGM05]  $H(\mathcal{A})^{F \neq 0}$  is denoted by  $c(\mathcal{A})$ .

In [BFGM05, Thm. 44], a characterization of cost-finiteness of trim wta was given for finitely factorizing monotonic semirings. In this section, we wish to give such a characterization for monotonic strong bimonoids. As a corollary, we obtain that cost-finiteness is decidable for wta over monotonic strong bimonoids with effective tests for  $0$  and  $1$ . Note however, that the weights of transitions of wta in [BFGM05] have a more general structure than the weights considered here (they are polynomials instead of monomials). Therefore, Lemma 9.1 and Corollary 9.2 cannot be considered as a generalization of the corresponding results [BFGM05, Thm. 44, 46]. Finally, we compare the concept of cost-finiteness [BFGM05] and the concept of having the finite-image property (cf. Corollary 9.3).

**Lemma 9.1.** Let  $B$  be a monotonic strong bimonoid, and let  $\mathcal{A}$  be trim. Then  $\mathcal{A}$  is cost-finite if and only if small loops of  $\mathcal{A}$  have weight  $1$ .

*Proof.* If small loops of  $\mathcal{A}$  have weight  $1$ , then  $\mathcal{A}$  is cost-finite by Corollary 6.7(1).

Now assume  $\mathcal{A}$  has a loop  $\rho$  on a context  $c$  with  $1 \prec \text{wt}(c, \rho)$ . Then as in (4) of the proof of Theorem 7.1, we can produce an infinite set of weights of runs on trees. Hence  $H(\mathcal{A})^{F \neq 0}$  is infinite.  $\square$

As a consequence, we obtain that cost-finiteness of wta is decidable.

**Corollary 9.2.** Let  $B$  be monotonic and have effective tests for  $0$  and  $1$ . Then it is decidable whether an arbitrary  $(\Sigma, B)$ -wta is cost-finite.

*Proof.* Let  $\mathcal{A} = (Q, \delta, F)$  be an arbitrary  $(\Sigma, B)$ -wta. We first check whether  $\mathcal{A}$  has a useful state as in the proof of Lemma 7.9. If  $\mathcal{A}$  has no useful state, then  $H(\mathcal{A})^{F \neq 0} = \emptyset$ , and thus,  $\mathcal{A}$  is obviously cost-finite. Otherwise, by Theorem 4.1, we may assume that  $\mathcal{A}$  is trim. Moreover, by Lemma 9.1,  $\mathcal{A}$  is cost-finite if and only if small loops of  $\mathcal{A}$  have weight  $1$ . The latter property is decidable by the proof of Theorem 7.3.  $\square$

Lastly, we remark that cost-finiteness and having the finite-image property coincide for wta over additively locally finite and past-finite monotonic strong bimonoids.

**Corollary 9.3.** Let  $B$  be monotonic and  $\mathcal{A} = (Q, \delta, F)$  be a  $(\Sigma, B)$ -wta.

1. If  $B$  is additively locally finite and  $\mathcal{A}$  is cost-finite, then  $\mathcal{A}$  has the finite-image property.
2. If  $B$  is past-finite and  $\mathcal{A}$  has the finite-image property, then  $\mathcal{A}$  is cost-finite.

*Proof.* We first prove Statement 1. Since  $\mathcal{A}$  is cost-finite, the set  $C(\mathcal{A})$  is finite. Moreover, we note that  $\text{im}(\llbracket \mathcal{A} \rrbracket)$  is contained in  $\langle C(\mathcal{A}) \rangle_{\oplus}$ . By the assumption of Statement 1, the latter set is finite.

We prove Statement 2: Since  $B$  is monotonic, for every  $\xi \in T_\Sigma$ ,  $q \in Q$  with  $F_q \neq 0$ , and  $\rho \in R_{\mathcal{A}}(q, \xi)$ , we have  $\text{wt}(\xi, \rho) \preceq \text{wt}(\xi, \rho) \otimes F_q \preceq \llbracket \mathcal{A} \rrbracket(\xi)$ . Hence,

$$H(\mathcal{A})^{F \neq 0} \subseteq \bigcup_{b \in \text{im}(\llbracket \mathcal{A} \rrbracket)} \text{past}(b),$$

and the set on the right hand side of the inclusion is finite because  $B$  is past-finite and  $\mathcal{A}$  has the finite-image property. Thus  $\mathcal{A}$  is cost-finite.  $\square$

We note that, in general, the implication of Corollary 9.3(2) does not hold if the condition past-finite is dropped. In fact, the wta  $\mathcal{A}$  in Example 7.2 has the finite-image property, but it is not cost-finite.

