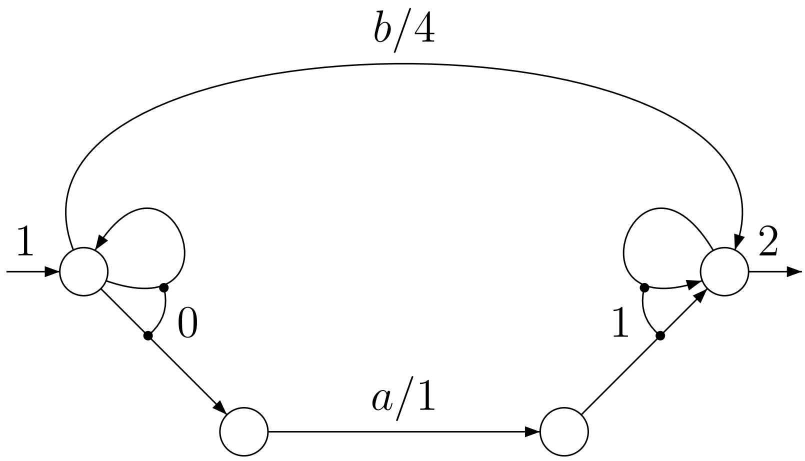
Ingmar Meinecke

Weighted Branching Automata

Combining Concurrency and Weights



DISSERTATION

Weighted Branching Automata
Combining Concurrency and Weights

Dissertation

zur Erlangung des akademischen Grades

Doctor rerum naturalium

(Dr. rer. nat.)

vorgelegt

der Fakultät Mathematik und Naturwissenschaften

der Technischen Universität Dresden

von

Dipl.-Math. Ingmar Meinecke

geboren am 27. August 1973 in Berlin

Gutachter : Prof. Dr. M. Droste (Leipzig)
Prof. Dr. P. Gastin (Paris)
Dr. P. Weil (Directeur de Recherches, Bordeaux)

Eingereicht am: 29. Juni 2004

Tag der Disputation: 14. Dezember 2004

Für meine Mutter

Mein junger Sohn fragt mich: Soll ich Mathematik lernen?
Wozu, möchte ich sagen. Dass zwei Stück Brot mehr ist als eines
Das wirst du auch so merken.
Mein junger Sohn fragt mich: Soll ich Französisch lernen?
Wozu, möchte ich sagen. Dieses Reich geht unter. Und
Reibe du nur mit der Hand den Bauch und stöhne
Und man wird dich schon verstehen.
Mein junger Sohn fragt mich: Soll ich Geschichte lernen?
Wozu, möchte ich sagen. Lerne du deinen Kopf in die Erde stecken
Da wirst du vielleicht übrigbleiben.

Ja, lerne Mathematik, sage ich
Lerne Französisch, lerne Geschichte!

Bertolt Brecht

Ingmar Meinecke

Weighted Branching Automata

Combining Concurrency and Weights

DISSERTATION

Ingmar Meinecke
Universität Leipzig
Institut für Informatik
PF 100920
D-04009 Leipzig

meinecke@informatik.uni-leipzig.de

Printed at *Universitätsdruckzentrum Leipzig*.

Acknowledgements

First of all, I would like to thank my supervisor Manfred Droste and my colleague Dietrich Kuske. Actually, the starting point of this dissertation was the question, raised by Manfred Droste, whether sequential-parallel posets may be equipped with weights. Subsequently, Dietrich Kuske got me on the track of branching automata. Both of them introduced me to the subject of formal language and automata theory. They taught me a lot about mathematical beauty, the importance of strict formalisms, and the persistency necessary to ask questions and find answers.

I am grateful to the Institute of Algebra at Dresden University of Technology. Here, I found a great place to work, many gifted people, and an open atmosphere filled with talks and discussions. Especially, I like to thank Ingrid Reim for helping me to overcome all the problems of everyday life at the institute.

I am indebted to *Deutsche Forschungsgemeinschaft* (DFG) for supporting me financially within the scope of the PhD programme 334 "Specifications of discrete processes and systems of processes by operational models and logics".

In particular, I thank my friends Karsten, Thierry, and Jan. I thank them for believing in me, for their challenges and their patience, and for turning me into quite a man not a mathematician only. Moreover, I thank Jan once more for taking the effort to improve my English.

Last but not least, I thank my good luck. There are so many gifted people. Most of them never get a chance to develop their capabilities. Many of them do not even know about their skills and giftedness. I hope this will change some day. By now I like to take my hat off to all who contributed by their labour to my opportunity to have written this thesis.

Dresden, June 2004 Ingmar Meinecke

Contents

Acknowledgements **vii**

1 Introduction **1**

2 Sequential-Parallel Posets and Bisemirings **11**
2.1 Sequential-Parallel Posets 11
2.2 Bisemirings 16

3 Weighted Branching Automata and Formal Series **25**
3.1 Weighted Branching Automata and their Behavior 25
3.1.1 The Behavior of Weighted Branching Automata . . 28
3.1.2 Examples of Weighted Branching Automata 34
3.2 Formal Power Series over Sequential-Parallel Posets 38

4 First Closure Properties of Regular Sequential-Parallel Series **47**
4.1 Closure under Sum and Scalar Products 47
4.2 Normalization 48
4.3 Closure under Parallel Product and Iteration 51
4.4 Closure under Sequential Product and Iteration 57
4.5 From Rationality to Regularity 67

