Available online at www.sciencedirect.com
SCIENCE DIRECT®
Theoretical Computer Science 328 (2004) 53–75
Theoretical Computer Science
www.elsevier.com/locate/tcs

Branching automata with costs—a way of reflecting parallelism in costs[☆]

Dietrich Kuske[a,*], Ingmar Meinecke[b,1]

[a] Institut für Algebra, Technische Universität Dresden, Zellescher Weg 12-14-Willersbau C244, D-01062 Dresden, Germany
[b] Institut für Informatik, Universität Leipzig, Augustusplatz 10-11, D-04109 Leipzig, Germany

Abstract

Extending work by Lodaya and Weil, we propose a model of *branching automata with costs* in which the calculation of the cost of a parallel composition is handled differently from the calculation of the cost of a sequential composition. Our main result characterizes the behavior of these automata in the spirit of Kleene's and Schützenberger's theorems.

Keywords: Concurrency; Weighted automata; sp-posets; Formal power series; Bisemiring

1. Introduction

This paper reports on our research into parallel systems with costs in the setting of sequential-parallel posets (sp-posets). One of its roots is the line of research initiated by Grabowski [8] and Gischer [7]. They extended previous ideas by Kleene on sequential systems build by nondeterministic choice, iteration and sequential composition. Gischer proposed, in order to model parallel systems, in addition a parallel composition. It turned

☆ An extended abstract of this paper was published in the proceedings of CIAA 2003 [14].

* Corresponding author. Tel.: +49-351-463-33642; fax: +49-351-463-34235.
E-mail addresses: kuske@math.tu-dresden.de (D. Kuske), meinecke@informatik.uni-leipzig.de (I. Meinecke).

1 Supported by the Postgraduate Programme 334 of the German Research Foundation.

0304-3975/$ - see front matter © 2004 Published by Elsevier B.V.
doi:10.1016/j.tcs.2004.07.005

out that sp-posets are ideally suited to describe executions of such systems. Later, Lodaya and Weil [16,15] proposed a finite-state device capable of accepting sp-posets. These automata model parallelism by branching—hence the name “branching automata”.

Suppose we wanted to calculate the minimal duration of a run in a modularly constructed system. The execution time of a sequential composition is the sum of the durations, and that of a parallel composition is the maximum of the durations of the arguments, possibly increased by some duration for the necessary fork and join operations at the beginning and end. A given sp-poset can be executed in different ways and we should consider the minimal duration of all possible executions. In order to accompany this situation, we introduce bisemirings, i.e. structures consisting of two semirings on a joint domain with the same additive operation. Costs of executions in our model of branching automata with costs are then evaluated in such bisemirings and the behavior of a branching automaton with costs is a function that associates with any sp-poset an element from the bisemiring.

It is the aim of this paper to characterize those functions that are associated with branching automata with costs. For this, we employ and extend the machinery from the theory of weighted automata (see [19,11,2,10] for expositions). In this field, one starts from a nondeterministic finite (word or tree) automaton and provides its transitions with weights, costs or multiplicities (depending on the community). Droste raised the question of whether branching automata can be provided with costs in a semiring. Our conceptional contribution in this respect is the observation that one should not just consider cost structures with *one* multiplication, but that several multiplications are necessary to model the phenomena of parallelism.

We characterize the behavior of branching automata with costs in the style of theorems by Kleene [9] and Schützenberger [20] stating the equivalence of regularity and rationality. Several related results are known: for trees, there is a wealth of results of different generality [1,10,6,17,18]; for Mazurkiewicz traces, Droste and Gastin proved a theorem á la Schützenberger [3] and also considered aperiodic formal power series [4]; and for infinite words, Droste and Kuske showed a result in the spirit of the theorems by Büchi and by Schützenberger [5]. When Lodaya and Weil considered languages accepted by branching automata, they observed that unbounded parallelism cannot be captured completely by rational operations, their main results hold for languages of bounded width [15]. Since, in a parallel system, the width corresponds to the number of independent processes, this boundedness restriction seems natural to us. Therefore and similarly, our characterization holds for branching automata with costs only that generate functions with support of bounded width. For this class, we get as our main result the equivalence of acceptance by branching automata with costs and rationality (see Theorem 5.2).

The paper is organized as follows: Section 2 recalls basic notions on sp-posets and defines the concept of a bisemiring. Section 3 introduces our model of branching automata with costs as well as their behavior. It proceeds by the necessary notions on formal power series over sp-posets that allow us to define when such a series is sequential-rational. Furthermore, Section 3 proves that regular series with support of bounded width can be recognized by a special class of branching automata with costs, i.e. those of bounded depth. Section 4 endeavors to prove that rationality implies regularity and Section 5 proves the remaining implication from regularity to rationality.

2. Sequential-parallel posets and bisemirings

Let Σ be a finite alphabet. A *Σ-labeled poset* $t = (V, \leqslant, \tau)$ is a finite poset[2] $(V, \leqslant)$ equipped with a labeling function $\tau : V \longrightarrow \Sigma$. The *width* $\mathrm{wd}(t)$ of t is the maximal size of a subset of V whose elements are mutually incomparable.

The *sequential product* $t_1 \cdot t_2$ of $t_1 = (V_1, \leqslant_1, \lambda_1)$ and $t_2 = (V_2, \leqslant_2, \lambda_2)$ is the Σ-labeled poset

$$(V_1 \dot{\cup} V_2, \leqslant_1 \cup (V_1 \times V_2) \cup \leqslant_2, \tau_1 \cup \tau_2).$$

Graphically, t_2 is put on top of t_1. The *parallel product* $t_1 \| t_2$ is defined as $(V_1 \dot{\cup} V_2, \leqslant_1 \cup \leqslant_2, \tau_1 \cup \tau_2)$, i.e. the two partial orders are put side by side. $\mathrm{SP}(\Sigma)$ or SP denotes the least class of Σ-labeled posets containing all labeled singletons and closed under the application of the sequential and the parallel product, its elements are *sequential-parallel posets*[3] or *sp-posets* for short. We say that t is *sequential* if it cannot be written as a parallel product $t = u \| v$. Dually, t is called *parallel* if it cannot be written as a sequential product $t = u \cdot v$ of $u, v \in \mathrm{SP}$. The only sp-posets which are both sequential and parallel are the singleton posets that we identify with the elements of Σ. By Gischer [7], every $t \in \mathrm{SP}$ admits a maximal *parallel factorization* $t = t_1 \| \dots \| t_n$ (which is unique up to commutativity) where $n \geqslant 1$ and each $t_i \in \mathrm{SP}$ $(i = 1, \dots, n)$ is sequential, and a unique maximal *sequential decomposition* $t = t'_1 \cdot \dots \cdot t'_m$ where $m \geqslant 1$ and each $t'_i \in \mathrm{SP}$ $(i = 1, \dots, m)$ is parallel. Hence, SP is freely generated by Σ subject to associativity of both operations and commutativity of the parallel product.

Definition 2.1. A *bisemiring* $\mathbb{K} = (K, \oplus, \circ, \diamond, 0, 1)$ is a set K equipped with three binary operations called *addition* $\oplus$, *sequential multiplication* $\circ$ and *parallel multiplication* $\diamond$ such that:

- $(K, \oplus, 0)$ is a commutative monoid, $(K, \circ, 1)$ a monoid, and $(K, \diamond)$ a commutative semigroup,
- both $\circ$ and $\diamond$ distribute over $\oplus$, and
- 0 is absorbing for $\circ$ and $\diamond$, i.e. $k \circ 0 = 0 \circ k = k \diamond 0 = 0$ for all $k \in K$.

The structure $(K, \oplus, \circ, 0, 1)$ is a semiring. Moreover, $(K, \oplus, \diamond, 0)$ is almost a semiring; only the parallel multiplication does not have to admit a unit. Let $(K, \oplus, \circ, 0, 1)$ be a semiring and define $x \diamond y = 0$ for all $x, y \in K$. Then $(K, \oplus, \circ, \diamond, 0, 1)$ is a bisemiring. If the semiring K is commutative, also the structure $(K, \oplus, \circ, \circ, 0, 1)$ is a bisemiring. Later in this paper, we will find the *Boolean bisemiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, \wedge, 0, 1)$ that is of this form.

Example 2.2. The structure $(\mathbb{R} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$ is a bisemiring that we referred to in the introduction. Here, 0 is the unit for the sequential multiplication $+$ and $+\infty$ is the absorbing zero of the bisemiring.

---

[2] In this paper, we consider nonempty posets, only.

[3] Called series-rational in [15,16,12].

Let $a \in \Sigma$. We interpret a as some action and assume a has a duration of $\mathrm{time}(a)$. Let $\mathrm{time}(a) = +\infty$ if a cannot be performed. For any $t = t_1 \cdot \ldots \cdot t_n \in \mathrm{SP}$ we put $\mathrm{time}(t) = \mathrm{time}(t_1) + \ldots + \mathrm{time}(t_n)$, and for $t = t_1 \| \ldots \| t_m \in \mathrm{SP}$ we put $\mathrm{time}(t) = \max\{\mathrm{time}(t_1), \ldots, \mathrm{time}(t_m)\}$. Hence, $\mathrm{time} : (\mathrm{SP}, \cdot, \|) \to (\mathbb{R} \cup \{+\infty\}, +, \max)$ is a homomorphism and can be interpreted as the duration time of an sp-poset t. In Example 3.2, we will present an automaton that computes the minimal execution time of an sp-poset using the bisemiring $(\mathbb{R} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$.

Example 2.3. Let Σ be a fixed finite alphabet and let $\mathrm{SP}^1 = \mathrm{SP} \,\dot{\cup}\, \{\varepsilon\}$ where ε acts as unit w.r.t. $\cdot$ and $\|$. Then the class of sp-languages $(\mathfrak{P}(\mathrm{SP}^1), \cup, \cdot, \|, \emptyset, \{\varepsilon\})$ is a bisemiring. Here the multiplications $\cdot$ and $\|$ are defined elementwise.

3. Branching automata with costs

In this section we introduce a model of automata generalizing the concept of branching automata by Lodaya and Weil [15]. We fix an alphabet Σ and a bisemiring $\mathbb{K}$. By $\mathfrak{P}_2(Q)$ we denote the collection of subsets of Q of cardinality 2.

Definition 3.1. A *branching automaton with costs* from $\mathbb{K}$ over the alphabet Σ, or a *BRAC* for short, is a tuple $\mathcal{A} = (Q, T_{\mathrm{seq}}, T_{\mathrm{fork}}, T_{\mathrm{join}}, \lambda, \gamma)$ where

- Q is a finite set of states,
- $T_{\mathrm{seq}} : Q \times \Sigma \times Q \longrightarrow \mathbb{K}$ is the *sequential transition function*,
- $T_{\mathrm{fork}} : Q \times \mathfrak{P}_2(Q) \longrightarrow \mathbb{K}$ is the *fork transition function*,
- $T_{\mathrm{join}} : \mathfrak{P}_2(Q) \times Q \longrightarrow \mathbb{K}$ is the *join transition function*,
- $\lambda, \gamma : Q \longrightarrow \mathbb{K}$ are the *initial* and the *final* cost function, respectively.