In the following example we show that past-finiteness of  $B$  and  $\mathcal{A}$  being cost-finite do not imply that  $\mathcal{A}$  has the finite-image property. In fact, we can give a  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A}$  such that  $\mathcal{A}$  is cost-finite but  $\mathcal{A}$  does not have the finite-image property as follows. (Note that  $\mathbb{N}$  is a past-finite monotonic.)

**Example 9.4.** Let  $\Sigma = \{\gamma^{(1)}, e^{(0)}\}$ . We consider the  $(\Sigma, \mathbb{N})$ -wta  $\mathcal{A} = (\{p, q, r\}, \delta, F)$  with  $\delta_0(\varepsilon, e, p) = \delta_1(p, \gamma, q) = \delta_1(q, \gamma, p) = \delta_1(p, \gamma, r) = \delta_1(r, \gamma, p) = F_p = 1$ , all other transitions have weight 0, and  $F_q = F_r = 0$ .

Clearly,  $H(\mathcal{A})^{F \neq 0} = \{1\}$ , i.e.,  $\mathcal{A}$  is cost-finite. Moreover, for every  $n \in \mathbb{N}$  and  $\gamma^{2n}(e) \in T_\Sigma$ , we have  $\llbracket \mathcal{A} \rrbracket(\gamma^{2n}(e)) = 2^n$ . Since  $\{2^n \mid n \in \mathbb{N}\} \subseteq \text{im}(\llbracket \mathcal{A} \rrbracket)$ , the wta  $\mathcal{A}$  does not have the finite-image property.

We remark that, though  $\mathbb{N}$  is commutative and one-product-free, small loops of  $\mathcal{A}$  have weight 1, the conditions of Corollary 6.7(2) do not hold. In fact,  $C(\mathcal{A}) = \{0, 1\}$  and neither  $f_{\mathcal{A}, 1}$  is bounded nor 1 has finite additive order.  $\square$

## 10 Results for weighted string automata

Let  $\Delta$  be an alphabet. A *weighted string automaton (over  $\Delta$  and  $B$ )* (for short:  $(\Delta, B)$ -wsa) [Sch61, Eil74] is a tuple  $\mathcal{A} = (Q, I, \delta, F)$ , where  $Q$  is a finite set of *states*,  $I : Q \rightarrow B$  is the *initial weight mapping*,  $\delta : Q \times \Delta \times Q \rightarrow B$  is the *transition mapping*, and  $F : Q \rightarrow B$  is the *final weight mapping*. For each  $q \in Q$ , we abbreviate  $I(q)$  and  $F(q)$  by  $I_q$  and  $F_q$ , respectively.

We define the run semantics for  $\mathcal{A}$  as follows. Let  $w = a_1 \cdots a_n$  be a string in  $\Delta^*$  with  $n \in \mathbb{N}$  and  $a_i \in \Delta$  for each  $i \in [n]$ . A *run of  $\mathcal{A}$  on  $w$*  is a string  $\rho = q_0 \cdots q_n$  in  $Q^{n+1}$ . The *weight of  $\rho$  for  $w$* , denoted by  $\text{wt}(w, \rho)$ , is the element of  $B$  defined by

$$\text{wt}(w, \rho) = I_{q_0} \otimes \delta(q_0, a_1, q_1) \otimes \cdots \otimes \delta(q_{n-1}, a_n, q_n) \otimes F_{q_n}.$$

Then the *run semantics of  $\mathcal{A}$*  is the weighted language  $\llbracket \mathcal{A} \rrbracket : \Delta^* \rightarrow B$  defined by

$$\llbracket \mathcal{A} \rrbracket(w) = \bigoplus_{\rho \in Q^{|w|+1}} \text{wt}(w, \rho)$$

for every  $w \in \Delta^*$ . In particular,  $\llbracket \mathcal{A} \rrbracket(\varepsilon) = \bigoplus_{q \in Q} I_q \otimes F_q$ . A weighted language  $r : \Delta^* \rightarrow B$  is *run-recognizable* if there exists a  $(\Delta, B)$ -wsa  $\mathcal{A}$  such that  $r = \llbracket \mathcal{A} \rrbracket$ .