We write $p \xrightarrow{a}_k q$ if $T_{\mathrm{seq}}(p, a, q) = k \neq 0$ and call it a *sequential transition*; if it only matters that the costs are distinct from 0, we write $p \xrightarrow{a} q$. Similarly, we write $p \to_k \{p_1, p_2\}$ and $p \to \{p_1, p_2\}$ if $T_{\mathrm{fork}}(p, \{p_1, p_2\}) = k \neq 0$. In the same way, we understand $\{q_1, q_2\} \to_k q$ and $\{q_1, q_2\} \to q$. A state $q \in Q$ is an *initial state* if $\lambda(q) \neq 0$. Dually, q is a *final state* if $\gamma(q) \neq 0$.

3.1. The behavior of branching automata

In order to calculate the cost of an sp-poset $t \in \mathrm{SP}$ in a BRAC $\mathcal{A}$ we introduce the notion of a *path* in $\mathcal{A}$. We consider labeled directed graphs $G = (V, E, \nu, \eta)$ with a unique source $\mathrm{src}(G)$ and a unique sink $\mathrm{sk}(G)$ where $\nu : V \longrightarrow Q$ is a total and $\eta : E \longrightarrow \Sigma$ is a partial function. The labeled graph G with $V = \{v_1, v_2\}$, $E = \{(v_1, v_2)\}$, $\nu(v_i) = p_i$ $(i = 1, 2)$ and $\eta(v_1, v_2) = a$ is an *atomic* path if $p_1 \xrightarrow{a} p_2$ is a sequential transition of $\mathcal{A}$. Now let $G_i = (V_i, E_i, \nu_i, \eta_i)$ be paths for $i = 1, 2$. If $\nu_1(\mathrm{sk}(G_1)) = \nu_2(\mathrm{src}(G_2))$, we define the *sequential product* $G = G_1 \cdot G_2$ of G_1 and G_2. At that, G is the union of both graphs where the sink of G_1 and the source of G_2 (which have the same label) are fused to one vertex. If $p \to \{\nu_1(\mathrm{src}(G_1)), \nu_2(\mathrm{src}(G_2))\}$ is a fork and $\{\nu_1(\mathrm{sk}(G_1)), \nu_2(\mathrm{sk}(G_2))\} \to q$ a join transition of $\mathcal{A}$ then the *p–q-parallel product* $G_1 \|_{p,q} G_2 = (V, E, \nu, \eta)$ is a path with vertices $V = V_1 \dot\cup V_2 \dot\cup \{u, w\}$ and edges $E = E_1 \dot\cup E_2 \dot\cup \{(u, \mathrm{src}(G_i)), (\mathrm{sk}(G_i), w) \mid i = 1, 2\}$. For $v \in V_i$ we put $\nu(v) = \nu_i(v)$ $(i = 1, 2)$, furthermore $\nu(u) = p$ and $\nu(w) = q$, and $\eta = \eta_1 \dot\cup \eta_2$. The sequential product is associative, and every *p*–*q*-parallel product is commutative, but not associative. The *set* $\mathrm{PT}(\mathcal{A})$ *of paths* of the BRAC $\mathcal{A}$ is the smallest set of labeled directed graphs *G* that contains all atomic paths of $\mathcal{A}$ and is closed under the sequential product and under all *p*–*q*-parallel products with $p, q \in Q$ as defined above.

Inductively, we define the label $\mathrm{lab}(G) \in \mathrm{SP}$ and the cost $\mathrm{cost}(G) \in \mathbb{K}$ for any path *G*. For an atomic path $G : p \xrightarrow{a} q$, we put $\mathrm{lab}(G) = a$ and $\mathrm{cost}(G) = T_{\mathrm{seq}}(p, a, q)$. Further, $\mathrm{lab}(G_1 \cdot G_2) = \mathrm{lab}(G_1) \cdot \mathrm{lab}(G_2)$, $\mathrm{cost}(G_1 \cdot G_2) = \mathrm{cost}(G_1) \circ \mathrm{cost}(G_2)$, and $\mathrm{lab}(G_1 \|_{p,q} G_2) = \mathrm{lab}(G_1) \| \mathrm{lab}(G_2)$. To define the cost of $G = G_1 \|_{p,q} G_2$, let $p_i = \nu_i(\mathrm{src}(G_i))$ and $q_i = \nu_i(\mathrm{sk}(G_i))$ for $i = 1, 2$. Then

$$\mathrm{cost}(G) = T_{\mathrm{fork}}(p, \{p_1, p_2\}) \circ [\mathrm{cost}(G_1) \diamond \mathrm{cost}(G_2)] \circ T_{\mathrm{join}}(\{q_1, q_2\}, q) \ .$$

The cost of such a parallel path can be interpreted as follows. At first we have a cost for branching the process, then the cost for the two subprocesses and finally the cost for joining the subprocesses. These costs come up one after the other and, therefore, are multiplied sequentially. On the other hand, the costs of the two subprocesses are multiplied in parallel.

We denote by $G : p \xrightarrow{t} q$ that *G* is a path from state *p* to state *q* with label $t \in \mathrm{SP}$. Then the cost of some $t \in \mathrm{SP}$ from *p* to *q* in $\mathcal{A}$ is given by summing up the costs of all possible paths from *p* to *q* with label *t*

$$\mathrm{cost}_{p,q}(t) = \bigoplus_{G : p \xrightarrow{t} q} \mathrm{cost}(G) \ .$$

The cost of $t \in \mathrm{SP}$ in $\mathcal{A}$ is defined as

$$(\mathcal{S}(\mathcal{A}), t) = \mathrm{cost}_{\mathcal{A}}(t) = \bigoplus_{p,q \in Q} \lambda(p) \circ \mathrm{cost}_{p,q}(t) \circ \gamma(q) \ .$$

Then $\mathcal{S}(\mathcal{A}) : \mathrm{SP} \longrightarrow \mathbb{K}$ is the *behavior of* $\mathcal{A}$ or, equivalently, is *recognized* by $\mathcal{A}$. A function $S : \mathrm{SP} \longrightarrow \mathbb{K}$ is *regular* if there is a BRAC $\mathcal{A}$ such that $S = \mathcal{S}(\mathcal{A})$.

Example 3.2. In this example, we define a branching automaton with costs $\mathcal{A}$ whose behavior measures the height of a poset, i.e. $(\mathcal{S}(\mathcal{A}), t) = \mathrm{height}(t)$ for any sp-poset *t*. For this to work, we use the bisemiring $(\mathbb{R} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$ from Example 2.2. The automaton has just three states p_0, p_1, p_2. Any of these states can fork into the other two at cost 0; similarly, any two distinct of these states can be joined into the remaining one at cost 0. In any state, we can execute any action at cost 1 (without changing the state). Any state is initial and final with $\lambda(p_i) = \gamma(p_i) = 0$. Fig. 1 depicts a run of this BRAC on the sp-poset $t = (aa\|b)(a\|bb)$. In this picture, join- and fork-transitions are indicated by a semi-circle between the edges involved. Next to these semi-circles, we denote the cost of the corresponding transition. The path is the sequential product of two "bubbles" whose costs we calculate first. The first bubble is the parallel product of an atomic *b*-transition and the sequential *aa*-path. Since the join- and fork-transitions involved in this product have cost 0, the cost of a bubble is $0 + \max(1 + 1, 1) + 0 = 2$. Since this holds for both bubbles, the total cost is $2 + 2 = 4$ which equals the height of the poset $(aa\|b)(a\|bb)$.

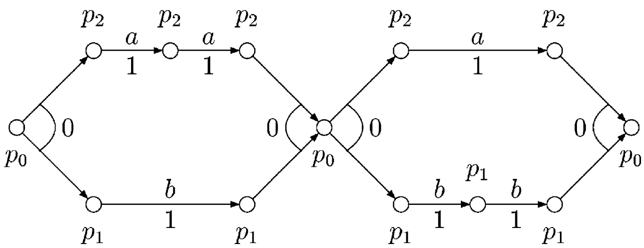

Fig. 1. A path measuring the height (and the width).

If all the actions to be executed in an sp-poset require one time unit, the automaton above calculates the execution time. The example can be easily modified to accompany the following situation: there is some system in which the execution time of atomic actions depends on the state in which they are executed. One can then construct a BRAC that calculates the minimal execution time of an sp-poset. If, instead of working in $(\mathbb{R} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$, we work in the bisemiring $(\mathbb{R} \cup \{-\infty\}, \max, +, \mathrm{nax}, -\infty, 0)$ where $\mathrm{nax}(k, l) = \max(k, l)$ if $k, l \in \mathbb{R}$ and $\mathrm{nax}(k, l) = -\infty$ otherwise, this automaton would compute the maximal execution time.

Example 3.3. In this example, we present a BRAC that measures the width of a poset, i.e. $(\mathcal{S}(\mathcal{A}), t) = \mathrm{wd}(t)$ for any sp-poset t. To achieve this, we take the bisemiring $(\mathbb{R} \cup \{-\infty, +\infty\}, \min, \max, +, +\infty, -\infty)$ and use the automaton from Example 3.2. Consider Fig. 1 that depicts a run on the sp-poset $(aa\|b)(a\|bb)$. But this time, the cost of the *aa*-path is evaluated by $\max(1, 1) = 1$. Hence the cost of the first bubble is $\max(0, 1 + 1, 0) = 2$ and similarly for the second bubble. Hence the total cost is $\max(2, 2) = 2$ which equals the width of the poset in question.

In [13], we give another example of a BRAC, this time over the bisemiring of subsets of Σ^*. It calculates, from an sp-poset t, the set of words that label a maximal linearly ordered subset of t.

3.2. Formal power series over sp-posets

To characterize the possible behavior of branching automata with costs, we introduce the notion of formal power series over sp-posets with values in a bisemiring. This concept is both a generalization of the well known formal power series over words (cf. [19]) and the sp-languages as introduced by Lodaya and Weil [15].

A *formal power series over* SP *with values in the bisemiring* $\mathbb{K}$ or *sp-series* is a function $S : \mathrm{SP} \longrightarrow \mathbb{K}$. With $(S, t) := S(t)$, it is written as a formal sum

$$S = \sum_{t \in \mathrm{SP}} (S, t)t.$$

The *support* of S is $\operatorname{supp} S := \{t \in \mathrm{SP} \mid (S, t) \neq 0\}$. Formal power series whose support is a singleton are called *monomials*. The class of all formal power series over SP with values in $\mathbb{K}$ is denoted by $\mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$.

Now we introduce some operations for sp-series. Let $S, T \in \mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$. We define:

1. the *sum* $S + T$ by $(S + T, t) := (S, t) \oplus (T, t)$,
2. the *scalar products* $k \cdot S$ and $S \cdot k$ for $k \in \mathbb{K}$ by $(k \cdot S, t) := k \circ (S, t)$ and $(S \cdot k, t) := (S, t) \circ k$,
3. the *sequential product* $S \cdot T$ by $(S \cdot T, t) := \bigoplus_{t = u \cdot v} (S, u) \circ (T, v)$ where the sum is taken over all sequential factorizations $t = u \cdot v$ with $u, v \in \mathrm{SP}$,
4. the *parallel product* $S \| T$ by $(S \| T, t) := \bigoplus_{(u,v)\,:\, t = u \| v} (S, u) \diamond (T, v)$ where we add over all pairs (u, v) such that $t = u \| v$ with $u, v \in \mathrm{SP}$ (because of the commutativity of $\|$ in SP, both $(S, u) \diamond (T, v)$ and $(S, v) \diamond (T, u)$ contribute to the sum),
5. the *sequential iteration* S^+ of an sp-series S by
$$(S^+, t) := \bigoplus_{1 \leqslant n \leqslant |t|} \quad \bigoplus_{t = u_1 \cdot \ldots \cdot u_n} (S, u_1) \circ \ldots \circ (S, u_n),$$
where we sum up over all possible sequential factorizations of t.

Collectively, we refer to these operations as the *sequential-rational operations of* $\mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$.

Similar to the sequential iteration, one can define the parallel iteration. Already in the theory of sp-languages, this parallel iteration causes severe problems [16]. Smoother results are obtained if one does not allow the parallel iteration in rational expressions [15,12].

The operations $+$, $\cdot$, and $\|$ are associative on $\mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$, and $+$ and $\|$ are even commutative. The series $\mathbf{0}$ with $(\mathbf{0}, t) = 0$ for all $t \in \mathrm{SP}$ is the unit w.r.t. $+$ and absorbing w.r.t. $\cdot$ and $\|$. Note that we do not have an sp-series that acts as unit with respect to the sequential product $\cdot$ (we excluded the empty poset from our considerations). Hence $\mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$ is not a bisemiring.

The class $\mathbb{K}^{\mathrm{s-rat}}\langle\langle \mathrm{SP}\rangle\rangle$ of *sequential-rational sp-series* over Σ with values in $\mathbb{K}$ is the smallest class containing all monomials that is closed under the sequential-rational operations of $\mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$. If $\mathbb{K}$ is the Boolean bisemiring, these are the series-rational sp-languages of Lodaya and Weil.

Let $\mathbb{K}$ and $\mathbb{K}'$ be bisemirings, $h : \mathbb{K} \to \mathbb{K}'$ a bisemiring homomorphism and $f : \mathrm{SP}(\Sigma_1) \to \mathrm{SP}(\Sigma_2)$ an homomorphism of the sp-algebras. Further, let $S \in \mathbb{K}\langle\langle \mathrm{SP}(\Sigma_1)\rangle\rangle$. We define the sp-series $\overline{h}(S) \in \mathbb{K}'\langle\langle \mathrm{SP}(\Sigma_1)\rangle\rangle$ by $(\overline{h}(S), t) = h(S, t)$ for any $t \in \mathrm{SP}(\Sigma_1)$. Further, we define the sp-series $\overleftarrow{f}(S) \in \mathbb{K}\langle\langle \mathrm{SP}(\Sigma_2)\rangle\rangle$ for any $t \in \mathrm{SP}(\Sigma_2)$ by $(\overleftarrow{f}(S), t) = \bigoplus_{s \in f^{-1}(t)} (S, s)$. Note that the last sum is finite since $|s| \leqslant |f(s)|$ holds for all $s \in \mathrm{SP}(\Sigma_1)$.

Proposition 3.4. *For any $k \in \mathbb{K}$, $\overline{h}(k \cdot S) = h(k) \cdot \overline{h}(S)$, and $\overline{h}(S \cdot k) = \overline{h}(S) \cdot h(k)$. Further, $\overline{h}$ commutes with all other sequential-rational operations, and $\overleftarrow{f}$ commutes with all sequential-rational operations. In particular, $\overline{h}$ and $\overleftarrow{f}$ preserve sequential-rationality.*

Proof. The proof for $\overline{h}$ is straightforward. Obviously, $\overleftarrow{f}$ commutes with sum and the scalar products. We show that $\overleftarrow{f}$ commutes with the sequential product. Let $t \in \mathrm{SP}(\Sigma_2)$, and $S, T \in \mathbb{K}\langle\langle \mathrm{SP}(\Sigma_1)\rangle\rangle$. Then we have

$$(\overleftarrow{f}(S\cdot T), t) = \bigoplus_{s\in f^{-1}(t)} (S\cdot T, s) = \bigoplus_{s\in f^{-1}(t)} \bigoplus_{u,v:\, u\cdot v=s} (S,u)\circ(T,v)$$
$$= \bigoplus_{u,v:\, f(u\cdot v)=t} (S,u)\circ(T,v).$$

On the other hand,

$$(\overleftarrow{f}(S)\cdot\overleftarrow{f}(T), t) = \bigoplus_{u',v':\, u'\cdot v'=t} (\overleftarrow{f}(S), u')\circ(\overleftarrow{f}(T), v')$$
$$= \bigoplus_{u',v':\, u'\cdot v'=t} \left[\bigoplus_{u\in f^{-1}(u')} (S,u)\right]\circ\left[\bigoplus_{v\in f^{-1}(v')} (T,v)\right]$$
$$= \bigoplus_{u',v':\, u'\cdot v'=t} \bigoplus_{\substack{u\in f^{-1}(u')\\ v\in f^{-1}(v')}} (S,u)\circ(T,v)$$
$$= \bigoplus_{u,v:\, f(u)\cdot f(v)=t} (S,u)\circ(T,v).$$

Thus, $\overleftarrow{f}$ commutes with the sequential product. One shows similarly that $\overleftarrow{f}$ commutes with the parallel product and the sequential iteration. □

We consider as a special case the Boolean bisemiring $\mathbb{B}$. An *sp-language* L is a subset of SP. Any sp-language $L \subseteq \mathrm{SP}$ can be identified with its *characteristic series* $\mathbf{1}_L$ where $(\mathbf{1}_L, t) = 1$ iff $t \in L$. This isomorphism maps the class $\mathbb{B}^{\mathrm{s-rat}}\langle\langle \mathrm{SP}\rangle\rangle$ to the class of *sequential-rational sp-languages* $\mathrm{SP}^{\mathrm{s-rat}}$ (cf. [15] for the definition). Therefore, the theory of sp-series is a generalization of the theory of sp-languages as investigated by Lodaya and Weil [15].

An sp-language $L \subseteq \mathrm{SP}$ has *bounded width* if there exists an integer n such that for each element $t \in L$ we have $\mathrm{wd}(t) \leqslant n$. Similarly, we call $S \in \mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$ *width-bounded* if supp S has bounded width. From the definition of sequential-rational sp-series we get immediately:

Proposition 3.5. *Any sequential-rational sp-series has bounded width*.

As for sp-languages the opposite is not true.

3.3. Bounded width and bounded depth

The bounded width of a regular sp-series is reflected by the "bounded depth" of a BRAC. Every atomic path is of depth 0, $\mathrm{dp}(G_1\cdot G_2) = \max\{\mathrm{dp}(G_1), \mathrm{dp}(G_2)\}$ and $\mathrm{dp}(G_1\|_{p,q} G_2) = 1+\max\{\mathrm{dp}(G_1), \mathrm{dp}(G_2)\}$. Therefore, the depth of a path measures the nesting of branchings within the path. A BRAC $\mathcal{A}$ is of *bounded depth* if the depth of its paths is uniformly bounded (Lodaya and Weil [15] require this for successful paths, only). Any series recognized by a BRAC of bounded depth is of bounded width. The converse for sp-languages was shown in [12] by just counting and thereby limiting the depth of a path. That proof can be extended to bisemirings that do not allow an additive decomposition of 0. The problem in the general case arises from the existence of paths of different depths having the same label t. Then two such paths can have non-zero costs, but the sum of these costs can be 0. If now the path with larger depth is disabled, the cost of the sp-poset t changes (see the technical report [13] for a more elaborated example). To overcome this problem, we will keep track of the actual width (and not just the depth) of a poset. This is achieved by a stack where the widths encountered up to the last fork transition are stored. More precisely, let G be a path and x a node in G. We describe the content of the stack that the new automaton assumes at the node x. Those fork transitions between the source of G and x that are unmatched (i.e., not closed by a join transition) before x forms a sequence. Two consecutive such forks limit a subpath of G consisting of all the nodes in between them. The stack at x consists of the sequence of widths of the labels of these subpaths. In addition, it contains the width of the subpath before the first unmatched fork as well as after the last unmatched fork transition. In order to limit the successful paths to those with label of width at most n, we limit the size of the stack as well as the numbers to be stored therein to n. This allows to perform the construction within the realm of finite-state systems.

Here are some definitions needed in the sequel. If $G = G_1 \cdot \ldots \cdot G_n$ then G_i is a *direct subpath* of G for $i = 1, \ldots, n$. Similarly, for $G = G_1 \|_{p,q} G_2$ both G_1 and G_2 are *direct subpaths* of G. We write $H \sqsubset G$ if H is a direct subpath of G. The transitive closure of $\sqsubset$ is denoted by $\prec$. If $H \prec G$ we say H is a *subpath* of G. By $S_{|n}$ we denote the restriction of S to the sp-posets of width less or equal to n, i.e.

$$(S_{|n}, t) = \begin{cases} (S, t) & \text{if } \mathrm{wd}(t) \leqslant n, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 3.6. *Let $\mathbb{K}$ be an arbitrary bisemiring, $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ and $n \in \mathbb{N}$. If S is regular then $S_{|n}$ can be recognized by a depth-bounded BRAC.*

Proof. Let $\mathcal{A}$ be a BRAC recognizing S. We put $[n] := \{1, \ldots, n\}$, and $[n]^\star$ denotes the set of all finite words over $[n]$ and $[n]^+$ the set of nonempty words over $[n]$. From $\mathcal{A}$ we construct a new automaton $\mathcal{A}'$ with state set $Q' = \{(p, w) \in Q \times [n]^+ \quad : \quad |w| \leqslant n\}$ such that $\mathcal{A}'$ simulates $\mathcal{A}$ in the first component and counts the nesting of fork and join transitions in the second component that we think of as a stack. More detailed, the height of the stack counts the depth of the path and the values stored within this stack keep track of the width of the label of the path.