In [FV09, p. 324] it is shown that, for each semiring  $B$ , the concept of  $(\Delta, B)$ -wsa and the concept of  $(\Sigma, B)$ -wta where  $\Sigma$  is a string ranked alphabet are essentially the same. A string ranked alphabet is a ranked alphabet  $\Sigma$  for which  $\Sigma = \Sigma^{(0)} \cup \Sigma^{(1)}$  and  $|\Sigma^{(0)}| = 1$ . In fact, for each  $(\Delta, B)$ -wsa  $\mathcal{A}$  there exists a string ranked alphabet  $\Sigma$ , a bijection  $\text{tree} : \Delta^* \rightarrow T_\Sigma$ , and a  $(\Sigma, B)$ -wta  $\mathcal{B}$  such that  $\llbracket \mathcal{A} \rrbracket(w) = \llbracket \mathcal{B} \rrbracket(\text{tree}(w))$  for each  $w \in \Delta^*$ . The inverse of this statement also holds, and the proof of both directions also works if  $B$  is a strong bimonoid.

Since each string ranked alphabet is a particular ranked alphabet, each of our results for wta also holds for wsa with run semantics. Moreover, the wta in Examples 3.2, 9.4, and 7.2 are examples for wsa with run semantics because in each of these examples, the ranked alphabet of the wta is a string ranked alphabet.

**Acknowledgments.** The authors had obtained Theorem 7.14 for semi-rings. They are thankful to Uli Fahrenberg for a question which prompted the extension of this semiring-result to left or right distributive strong bimonoids.

## References

- [AB87] A. Alexandrakis and S. Bozapalidis. Weighted grammars and Kleene’s theorem. *Inform. Process. Lett.*, 24(1):1–4, 1987.
- [BFGM05] B. Borchardt, Z. Fülöp, Z. Gazdag, and A. Maletti. Bounds for tree automata with polynomial costs. *J. Autom., Lang. Comb.*, 10:107–157, 2005.
- [BMŠ<sup>+</sup>06] B. Borchardt, A. Maletti, B. Šešelja, A. Tepavčević, and H. Vogler. Cut sets as recognizable tree languages. *Fuzzy Sets and Systems*, 157:1560–1571, 2006.
- [Bor04] B. Borchardt. A pumping lemma and decidability problems for recognizable tree series. *Acta Cybernet.*, 16(4):509–544, 2004.
- [Bor05] B. Borchardt. *The Theory of Recognizable Tree Series*. Verlag für Wissenschaft und Forschung, 2005. (Ph.D. thesis, 2004, TU Dresden, Germany).
- [BR82] J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoret. Comput. Sci.*, 18(2):115–148, 1982.
- [BR88] J. Berstel and Ch. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1988.
- [CDG<sup>+</sup>08] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: <http://tata.gforge.inria.fr>, 2008.

- [CDIV10] M. Ćirić, M. Droste, J. Ignjatović, and H. Vogler. Determinization of weighted finite automata over strong bimonoids. *Inform. Sci.*, 180(18):3479–3520, 2010.
- [DFKV20] M. Droste, Z. Fülöp, D. Kószó, and H. Vogler. Crisp-determinization of weighted tree automata over additively locally finite and past-finite monotonic strong bimonoids is decidable. In G. Jirásková and G. Pighizzini, editors, *Descriptive Complexity of Formal Systems (DCFS 2020)*, volume 12442 of *LNCS*, pages 39–51. Springer Nature Switzerland, 2020.
- [DG05] M. Droste and P. Gastin. Weighted automata and weighted logics. In L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, editors, *Automata, Languages and Programming – 32nd Int. Colloquium, ICALP 2005*, volume 3580 of *LNCS*, pages 513–525. Springer-Verlag, 2005.
- [DGMM11] M. Droste, D. Götze, S. Märcker, and I. Meinecke. Weighted tree automata over valuation monoids and their characterizations by weighted logics. In W. Kuich and G. Rahonis, editors, *Algebraic Foundations in Computer Science*, volume 7020 of *LNCS*, pages 30–55. Springer, 2011.
- [DHFV15] M. Droste, D. Heusel, and H. Vogler. Weighted unranked tree automata over tree valuation monoids and their characterization by weighted logics. In A. Maletti, editor, *Algebraic Informatics (CAI 2015)*, volume 9270 of *LNCS*, pages 90–102. Springer, 2015.
- [DK09] M. Droste and W. Kuich. Semirings and formal power series. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, Monographs in Theoretical Computer Science. An EATCS Series, chapter 1, pages 3–28. Springer-Verlag, 2009.
- [DKV09] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 2009.
- [DM12] M. Droste and I. Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Information and Computation*, 220–221:44–59, 2012.
- [DP16] M. Droste and V. Perevoshchikov. Multi-weighted automata and MSO logic. *Theory Comput. Syst.*, 59:231–261, 2016.
- [DSV10] M. Droste, T. Stüber, and H. Vogler. Weighted finite automata over strong bimonoids. *Inform. Sci.*, 180(1):156–166, 2010.
- [DV06] M. Droste and H. Vogler. Weighted tree automata and weighted logics. *Theoret. Comput. Sci.*, 366:228–247, 2006.