- A sequential transition does not change the width of the label, hence the stack is left untouched:

$$T'_{\mathrm{seq}}((p, u), a, (q, v)) = \begin{cases} T_{\mathrm{seq}}(p, a, q) & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

- A fork transition increases the depth of a path, hence it pushes a new value onto the stack. Since there are no parallel actions after the fork yet, this value is 1:

$$T'_{\mathrm{fork}}((p, u), \{(p_1, u_1), (p_2, u_2)\}) \\ = \begin{cases} T_{\mathrm{fork}}(p, \{p_1, p_2\}) & \text{if } u_1 = u_2 = u1, \\ 0 & \text{otherwise.} \end{cases}$$

- Since a join transition results in a node of smaller depth, it decreases the size of the stack. The width of the subpath since the matching fork transition f is the sum of its two parallel subpaths, i.e., of the two top stack elements at the nodes joined. The width of the subpath since the previous unmatched fork transition f' is the maximum of this sum and the width of the subpath between these two fork transitions f' and f. Since the fork f' is now the last unmatched one, this maximum is pushed onto the stack:

  $T'_{\mathrm{join}}(\{(q_1, v_1), (q_2, v_2)\}, (q, v)) = T_{\mathrm{join}}(\{q_1, q_2\}, q)$ if there are $w \in [n]^\star$ and $x, y_1, y_2, z \in [n]$ such that $v_1 = wxy_1$, $v_2 = wxy_2$, $v = wz$ with $z = \max\{x, y_1 + y_2\}$. Otherwise, $T'_{\mathrm{join}}(\{(q_1, v_1), (q_2, v_2)\}, (q, v)) = 0$.
- At the source of a path, no parallel actions have been executed. Hence the stack contains just the number 1:

$$\lambda'(p, u) = \begin{cases} \lambda(p) & \text{if } u = 1, \\ 0 & \text{otherwise.} \end{cases}$$

  A successful path does not contain any unmatched fork transitions, hence the stack contains just one element:

$$\gamma'(q, v) = \begin{cases} \gamma(q) & \text{if } |v| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

As any fork transition increments the height of the stack and since the height is bounded by n, any path of $\mathcal{A}'$ has depth at most $n - 1$. Hence $\mathcal{A}'$ is of bounded depth. We will show that $\mathcal{A}'$ recognizes $S_{|n}$. To this aim, we first show that a path in $\mathcal{A}'$ changes at most the topmost element of the stack. This happens only if the width of the label is larger than the topmost element before starting the path in which case it gets replaced.

Claim 3.7. *Let $G : (p, u) \xrightarrow{t} (q, v)$ be a path of $\mathcal{A}'$ with label t such that* $\mathrm{wd}(t) \leqslant n$*. Then there exist $w \in [n]^\star$ and $x, y \in [n]$ such that $u = wx$, $v = wy$, and $y = \max\{x, \mathrm{wd}(t)\}$.*

Assume $G = (p, u) \xrightarrow{t} (q, v)$ is atomic. Then $t \in \Sigma$, i.e. $\mathrm{wd}(t) = 1$. Hence, Claim 3.7 follows immediately from the definition of T'_{seq}. If $G = G_1 \cdot \ldots \cdot G_k$ is the sequential decomposition of G with $\mathrm{lab}(G_i) = t_i$, $\mathrm{lab}(G) = t$ and $\mathrm{wd}(t) \leqslant n$ we have $\mathrm{wd}(t) = \max\{\mathrm{wd}(t_i) : i = 1, \ldots, k\}$. By structural induction we get Claim 3.7 for the path G. Now let $G = G_1 \|_{(p,u),(q,v)} G_2$ with $G_i = (p_i, u_i) \xrightarrow{t_i} (q_i, v_i)$ for $i = 1, 2$. For $t = \mathrm{lab}(G)$ we have $\mathrm{wd}(t) \leqslant n$ by assumption and $t = t_1 \| t_2$. Hence $\mathrm{wd}(t_i) < n$. By the definition of fork transitions in $\mathcal{A}$ we have $u_1 = u_2 = u1$. By induction there are $x_i \in [n]$ such that $v_i = ux_i$ and $x_i = \mathrm{wd}(t_i)$ for $i = 1, 2$. Let $u = wx$ for $w \in [n]^\star$ and $x \in [n]$. Thus by the definition of a join transition in $\mathcal{A}'$ we get $v = wy$ for $y = \max\{x, x_1 + x_2\} = \max\{x, \mathrm{wd}(t_1) + \mathrm{wd}(t_2)\} = \max\{x, \mathrm{wd}(t)\}$. This is Claim 3.7.

Now consider $t \in \mathrm{SP}$ with $\mathrm{wd}(t) > n$ and suppose H is a path in $\mathcal{A}'$ with label t. By decomposition of H there is a subpath G of H with $\mathrm{wd}(\mathrm{lab}(G)) > n$ and $G = G_1 \|_{p',q'} G_2$ such that $\mathrm{wd}(\mathrm{lab}(G_i)) \leqslant n$ for $i = 1, 2$. Let $p' = (p, u)$ and $q' = (q, v)$. For G_1 and G_2 we can apply Claim 3.7. If $u = wx$ with $w \in [n]^\star$ and $x \in [n]$ then $v = wy$ with

$y = \max\{x, \mathrm{wd}(t)\} > n$. Since $y \leqslant n$ by definition, a path G with $\mathrm{wd}(\mathrm{lab}(G)) > n$ cannot exist. Hence, for all $t \in \mathrm{SP}$ with $\mathrm{wd}(t) > n$ we have $t \notin \mathrm{supp}(\mathcal{S}(\mathcal{A}'))$.

Let $r \in Q$ be an initial and $s \in Q$ a final state of $\mathcal{A}$. Now we consider the following sets of paths:

- $\mathrm{PT}_{r,s,\leqslant n}(\mathcal{A})$ is the set of all paths from r to s in $\mathcal{A}$ whose labels have width at most n, and
- $\mathrm{PT}_{r',s'(\forall\alpha)}(\mathcal{A}')$ is the set of all paths from $r' = (r, 1)$ to some $s' = (s, \alpha)$ in $\mathcal{A}'$ where $\alpha \in [n]$.

We define $\varphi_{r,s} : \mathrm{PT}_{r',s'(\forall\alpha)}(\mathcal{A}') \longrightarrow \mathrm{PT}_{r,s,\leqslant n}(\mathcal{A})$ by just forgetting the second component (i.e. the word) of the states of a path of $\mathcal{A}'$. By definition of $\mathcal{A}'$ the mapping $\varphi_{r,s}$ is well defined and preserves labels and costs of a path. Moreover, $\varphi_{r,s}$ is injective because the second component of the states of a path $G' \in \mathrm{PT}_{r',s'(\forall\alpha)}(\mathcal{A}')$ is determined by $\varphi_{r,s}(G')$ and r'. Our next aim is to show that any path of $\mathcal{A}$ can be simulated by a path of $\mathcal{A}'$ provided the width of the label is at most n, i.e., we want to show surjectivity of $\varphi_{r,s}$. We prove this by induction on the depth of the path, the following claim forms the inductive argument:

Claim 3.8. *Let $G : p \xrightarrow{t} q$ be a path in $\mathcal{A}$ with $\mathrm{wd}(t) \leqslant n$ and $\mathrm{dp}(G) = d$. Furthermore, let $u \in [n]^+$ with $|u| + d \leqslant n$. Then there are $v \in [n]^+$ and a path $G' = (p, u) \xrightarrow{t} (q, v)$ in $\mathcal{A}'$ such $\varphi_{p,q}(G') = G$.*

For any atomic path G the depth of G is 0 and Claim 3.8 is obvious. Next, let $G = G_1 \cdot \ldots \cdot G_k$ allow a sequential decomposition. Note that $\mathrm{dp}(G) = \max\{\mathrm{dp}(G_i) : i = 1, \ldots, k\}$. By structural induction, Claim 3.8 is true for G_1. Considering Claim 3.7 and applying induction to $G_2, \ldots, G_k$, Claim 3.8 holds also true for G. Now, let $G = G_1 \|_{p,q} G_2$ have a parallel decomposition. We construct G' as follows: It starts in (p, u) with a fork simulating the first one in G. This is possible because $|u| + d \leqslant n$. This fork branches into states whose second component is $u1$. Note that $\mathrm{dp}(G_i) \leqslant d - 1$ for $i = 1, 2$. Using the induction hypothesis for G_1 and G_2 we get two paths G_1' and G_2'. By Claim 3.7, for $i = 1, 2$ the path G_i' ends in a state whose second component is ux_i where $x_i = \max\{1, \mathrm{wd}(\mathrm{lab}(G_i'))\} = \mathrm{wd}(\mathrm{lab}(G_i'))$. Since $x_1 + x_2 = \mathrm{wd}(\mathrm{lab}(G_1')) + \mathrm{wd}(\mathrm{lab}(G_2')) = \mathrm{wd}(\mathrm{lab}(G)) \leqslant n$ there is the required join transition copying the last join in G. Then $\varphi_{p,q}(G') = G$ is obvious by the construction. This proves Claim 3.8.

Now we return to the mapping $\varphi_{r,s}$. Any path G from $\mathrm{PT}_{r,s,\leqslant n}(\mathcal{A})$ satisfies the prerequisites of Claim 3.8. By Claims 3.8 and 3.7, there is $G' \in \mathrm{PT}_{r',s'(\forall\alpha)}(\mathcal{A}')$ with $\varphi_{r,s}(G') = G$. Hence, $\varphi_{r,s}$ is surjective. Thus, there is a bijective mapping $\varphi_{r,s} : \mathrm{PT}_{r',s'(\forall\alpha)}(\mathcal{A}') \longrightarrow \mathrm{PT}_{r,s,\leqslant n}(\mathcal{A})$ preserving labels and costs for all pairs (r, s) of an initial state r and a final state s of $\mathcal{A}$.

Considering the initial and final states of $\mathcal{A}'$ and $t \notin \mathrm{supp}(\mathcal{S}(\mathcal{A}'))$ for any $t \in \mathrm{SP}$ with $\mathrm{wd}(t) > n$, we get immediately $\mathcal{S}(\mathcal{A}') = S_{|n}$. $\square$

As a consequence of the last result we get:

Corollary 3.9. *Let S be a regular sp-series. Then S is of bounded width if, and only if, it can be recognized by a BRAC of bounded depth.*

Proof. Let S be of bounded width, and let $n = \max(\mathrm{wd}(\mathrm{supp}(S)))$. Hence $S = S_{|n}$. By Proposition 3.6, S can be recognized by a BRAC of bounded depth. On the other hand, if S is recognized by a BRAC of bounded depth d, then S is of bounded width at most 2^d. □

4. Closure properties of regular sp-series

Proposition 4.1. *Let $S_1, S_2 \in \mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$ be regular sp-series. Then $S_1 + S_2$ is again a regular sp-series.*

Proof. Let S_i be recognized by the BRAC $\mathcal{A}_i$ $(i = 1, 2)$. We define the disjoint union $\mathcal{A} = (Q_1 \dot{\cup} Q_2, T_{\mathrm{seq}}, T_{\mathrm{fork}}, T_{\mathrm{join}}, \lambda, \gamma)$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ by

$$T_{\mathrm{seq}}(p, a, q) = \begin{cases} T_{i\,\mathrm{seq}}(p, a, q) & \text{if } p, q \in Q_i, \\ 0 & \text{otherwise.} \end{cases}$$

and similarly for $T_{\mathrm{fork}}, T_{\mathrm{join}}, \lambda, \gamma$.

Then $S_1 + S_2$ is recognized by $\mathcal{A}$. This can be easily seen by considering that a path of $\mathcal{A}$ is either completely in $\mathcal{A}_1$ or in $\mathcal{A}_2$ since a path contains only transitions with non-zero cost. □

In the sequel, we will use branching automata with restricted possibilities to enter and to leave the automaton. A BRAC $\mathcal{A}$ is called *initial-normalized* if there is a unique initial state i and this state satisfies $\lambda(i) = 1$ and $T_{\mathrm{seq}}(p, a, i) = 0$ as well as $T_{\mathrm{join}}(\{p_1, p_2\}, i) = 0 = T_{\mathrm{fork}}(p_1, \{i, p_2\})$ for all $p, p_1, p_2 \in Q$ and $a \in \Sigma$. The BRAC $\mathcal{A}$ is *final-normalized* if there is a unique final state f and this state satisfies $\gamma(f) = 1$ and $T_{\mathrm{seq}}(f, a, q) = 0$ and $T_{\mathrm{fork}}(f, \{q_1, q_2\}) = 0 = T_{\mathrm{join}}(\{q_1, f\}, q_2)$ for all $q, q_1, q_2 \in Q$ and $a \in \Sigma$. If $\mathcal{A}$ is both initial- and final-normalized then $\mathcal{A}$ is said to be *normalized*.

Proposition 4.2. *Let $\mathbb{K}$ be a bisemiring, Σ an alphabet and $\mathcal{A}$ a BRAC over Σ with costs from $\mathbb{K}$. Then there is a normalized BRAC with the same behavior as $\mathcal{A}$. If $\mathcal{A}$ is of bounded depth, then also the normalized BRAC is of bounded depth.*

Proof. We show how to transform $\mathcal{A} = (Q, T_{\mathrm{seq}}, T_{\mathrm{fork}}, T_{\mathrm{join}}, \lambda, \gamma)$ into an initial-normalized BRAC. The BRAC $\mathcal{A}^{(I)}$ is defined as follows

- $Q^{(I)} = Q \dot{\cup} \{i\}$.
- $$T^{(I)}{}_{\mathrm{seq}}(p, a, q) = \begin{cases} T_{\mathrm{seq}}(p, a, q) & p, q \in Q, \\ \bigoplus_{r \in Q}[\lambda(r) \circ T_{\mathrm{seq}}(r, a, q)] & p = i,\ q \in Q, \\ 0 & \text{otherwise.} \end{cases}$$

- $T^{(I)}{}_{\mathrm{fork}}(p, \{p_1, p_2\}) = \begin{cases} T_{\mathrm{fork}}(p, \{p_1, p_2\}) & p, p_1, p_2 \in Q, \\ \bigoplus_{r\in Q}[\lambda(r) \circ T_{\mathrm{fork}}(r, \{p_1, p_2\})] & p = i,\ \ p_1, p_2 \in Q, \\ 0 & \text{otherwise.} \end{cases}$
- $T^{(I)}{}_{\mathrm{join}}(\{q_1, q_2\}, q) = \begin{cases} T_{\mathrm{join}}(\{q_1, q_2\}, q) & q_1, q_2, q \in Q, \\ 0 & \text{otherwise.} \end{cases}$
- $\lambda^{(I)}(p) = \begin{cases} 1 & p = i, \\ 0 & p \in Q, \end{cases}$ and $\gamma^{(I)}(q) = \begin{cases} 0 & q = i, \\ \gamma(q) & q \in Q. \end{cases}$

Clearly, $\mathcal{A}^{(I)}$ is initial-normalized with i as its unique initial state. Now we show $\mathcal{S}(\mathcal{A}^{(I)}) = \mathcal{S}(\mathcal{A})$. For a path $G = (V, E, \nu, \eta)$ of $\mathcal{A}$, we define a path $G' = (V, E, \nu', \eta)$ of $\mathcal{A}^{(I)}$: $\nu'(\mathrm{src}(G')) = i$ and $\nu'(v) := \nu(v)$ for all other $v \in V$. If G' is a path of $\mathcal{A}^{(I)}$, we set $\varrho(G) = G'$; otherwise, $\varrho(G)$ is undefined. Clearly, $\mathrm{lab}(G) = \mathrm{lab}(\varrho(G))$ and every path of $\mathcal{A}^{(I)}$ whose source is labeled with i is in the image of ϱ. If G' is a path of $\mathcal{A}^{(I)}$ starting with initial state i then we have

$$\mathrm{cost}(G') = \bigoplus_{G\in\varrho^{-1}(G')} \lambda(\nu(\mathrm{src}(G))) \circ \mathrm{cost}(G).$$

Let $\mathrm{PT}^0_t(\mathcal{A})$ be the collection of all paths of $\mathcal{A}$ with label t whose image under ϱ is undefined. Let $H \in \mathrm{PT}^0_t(\mathcal{A})$ and let $[H]$ be the collection of all paths of $\mathcal{A}$ that differ from H only in the label of their source. Then $H \in \mathrm{PT}^0_t(\mathcal{A})$ and $\tilde{H} \in [H]$ imply $\tilde{H} \in \mathrm{PT}^0_t(\mathcal{A})$ by the definition of ϱ. We fix a path $H \in \mathrm{PT}^0_t(\mathcal{A})$ and assume H starts with a sequential transition. So $H = H_1 \cdot H_2$ where $H_1 = p \xrightarrow{a} q$ for some $p, q \in Q$ and $a \in \Sigma$. Since $H \in \mathrm{PT}^0_t(\mathcal{A})$ we get by definition of ϱ and $\mathcal{A}^{(I)}$

$$T^{(I)}{}_{\mathrm{seq}}(i, a, q) = \bigoplus_{r\in Q} [\lambda(r) \circ T_{\mathrm{seq}}(r, a, q)] = 0.$$

Hence, we have

$$\begin{aligned} \bigoplus_{\tilde{H}\in[H]} \lambda(\nu(\mathrm{src}(\tilde{H}))) \circ \mathrm{cost}(\tilde{H}) &= \bigoplus_{r\in Q} [\lambda(r) \circ T_{\mathrm{seq}}(r, a, q) \circ \mathrm{cost}(H_2)] \\ &= \left[\bigoplus_{r\in Q} \lambda(r) \circ T_{\mathrm{seq}}(r, a, q)\right] \circ \mathrm{cost}(H_2) \\ &= 0. \end{aligned}$$

Similarly, we get the same result if H starts with a fork transition. Now, note that for $H \in \mathrm{PT}^0_t(\mathcal{A})$ the classes $[H]$ define a partition of $\mathrm{PT}^0_t(\mathcal{A})$. Thus:

$$\bigoplus_{H\in\mathrm{PT}^0_t(\mathcal{A})} \lambda(\nu(\mathrm{src}(H))) \circ \mathrm{cost}(H) = 0.$$

Therefore, it is sufficient for calculating $(\mathcal{S}(\mathcal{A}), t)$ to sum up only over the paths G with $\varrho(G) \in \mathrm{PT}(\mathcal{A}^{(I)})$. Hence we get for any $t \in \mathrm{SP}$

$$(\mathcal{S}(\mathcal{A}^{(I)}), t) = \bigoplus_{q\in Q}\left[\left(\bigoplus_{G':i \xrightarrow{t} q} \mathrm{cost}(G')\right) \circ \gamma^{(I)}(q)\right]$$

$$
\begin{aligned}
&= \bigoplus_{q\in Q}\Bigg[\Bigg(\bigoplus_{G\in\varrho^{-1}(G')} \lambda(\nu(\mathrm{src}(G)))\circ \mathrm{cost}(G)\Bigg)\circ\gamma(q)\Bigg]\\
&= \bigoplus_{r,q\in Q} \lambda(r)\circ \mathrm{cost}_{r,q}(t)\circ\gamma(q)\\
&= (\mathcal{S}(\mathcal{A}),t).
\end{aligned}
$$

Note that $\mathcal{A}^{(I)}$ does not have additional final states because $\gamma^{(I)}(i)=0$. We can now perform a similar transformation to obtain from $\mathcal{A}^{(I)}$ a final-normalized automaton. Since this transformation will not introduce any new initial states and transitions into i, the resulting BRAC will be normalized. $\square$

Next, we construct from two BRACs $\mathcal{A}_1$ and $\mathcal{A}_2$ a BRAC $\mathcal{A}$ with behavior $\mathcal{S}(\mathcal{A}) = \mathcal{S}(\mathcal{A}_1)\|\mathcal{S}(\mathcal{A}_2)$. At first, one would try to take $\mathcal{A}$ the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$, adding two states i and f as initial and final state, and, moreover, adding fork transitions $i\to\{p_1,p_2\}$ where p_i is initial in $\mathcal{A}_i$, and, similarly, adding join transitions $\{q_1,q_2\}\to f$ where q_i is final in $\mathcal{A}_i$. But this construction fails in general. We cannot concentrate the old entry costs of $\mathcal{A}_1$ and $\mathcal{A}_2$ in the new fork transitions $i\to\{p_1,p_2\}$ or in the entry cost of i because then the entry costs of $\mathcal{A}_1$ and $\mathcal{A}_2$ would not be multiplied in parallel anymore. But this is necessary for the behavior of $\mathcal{A}$. The construction can only be successful if all entry and leaving costs of $\mathcal{A}_1$ and $\mathcal{A}_2$ are either 1 or 0. Therefore, we assume in the proof of the next proposition the automata to be normalized. This can be done due to Proposition 4.2.

Proposition 4.3. *Let S_1 and S_2 be regular sp-series over an arbitrary bisemiring $\mathbb{K}$. Then $S_1\|S_2$ is regular.*

Proof. We give only a sketch of the proof. Let $\mathcal{A}_i$ be a normalized BRAC recognizing S_i for $i=1,2$. Moreover, let i_i and f_i be the unique initial and final state of $\mathcal{A}_i$, respectively. We construct a new automaton $\mathcal{A}$ by taking the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$, adding two new states i and f and, moreover, a fork $i\to_1\{i_1,i_2\}$ and a join $\{f_1,f_2\}\to_1 f$. We put $\lambda(i)=1$ and $\gamma(f)=1$. All other entry and leaving costs are equal to 0. Every path G in $\mathcal{A}$ from i to f is of the form $G=G_1\|_{i,f}G_2$ where G_i is a path in $\mathcal{A}_i$ from i_i to f_i for $i=1,2$. Using distributivity of $\diamond$ over $\oplus$, it is an easy exercise to show $(\mathcal{S}(\mathcal{A}),t)=(S_1\|S_2,t)$ for all $t\in\mathrm{SP}$. $\square$

The rest of this section is devoted to the closure of the class of regular series under sequential multiplication and sequential iteration. These facts have well-known counterparts in the theory of nondeterministic finite automata and it is tempting to believe that constructions familiar from this theory work here as well. But, as already observed for sp-languages by Lodaya and Weil, this is not the case. The following example shows that the obvious variant of the classical construction for the product does not yield the correct result. We call a path *successful* if it is a path from an initial to a final state with cost unequal to zero.