- [DV12] M. Droste and H. Vogler. Weighted automata and multi-valued logics over arbitrary bounded lattices. *Theoret. Comput. Sci.*, 418:14–36, 2012.
- [Eil74] S. Eilenberg. *Automata, Languages, and Machines – Volume A*, volume 59 of *Pure and Applied Mathematics*. Academic Press, 1974.
- [ÉK03] Z. Ésik and W. Kuich. Formal tree series. *J. Autom. Lang. Comb.*, 8(2):219–285, 2003.
- [ÉL07] Z. Ésik and G. Liu. Fuzzy tree automata. *Fuzzy Sets and Syst.*, 158:1450–1460, 2007.
- [Eng75] J. Engelfriet. Tree automata and tree grammars. Technical Report DAIMI FN-10, Inst. of Mathematics, University of Aarhus, Department of Computer Science, Denmark, 1975. see also: arXiv:1510.02036v1 [cs.FL] 7 Oct 2015.
- [FKV21] Z. Fülöp, D. Kószó, and H. Vogler. Crisp-determinization of weighted tree automata over strong bimonoids. *Discrete Mathematics & Theoretical Computer Science*, 23(1), 2021.
- [FMV09] Z. Fülöp, A. Maletti, and H. Vogler. A Kleene theorem for weighted tree automata over distributive multioperator monoids. *Theory Comput. Syst.*, 44:455–499, 2009.
- [FSV12] Z. Fülöp, T. Stüber, and H. Vogler. A Büchi-like theorem for weighted tree automata over multioperator monoids. *Theory Comput. Syst.*, 50(2):241–278, 2012. published online 28. October 2010, DOI 10.1007/s00224-010-9296-1.
- [FV09] Z. Fülöp and H. Vogler. Weighted tree automata and tree transducers. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, chapter 9, pages 313–403. Springer-Verlag, 2009.
- [GS84] F. Gécseg and M. Steinby. *Tree Automata*. Akadémiai Kiadó, Budapest, 1984. see also: arXiv:1509.06233v1 [cs.FL] 21 Sep 2015.
- [Har78] M.A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, 1978.
- [HMU07] J.E. Hopcroft, R. Motawi, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Pearson, Addison-Wesley, 2007.
- [IF75] Y. Inagaki and T. Fukumura. On the description of fuzzy meaning of context-free languages. pages 301–328. Academic Press, New York, 1975.

- [KLMP04] I. Klimann, S. Lombardy, J. Mairesse, and C. Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theor. Comput. Sci.*, 327(3):349–373, 2004.
- [KS86] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs in Theoretical Computer Science*. Springer-Verlag, 1986.
- [Kui97] W. Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 9, pages 609–677. Springer-Verlag, 1997.
- [Kui99] W. Kuich. Linear systems of equations and automata on distributive multioperator monoids. In *Contributions to General Algebra 12 - Proceedings of the 58th Workshop on General Algebra “58. Arbeitstagung Allgemeine Algebra”, Vienna University of Technology. June 3-6, 1999*, pages 1–10. Verlag Johannes Heyn, 1999.
- [LB99] O. Louscou-Bozapalidou. Some remarks on recognizable tree series. *Intern J. Computer Math.*, 70:649–655, 1999.
- [MS77] A. Mandel and I. Simon. On finite semigroups of matrices. *Theoret. Comput. Sci.*, 5:101–111, 1977.
- [Rad10] D. Radovanović. Weighted tree automata over strong bimonoids. *Novi Sad J. Math.*, 40(3):89–108, 2010.
- [Rah09] G. Rahonis. Fuzzy languages. In M. Droste, W. Kuich, and H. Vogler, editors, *Handbook of Weighted Automata*, chapter 12, pages 481–517. Springer-Verlag, 2009.
- [RR85] A. Restivo and C. Reutenauer. Rational languages and the burnside problem. *Theoret. Comput. Sci.*, 40:13–30, 1985.
- [Sak09] J. Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009.
- [Sch61] M.P. Schützenberger. On the definition of a family of automata. *Inf. and Control*, 4:245–270, 1961.
- [Sei89] H. Seidl. On the finite degree of ambiguity of finite tree automata. *Acta Informatica*, 26(1):527–542, 1989.
- [SS78] A. Salomaa and M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Texts and Monographs in Computer Science, Springer-Verlag, 1978.
- [Wec78] W. Wechler. *The Concept of Fuzziness in Automata and Language Theory*. Studien zur Algebra und ihre Anwendungen. Akademie-Verlag Berlin, 5. edition, 1978.