Example 4.4. We work with the Boolean bisemiring $\mathbb{B}=(\{0,1\},\vee,\wedge,\wedge,0,1)$, i.e. in the setting of sp-languages. Suppose that the BRAC $\mathcal{A}_1$ consists of the following transitions:

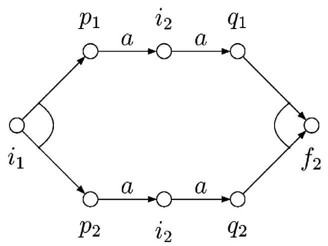

Fig. 2. A problematic path in the classical product construction.

a fork transition $i_1 \to_1 \{p_1, p_2\}$ and sequential transitions $p_1 \xrightarrow{a}_1 f_1$ and $p_2 \xrightarrow{a}_1 f_1$. Further, i_1 is the only initial state and f_1 the only final state. The BRAC $\mathcal{A}_2$ consists of the following transitions: a join transition $\{q_1, q_2\} \to_1 f_2$ and sequential transitions $i_2 \xrightarrow{a}_1 q_1$ and $i_2 \xrightarrow{a}_1 q_2$. The unique initial state is i_2 while f_2 is the only accepting state. Then the classical construction suggests to consider the BRAC consisting of all the transitions mentioned so far and, in addition, in particular sequential transitions $p_1 \xrightarrow{a}_1 i_2$ and $p_2 \xrightarrow{a}_1 i_2$. Fig. 2 gives one successful path of the resulting BRAC, its label is $(aa)\|(aa)$. Since the language of both $\mathcal{A}_1$ and $\mathcal{A}_2$ is empty, the composition should not allow any successful path whatsoever.

The problem in the example above is that the newly constructed automaton can switch from $\mathcal{A}_1$ into $\mathcal{A}_2$ independently in parallel subpaths. Lodaya and Weil showed that this problem does not arise when one restricts to “behaved automata”. Then they show that one can transform any branching automaton into an equivalent behaved one. We proceed differently giving a direct construction for the sequential product. More precisely, we “send a signal” from the initial state along the path. In fork transitions, this signal is only propagated along one branch. In order not to duplicate paths, the signal is sent to the “smaller” of the two states that arise from the fork transition. [4] Further, the newly constructed BRAC can only switch from $\mathcal{A}_1$ into $\mathcal{A}_2$ in the presence of this signal, and in any successful path, the signal has to be present at the final state.

Proposition 4.5. *Let $S_1, S_2 \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be two regular sp-series. Then $S_1 \cdot S_2$ is regular.*

Proof. Let $\mathcal{A}_i = (Q_i, T_{i\,\mathrm{seq}}, T_{i\,\mathrm{fork}}, T_{i\,\mathrm{join}}, \lambda_i, \gamma_i)$ be a BRAC with $\mathcal{S}(\mathcal{A}_i) = S_i$ for $i = 1, 2$. We fix an arbitrary linear order $\leqslant$ on the set Q_1 of the states of $\mathcal{A}_1$. The construction of a BRAC $\mathcal{A}$ recognizing $S_1 \cdot S_2$ is done in two steps. At first, we construct an automaton $\mathcal{A}'$ with $\mathcal{S}(\mathcal{A}') = \mathcal{S}(\mathcal{A}_1)$ as follows:

- $Q' = Q \times \{0, 1\}$.

[4] This is actually the reason why we have to assume that these two states are different, i.e. that we work with sets in the definition of fork- and join-transitions and not with multisets as Lodaya and Weil do.

- $T'_{\text{seq}}((p,x),a,(q,y)) = \begin{cases} T_{\text{seq}}(p,a,q) & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$
- $T'_{\text{fork}}((p,x),\{(p_1,x_1),(p_2,x_2)\}) = \begin{cases} T_{\text{fork}}(p,\{p_1,p_2\}) & \text{if } p_1 < p_2, x = x_1, x_2 = 0, \\ 0 & \text{otherwise.} \end{cases}$
- $T'_{\text{join}}(\{(q_1,x_1),(q_2,x_2)\},(q,x)) = \begin{cases} T_{\text{join}}(\{q_1,q_2\},q) & \text{if } x = x_1, x_2 = 0, \\ 0 & \text{otherwise.} \end{cases}$
- $\lambda'(p,x) = \begin{cases} \lambda(p) & \text{if } x = 1, \\ 0 & \text{otherwise,} \end{cases}$ and $\gamma'(q,x) = \begin{cases} \gamma(q) & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases}$

In the sequel, we refer to the second component of a state of $\mathcal{A}'$ as a signal. It is either 0 or 1. By induction we get: If $G' : (p,x) \xrightarrow{t} (q,y)$ is a path of $\mathcal{A}'$ then $x = y$. We define the following sets of paths:

- $\mathrm{PT}_{p,q}(\mathcal{A})$ is the set of paths from p to q in $\mathcal{A}$.
- $\mathrm{PT}^0_{p,q}(\mathcal{A}')$ and $\mathrm{PT}^1_{p,q}(\mathcal{A}')$ are the sets of paths in $\mathcal{A}'$ from $(p,0)$ to $(q,0)$ and from $(p,1)$ to $(q,1)$, respectively.

We define $\varrho_0 : \mathrm{PT}^0_{p,q}(\mathcal{A}') \longrightarrow \mathrm{PT}_{p,q}(\mathcal{A})$ by dropping the signals of the states of a path $G' \in \mathrm{PT}^0_{p,q}(\mathcal{A}')$. Note that all states of such a path G' have signal 0. Then ϱ_0 is a bijective mapping preserving labels and costs. Next, we define $\varrho_1 : \mathrm{PT}^1_{p,q}(\mathcal{A}') \longrightarrow \mathrm{PT}_{p,q}(\mathcal{A})$ also by dropping the signals of the states. Note that paths from $\mathrm{PT}^1_{p,q}(\mathcal{A}')$ can well contain nodes with signal 0. Thus, in order to show that ϱ_1 is bijective and preserves labels and costs, one uses the corresponding result on ϱ_0. Considering that all initial and final states of $\mathcal{A}'$ have signal 1 it is clear by the definition of λ' and γ' that $\mathcal{S}(\mathcal{A}') = \mathcal{S}(\mathcal{A}_1)$.

Due to Proposition 4.2 we may assume both $\mathcal{A}_1$ and $\mathcal{A}_2$ to be normalized. Then $\mathcal{A}'$ is also normalized by definition. Now we construct a BRAC $\mathcal{A}$ that realizes $S_1 \cdot S_2$ as follows. We take the disjoint union of $\mathcal{A}'$ and $\mathcal{A}_2$ but replace the unique final state $(f_1, 1)$ of $\mathcal{A}'$ by the unique initial state i_2 of $\mathcal{A}_2$, and call it s. The transitions and their costs carry over from $\mathcal{A}'$ and $\mathcal{A}_2$ to $\mathcal{A}$ as far as the state s is not involved. All transitions ending in $(f_1, 1)$ turn to transitions with same cost ending in s, and dually for the transitions starting in i_2. We put $\lambda(s) = \gamma(s) = 0$. The initial state of $\mathcal{A}$ is the unique initial state $(i_1, 1)$ of $\mathcal{A}'$, the final state of $\mathcal{A}$ is the state f_2 of $\mathcal{A}_2$ where their entry and leaving costs carry over, respectively.

For $p' \in Q'$ and $q \in Q_2$ with $p', q \neq s$ let G be an arbitrary path of $\mathcal{A}$ from p' to q. We show that each such path G decomposes into $G = G' \cdot G''$ with $G' : p' \to s$ a path only in $\mathcal{A}'$ and $G'' : s \to q$ a path only in $\mathcal{A}_2$. Indeed, G cannot be atomic because otherwise $p = s$ or $q = s$. If $G = G_1 \cdot \ldots \cdot G_n$ allows a sequential decomposition then there is an $i \in \{1, \ldots, n\}$ such that $G_1 \cdot \ldots \cdot G_i$ is a path in $\mathcal{A}'$ from p' to $(f_1, 1) = s$ and $G_{i+1} \cdot \ldots \cdot G_n$ is a path from $s = i_2$ to q in $\mathcal{A}_2$. Otherwise, there would have to be either a path from some $r_2 \in Q_2$ to some $r' \in Q'$ contradicting to the definition of $\mathcal{A}$. Or there would be $\tilde{p} \in Q'$ and $\tilde{q} \in Q_2$ and a $j \in \{1, \ldots, n\}$ such that $G_j = H_1 \|_{\tilde{p},\tilde{q}} H_2$ for some paths H_1, H_2 in $\mathcal{A}$. But then $H_i : \tilde{p}_i \to \tilde{q}_i$ for $\tilde{p}_i = (p_i, x_i) \in Q'$ and $\tilde{q}_i \in Q_2$ with $i = 1, 2$. By induction there would be a factorization $H_i = H_i' \cdot H_i''$ with $H_i' : \tilde{p}_i \to s$ in $\mathcal{A}'$ and $H_i'' : s \to \tilde{q}_i$ in $\mathcal{A}_2$ for $i = 1, 2$. Assume $p_1 < p_2$. Then $x_2 = 0$. But this contradicts $H_2' : (p_2, x_2) \to (f_1, 1)$ being a path. If the path G allows a parallel decomposition $G_1 \|_{p',q} G_2$ we would get a contradiction in a similar way. Hence, every path G from p' to q decomposes sequentially in the given form.

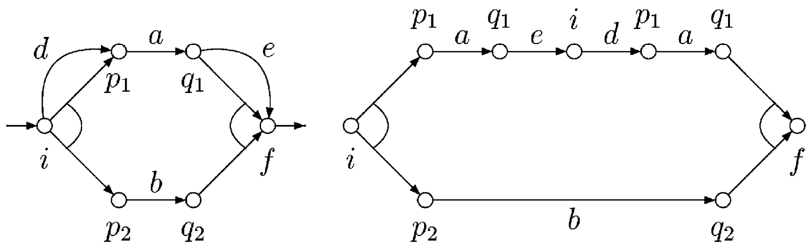

Fig. 3. A BRAC and a problematic path in the classical construction.

Using normalization, commutativity of $\oplus$ and distributivity, we get for all $t \in \mathrm{SP}$:

$$
\begin{aligned}
(\mathcal{S}(\mathcal{A}), t) &= \bigoplus_{G:(i_1,1) \xrightarrow{t} f_2} \mathrm{cost}(G) \\
&= \bigoplus_{\substack{t=t_1 \cdot t_2 \\ G_1:(i_1,1) \xrightarrow{t_1} s \\ G_2: s \xrightarrow{t_2} f_2}} \mathrm{cost}(G_1) \circ \mathrm{cost}(G_2) \\
&= \bigoplus_{t=t_1 \cdot t_2} \left(\bigoplus_{G_1:(i_1,1) \xrightarrow{t_1} s} \mathrm{cost}(G_1) \right) \circ \left(\bigoplus_{G_2: s \xrightarrow{t_2} f_2} \mathrm{cost}(G_2) \right) \\
&= \bigoplus_{t=t_1 \cdot t_2} (\mathcal{S}(\mathcal{A}_1), t_1) \circ (\mathcal{S}(\mathcal{A}_2), t_2) \\
&= (S_1 \cdot S_2, t).
\end{aligned}
$$

This concludes the proof. $\square$

Similarly to the sequential composition, the classical construction for the sequential iteration suggests itself—and yields an incorrect result as the following example shows.

Example 4.6. We work with the Boolean bisemiring $\mathbb{B} = (\{0, 1\}, \vee, \wedge, \wedge, 0, 1)$, i.e. in the setting of sp-languages. Consider the BRAC from Fig. 3 (left) where we omitted the costs; any transition depicted has cost 1 and no further transitions have nonzero cost. The support of the recognized sp-series is $\{a \| b, dae\}$. The classical construction for the sequential iteration tells us to add, among other transitions, one of the form $q_1 \xrightarrow{e} i$ since there is a sequential transition $q_1 \xrightarrow{e} f$ in the BRAC in consideration. But then we get the path depicted in Fig. 3 (right) whose label is $(aeda) \| b$ which does not belong to the sequential iteration of the sp-language generated by the BRAC we started with.

Lodaya and Weil's solution is, again, to use behaved automata. Our direct construction sends not just one, but two signals. These two signals travel along different ways: whenever they can separate in a fork transition, they do so. Then the newly constructed automaton is allowed to jump from the final state to the initial state only in case both signals are present. As before, in any successful path, both signals are present in the first and the last state.

We introduce a notion needed in the next proof. Let G be a path of some BRAC $\mathcal{A}$ and $G = G_1 \cdot \ldots \cdot G_n$ the sequential decomposition of G. If a state p is the label of the source or the sink of one of the G_i $(i = 1, \ldots, n)$ then we say p occurs on the *upper level* of G.

Proposition 4.7. *If $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is regular, then the sequential iteration S^+ is regular.*

Proof. Let $\mathcal{A}$ be a BRAC recognizing S. We assume an arbitrary but fixed linear order $\leqslant$ on the set of states Q of $\mathcal{A}$. Again we do the construction of a BRAC recognizing S^+ in two steps. At first, we construct an automaton $\mathcal{A}'$ with the same behavior as $\mathcal{A}$ similar to the construction in the last proof, but this times with two signals for the state:

- $Q' = Q \times \{0, 1\}^2$.
- $T'_{\mathrm{seq}}((p, x, x'), a, (q, y, y')) = \begin{cases} T_{\mathrm{seq}}(p, a, q) & \text{if } x = y, x' = y', \\ 0 & \text{otherwise.} \end{cases}$
- $T'_{\mathrm{fork}}((p, x, x'), \{(p_1, x_1, x'_1), (p_2, x_2, x'_2)\})$
  $= \begin{cases} T_{\mathrm{fork}}(p, \{p_1, p_2\}) & \text{if } p_1 < p_2, x = x_1, x'_1 = 0, x_2 = 0, x' = x'_2 \\ 0 & \text{otherwise.} \end{cases}$
- $T'_{\mathrm{join}}(\{(q_1, x_1, x'_1), (q_2, x_2, x'_2)\}, (q, x, x'))$
  $= \begin{cases} T_{\mathrm{join}}(\{q_1, q_2\}, q) & \text{if } x = x_1, x'_1 = 0, x_2 = 0, x' = x'_2, \\ 0 & \text{otherwise.} \end{cases}$
- $\lambda'(p, x, x') = \begin{cases} \lambda(p) & \text{if } x = x' = 1, \\ 0 & \text{otherwise,} \end{cases}$ and $\gamma'(q, x, x') = \begin{cases} \gamma(q) & \text{if } x = x' = 1, \\ 0 & \text{otherwise.} \end{cases}$

We refer to the second and third component of a state as the signals of this state. Similar to the proof of Proposition 4.5 we get $\mathcal{S}(\mathcal{A}') = \mathcal{S}(\mathcal{A})$. Moreover, if $G' : (p, x, x') \to (q, y, y')$ is a path of $\mathcal{A}'$ then $x = y$ and $x' = y'$.

Due to Proposition 4.2 $\mathcal{A}$ can be assumed to be normalized. Then $\mathcal{A}'$ is normalized too. Let i and f denote the unique initial and final state of $\mathcal{A}$. Then $i' = (i, 1, 1)$ and $f' = (f, 1, 1)$ are the unique initial and final state of $\mathcal{A}'$, respectively. Now we construct from $\mathcal{A}'$ a BRAC $\mathcal{A}^+$ as follows. The states of $\mathcal{A}^+$ are the same as those of $\mathcal{A}'$. Moreover, every transition of $\mathcal{A}'$ is also a transition of $\mathcal{A}^+$. Now we add for every sequential transition $p' \xrightarrow{a}_k f'$ of $\mathcal{A}'$ a transition $p' \xrightarrow{a}_k i'$ in $\mathcal{A}^+$, and for every join transition $\{q'_1, q'_2\} \to_l f'$ of $\mathcal{A}'$ a further join $\{q'_1, q'_2\} \to_l i'$ in $\mathcal{A}^+$. The entry and leaving costs remain the same. Therefore, $\mathcal{A}^+$ has still the unique initial state i' and the unique final state f'.

At first we consider an arbitrary parallel path $H = H_1 \|_{p',q'} H_2$ in $\mathcal{A}^+$ starting with a fork $p' \to \{p'_1, p'_2\}$. Then none of the states in H_1 or H_2 carries signal $(1, 1)$. In particular, neither i' nor f' occur in H_1 or H_2. Therefore, H_1 and H_2 are paths in $\mathcal{A}'$ too. Now, let $\mathrm{PT}_{i',i'}(\mathcal{A}^+)$ be the set of all paths in $\mathcal{A}^+$ from i' to i' such that i' does appear only as the label of the source and the sink of the path. By $\mathrm{PT}_{i',f'}(\mathcal{A}')$ we denote the set of all paths in $\mathcal{A}'$ going from i' to f'. The mapping $\phi : \mathrm{PT}_{i',i'}(\mathcal{A}^+) \longrightarrow \mathrm{PT}_{i',f'}(\mathcal{A}')$ is defined as follows. It maps $G \in \mathrm{PT}_{i',i'}(\mathcal{A}^+)$ to $G' \in \mathrm{PT}_{i',f'}(\mathcal{A}')$ by labeling the sink of G with f' instead of i'. We show that ϕ is well defined. Indeed, because i' does appear in G only as the label of the source and the sink all but the last transition of G are also transitions in $\mathcal{A}'$. The last transition is either of the form $p' \xrightarrow{a}_k i'$ or is a join $\{q_1', q_2'\} \to_l i'$ for some states p', q_1', q_2' and $k, l \in \mathbb{K}$. But then $p' \xrightarrow{a}_k f'$, respectively, $\{q_1', q_2'\} \to_l f'$ are transitions of $\mathcal{A}'$ by definition of $\mathcal{A}^+$. Hence, $\phi(G) = G'$ is well defined. Now, by the definition of $\mathcal{A}^+$ and ϕ, it follows immediately that ϕ is bijective and preserves labels and costs.

Now consider a path G from i' to f' in $\mathcal{A}^+$. We state that G is of the form $G = G_1 \cdot \ldots \cdot G_n$ for some $n \geqslant 1$ where $G_j \in \mathrm{PT}_{i',i'}(\mathcal{A}^+)$ for $j = 1, \ldots, n-1$ and $G_n \in \mathrm{PT}_{i',f'}(\mathcal{A}')$ is a path from i' to f' both in $\mathcal{A}^+$ and in $\mathcal{A}'$. Indeed, either state i' appears only once and then G is also a path in $\mathcal{A}'$. Or the state i' appears more than once in G. Since i' cannot occur in a subpath of a parallel path, as stated above, it appears only at the upper level of G. Hence, G allows the decomposition given above. Vice versa, every path of this form is of course a path from i' to f' in $\mathcal{A}^+$. Now we get for any $t \in \mathrm{SP}$:

$$\begin{aligned}
(\mathcal{S}(\mathcal{A}^+), t) &= \bigoplus_{G: i' \xrightarrow{t} f'} \mathrm{cost}(G) \\
&= \bigoplus_{\substack{G_1 \cdot \ldots \cdot G_n: \\ i' \xrightarrow{t_1} i' \xrightarrow{t_2} \ldots \xrightarrow{t_n} f'}} \mathrm{cost}(G_1) \circ \ldots \circ \mathrm{cost}(G_n) \\
&= \bigoplus_{n \geqslant 1} \bigoplus_{t = t_1 \cdot \ldots \cdot t_n} \bigoplus_{\substack{G_1 : i' \xrightarrow{t_1} i' \\ \ldots \\ G_n : i' \xrightarrow{t_n} f'}} \mathrm{cost}(G_1) \circ \ldots \circ \mathrm{cost}(G_n) \\
&= \bigoplus_{n \geqslant 1} \bigoplus_{t = t_1 \cdot \ldots \cdot t_n} \bigoplus_{G_1, \ldots, G_n} \mathrm{cost}(\phi(G_1)) \circ \ldots \circ \mathrm{cost}(\phi(G_{n-1})) \circ \mathrm{cost}(G_n) \\
&= \bigoplus_{n \geqslant 1} \bigoplus_{t = t_1 \cdot \ldots \cdot t_n} (\mathcal{S}(\mathcal{A}'), t_1) \circ \ldots \circ (\mathcal{S}(\mathcal{A}'), t_n) \\
&= (S^+, t).
\end{aligned}$$

The step before the last one is due to the distributivity of $\mathbb{K}$ and the normalization of $\mathcal{A}'$. Hence, $\mathcal{A}^+$ recognizes the sequential iteration S^+. □

Now we can state the main theorem of this section.

Theorem 4.8. *Let $\mathbb{K}$ be an arbitrary bisemiring. Every sequential-rational sp-series $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is regular. Moreover, S is recognized by a normalized BRAC $\mathcal{A}$ of bounded depth.*

Proof. All monomials are obviously regular. By Propositions 4.1, 4.5, 4.3 and 4.7 the regular sp-series are closed under sum, sequential and parallel product as well as under sequential iteration. To show closure under scalar products, let $k \in \mathbb{K}$ and let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be recognized by a normalized BRAC $\mathcal{A}$ with unique initial state i. We define $\mathcal{A}'$ being equal to $\mathcal{A}$ with the exception of the entry cost $\lambda'(i) := k = k \circ \lambda(i)$. Then $\mathcal{A}'$ recognizes the sp-series $k \cdot S$. A similar proof holds for the series $S \cdot k$ changing the leaving instead of the entry cost. Hence, every sequential-rational sp-series is regular.

By Proposition 3.5 the sequential-rational sp-series S has bounded width. By Corollary 3.9, S can be recognized by a BRAC of bounded depth. Proposition 4.2 shows that a BRAC can be normalized such that the property of bounded depth is preserved. □

Note, that a short analysis of the proofs of the propositions of this section shows that all constructions preserve the bounded depth of the automata.

5. From regular and bounded depth to sequential-rational

Before we prove the converse of Theorem 4.8, we need the notion of a fork-acyclic BRAC. If $G = G_1 \|_{p,q} G_2$ is the parallel decomposition of a path G and f denotes the starting fork transition of G, j the finishing join transition of G, then we say that (f, j) is a *matched pair*. For two matched pairs (f, j) and (f', j') we put $(f, j) < (f', j')$ if there exists a parallel path G starting with f' and ending with j' that contains (f, j) as a matched pair of a proper subpath of G. If for a BRAC $\mathcal{A}$ the relation $<$ on the set of all matched pairs is irreflexive, then $\mathcal{A}$ is called *fork-acyclic*. This is the same as the fork acyclicity in the sense of Lodaya and Weil [15]. In the following proof, we will use the obvious fact that any BRAC of bounded depth is fork-acyclic.

Theorem 5.1. *Let $\mathcal{A}$ be a BRAC of bounded depth with costs from a bisemiring $\mathbb{K}$. Then $\mathcal{S}(\mathcal{A})$ is sequential-rational.*

The behavior of general BRACs cannot be captured by rational operations even in the case of the Boolean bisemiring. The obvious idea to allow in addition the parallel iteration does not give the desired result: If the sp-series S is generated by the rational operations (including the parallel iteration), then there exists $n \in \mathbb{N}$ with the following property: if $t \in \mathrm{SP}$ is in the support of S, then t can be constructed using at most n alternations of $\cdot$ and $\|$. On the other hand, there is a regular sp-series S with $\operatorname{supp} S = \mathrm{SP}$ which therefore does not have this property (cf. Lodaya and Weil [15, Section 5] for a more elaborated example).

Proof. Let $\mathcal{A}$ be a BRAC of bounded depth. By Proposition 4.2 we can assume $\mathcal{A}$ to be normalized. Since $\mathcal{A}$ is of bounded depth, it is fork-acyclic. In the sequel f denotes a fork and j a join transition. By M we denote the set of all matched pairs of $\mathcal{A}$. Let $G = G_1 \cdot \ldots \cdot G_n$ be a path of $\mathcal{A}$ with its sequential decomposition ($n \geqslant 1$). Then we say that a matched pair (f, j) is used at the upper level of G if there is a G_i that starts with f and ends with j.

Clearly, $<$ is transitive. Since $<$ is irreflexive, the reflexive closure $\leqslant$ of $<$ is antisymmetric. We fix an arbitrary linear extension $\sqsubseteq$ of the partial order $\leqslant$ on the set of matching pairs M and consider the linearly ordered set $(M, \sqsubseteq)$. Let $J \subseteq M$ and $p, q \in Q$ be states of $\mathcal{A}$. We denote by $S^J_{p,q}$ the series with

$$(S^J_{p,q}, t) = \bigoplus_{G:p \xrightarrow{t} q} \mathrm{cost}(G),$$

where $t \in \mathrm{SP}$ and the paths G, over which the sum extends, are such that only matched pairs of J are used in G. We will show that $S^J_{p,q}$ is sequential-rational for any initial segment $J = \{(f', j') \mid (f', j') \sqsubseteq (f, j)\}$ for some $(f, j) \in M$. We proceed by induction over $|J|$.

At first let $|J| = 0$, i.e. no forks and joins are used. Then $S^{\emptyset}_{p,q}$ is a regular series where the parallel product is not used, i.e. it is a regular word series with values from the semiring $(K, \oplus, \circ, 0, 1)$. By a result of Schützenberger [20] we know that $S^{\emptyset}_{p,q}$ is rational as a word series and, therefore, also sequential-rational as an sp-series. Now we assume $J = \{(f', j') \mid (f', j') \sqsubseteq (f, j)\}$ with $f : r \to_k \{r_1, r_2\}$ and $j : \{s_1, s_2\} \to_l s$. We define the sp-series $S(f, j)$ for every $t \in \mathrm{SP}$ by

$$(S(f, j), t) = \bigoplus_{G:r \xrightarrow{t} s} \mathrm{cost}(G)$$

where the sum ranges over all parallel paths $G = H_1 \|_{r,s} H_2$ starting with fork f, ending with join j and having label t. But H_1 and H_2 contain only matched pairs from $J' = J \setminus \{(f, j)\}$ by fork acyclicity of $\mathcal{A}$ and the definition of $\sqsubseteq$. Note that $J' \subset J$, and that J' is also an initial segment of $(M, \sqsubseteq)$. Thus, we have

$$S(f, j) = k \cdot \left(\left[S^{J'}_{r_1,s_1} \| S^{J'}_{r_2,s_2} \right] + \left[S^{J'}_{r_1,s_2} \| S^{J'}_{r_2,s_1} \right] \right) \cdot l$$

and the sp-series $S^{J'}_{r_\alpha,s_\beta}$ with $\{\alpha, \beta\} \subseteq \{1, 2\}$ are sequential-rational by induction hypothesis. Hence, $S(f, j)$ is sequential-rational.

Now, consider the sp-series $S^J_{p,q}$ again. Since $\mathcal{A}$ is fork acyclic, all paths from p to q with matched pairs only from J use the maximal element (f, j) of J only at the upper level. Therefore, we have (for $p \neq r$, $s \neq q$, and $r = s$): [5]

$$S^J_{p,q} = S^{J'}_{p,q} + S^{J'}_{p,r} \cdot S(f, j)^+ \cdot S^{J'}_{s,q} + S^{J'}_{p,r} \cdot S(f, j)^+ \cdot \left(S^{J'}_{s,r} \cdot S(f, j)^+ \right)^+ \cdot S^{J'}_{s,q}$$

The first sp-series $S^{J'}_{p,q}$ of this sum covers all paths that do not use (f, j) at the upper level. The second one covers all paths $G_1 \cdot G_2 \cdot G_3$ such that (f, j) appears as matched pair at the upper level in G_2, but neither in G_1 nor in G_3, and no other matched pair appears at the upper level of G_2. Thus, in these paths we have exactly one sequence of consecutive "bubbles" from $S(f, j)$. The third series covers all paths where we have more than one such sequence of consecutive "bubbles" from $S(f, j)$. The sp-series $S(f, j)$ is sequential-rational as seen before. All other sp-series appearing on the right hand side of the above equation are sequential-rational by the induction hypothesis because J' is an initial segment properly contained in J. Therefore, $S^J_{p,q}$ is sequential-rational itself. Since $\mathcal{A}$ is assumed to

[5] The other cases require some other summands like $S^{J'}_{p,r} \cdot S(f, j)^+$ in case $r = s = q$.

be normalized with initial state i and final state f, we have $\mathcal{S}(\mathcal{A}) = S^M_{i,f}$. M is an initial segment of $(M, \sqsubseteq)$. Hence, $\mathcal{S}(\mathcal{A})$ is sequential-rational. □

The special case $\mathbb{K} = \mathbb{B}$ was shown by Lodaya and Weil [15]. Their proof uses a nested induction which we simplified here to just one induction along the linear order of matched pairs.

Now we can prove the main theorem about regular and sequential-rational sp-series.

Theorem 5.2. *Let $\mathbb{K}$ be an arbitrary bisemiring and $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$. The following are equivalent*:
(1) *S is sequential-rational*.
(2) *S is recognized by a BRAC of bounded depth*.
(3) *S is regular and has bounded width*.

Proof. Due to Theorem 4.8 (1) implies (2). By Theorem 5.1 (2) implies (1). Statements (2) and (3) are equivalent by Corollary 3.9. □

By putting $\mathbb{K} = \mathbb{B}$, we get as a consequence of the last theorem the result of Lodaya and Weil [15].

Corollary 5.3. *Let $L \subseteq$ SP be a language of finite sp-posets. Then the following are equivalent*:
(1) *L is a sequential-rational language*.
(2) *L is recognized by a branching automaton of bounded depth*.
(3) *L is regular and has bounded width*.

References

[1] J. Berstel, C. Reutenauer, Recognizable formal power series on trees, Theoret. Comput. Sci. 18 (1982) 115–148.
[2] J. Berstel, C. Reutenauer, Rational Series and Their Languages, EATCS Monographs on Theoretical Computer Science, Vol. 12, Springer, Berlin, 1988.
[3] M. Droste, P. Gastin, The Kleene–Schützenberger theorem for formal power series in partially commuting variables, Inform. and Comput. 153 (1999) 47–80.
[4] M. Droste, P. Gastin, On aperiodic and star-free formal power series in partially commuting variables, in: Formal Power Series and Algebraic Combinatorics (Moscow 2000), Springer, Berlin, 2000, pp. 158–169.
[5] M. Droste, D. Kuske, Skew and infinitary formal power series, ICALP 2003, Lecture Notes in Computer Science, Vol. 2719, Springer, Berlin, 2003, pp. 426–438.
[6] M. Droste, Ch. Pech, H. Vogler, A Kleene theorem for weighted tree automata, Theory of Comput. Systems (2003), accepted for publication.
[7] J.L. Gischer, The equational theory of pomsets, Theoret. Comp. Sci. 61 (1988) 199–224.
[8] J. Grabowski, On partial languages, Ann. Soc. Math. Polon. Ser. IV: Fund. Inform. 4 (2) (1981) 427–498.
[9] S.E. Kleene, Representations of events in nerve nets and finite automata, in: C.E. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, Princeton, 1956, pp. 3–42.
[10] W. Kuich, Formal power series over trees, in: Proc. Third Internat. Conf. Developments in Language Theory, Aristotle University of Thessaloniki, 1997, pp. 60–101.

[11] W. Kuich, A. Salomaa, Semirings, Automata, Languages, EATCS Monographs on Theoretical Computer Science, Vol. 5, Springer, Berlin, 1986.
[12] D. Kuske, Towards a language theory for infinite N-free pomsets, Theoret. Comput. Sci. 299 (2003) 347–386.
[13] D. Kuske, I. Meinecke, Branching automata with costs—a way of reflecting parallelism in costs, Technical Report MATH-AL-4-2003, TU Dresden, March 2003. See www.math.tu-dresden.de/∼meinecke.
[14] D. Kuske, I. Meinecke, Branching automata with costs—a way of reflecting parallelism in costs, CIAA 2003, Lecture Notes in Computer Science, Vol. 2759, Springer, Berlin, 2003, pp. 150–162.
[15] K. Lodaya, P. Weil, Series-parallel languages and the bounded-width property, Theoret. Comput. Sci. 237 (2000) 347–380.
[16] K. Lodaya, P. Weil, Rationality in algebras with a series operation, Inform. and Comput. 171 (2001) 269–293.
[17] Ch. Pech, Kleene's theorem for weighted tree-automata, A. Lingas, B.J. Nilsson (Eds.), Fundamentals of Computation Theory, Lecture Notes in Computer Science, Vol. 2751, Springer, Berlin, 2003, pp. 387–399.
[18] Ch. Pech, Kleenes Theorem für Formale Baumreihen, Dissertation, TU Dresden, 2003.
[19] A. Salomaa, M. Soittola, Automata-Theoretic Aspects of Formal Power Series, Texts and Monographs in Computer Science, Springer, Berlin, 1978.
[20] M.P. Schützenberger, On the definition of a family of automata, Inform. and Control 4 (1961) 245–270.