5 Bounded Width and Bounded Depth **69**

6 The Fundamental Theorem **87**

7 The Different Concepts of Regularity **93**

8 The Hadamard Product for SP-Series **115**
8.1 The Hadamard Product for Doubled Semirings 117
8.2 The Hadamard Product for Distributive Bisemirings 133

9 Series and Languages **147**

10 Conclusion **153**

Bibliography **159**

1 Introduction

One of the most powerful extensions of classical formal language and automata theory is the consideration of weights, costs, or multiplicities. This line of research was initiated by Schützenberger [Sch61b] and Eilenberg [Eil74]. Most naturally, this generalization arises if we consider the number of successful runs of a finite automaton on a word and not only the existence of such a run. Most surprisingly, an enormous amount of different weight models all can be described by one algebraic structure: the semiring. Last but not least because of their importance in theoretical computer science the algebraic study of semirings intensified and was summarized, see *e.g.* the textbooks of Golan [Gol99] and Hebisch and Weinert [HW99]. Since the paper of Schützenberger [Sch61b], a tremendous amount of literature was published within the area of weighted automata over words. For an overview see [SS78, BR88, KS86, Kui97b]. Last but not least, the increasing application of weighted automata in language recognition (cf. [Moh97, MPR00]) and image compression (cf. [CK93, CK97, Haf99, Kat01]) as well as the work of the Max-Plus-community (cf. [GP97, CGQ99, Gun98, BCOQ92]) are responsible for the renaissance of weighted automata within the last years. Especially, weighted automata over other structures than words, like trees and Mazurkiewicz traces, have received a lot of attraction. It is the aim of this dissertation to investigate weighted automata over structures incorporating concurrency.

In order to model semantics of concurrent processes a variety of models have been proposed, *e.g.* the information systems of Scott [Sco82] or the event structures of Winskel [Win87]. Models closer to formal language theory are Mazurkiewicz traces [Maz87, DR95], the automata with concurrency relations of Droste [Dro92], or the asynchronous cellular automata introduced by Zielonka [Zie87, Zie89] for traces and extended to pomsets without auto-concurrency by Droste, Gastin, and Kuske [DGK00]. For an overview see also [NW95]. Another line of research was initiated by Grabowski [Gra81], Pratt [Pra86], and Gischer [Gis88]. They extended previous ideas by Kleene [Kle56] on sequential systems built by non-deterministic choice, sequential composition and iteration, and, in addition, proposed a parallel composition in order to model distributed systems. It turned out that sequential-parallel posets[1] are ideally suited to describe executions of such systems. However, sequential-parallel posets have one significant weakness: they can model a hierarchy of parallel processes only, but no passing of messages. On the other hand, they allow auto-concurrency. This, for instance, is a difference to Mazurkiewicz traces. Later on, Lodaya and Weil [LW00] proposed a finite-state device capable of accepting languages of sequential-parallel posets which was extended by Kuske [Kus03] to infinite sequential-parallel posets. These automata model parallelism by branching – hence the name "branching automata". Runs of branching automata are built of atomic runs by two compositions: a sequential one, as in the case of words, and a parallel one. Suppose we wanted to calculate the minimal duration of a run in a system constructed by sequential and parallel composition. The execution time of a sequential composition is the sum of the durations of the sub-runs. Whereas the execution time of a parallel composition is the maximum of the durations of the sub-runs, possibly increased by some time for the necessary forking and joining of the processes. This is natural because the system has to wait till the last sub-process is finished. Thus, the two operations for calculating the durations of sequential and parallel compositions are different. If the system is non-deterministic a given sequential-parallel poset can be executed in different ways and we should consider the minimal duration of all possible executions. Thus, our driving example of calculating minimal duration times requires three operations on the underlying weight structure: addition for sequential composition, maximum for the parallel one, and minimum to handle non-determinism.

In weighted automata over words, sequential composition is modeled by the multiplication of the underlying semiring, and non-determinism by the addition of the semiring. In order to accompany the situation described above, we introduce a second multiplication to deal with parallel composition. This multiplication should be commutative because parallel

[1] The term "poset" stands for "partially ordered set". Gischer [Gis88] called these posets "series-parallel pomsets" (partially ordered marked sets), and Lodaya and Weil [LW00] "series-parallel posets". To avoid any confusion with the notion "series", used here for formal power series, we will call them "sequential-parallel posets".

composition is so. This observation results in the notion of bisemirings, *i.e.* structures consisting of two semirings on a joint domain with the same additive operation. Weights of executions in our model of weighted branching automata are then evaluated in such bisemirings and the behavior of a weighted branching automaton is a function that associates with every sequential-parallel poset an element from the bisemiring.

It is the aim of this dissertation to characterize those functions that are associated with weighted branching automata, to examine closure properties of them, and to explore the connections to languages of sequential-parallel posets as considered by Lodaya and Weil [LW00, LW01]. A part of the results was already published in [KM03].

After introducing sequential-parallel posets and bisemirings in Chapter 2, we define the model of weighted branching automata in Chapter 3. A weighted branching automaton is a finite-state device with three different types of transitions. Sequential transitions are defined as in classical weighted automata: they transform a state into another one by executing an action from the alphabet, and they carry a weight from the bisemiring. On the other hand, fork and join transitions are responsible for branching. A fork transition forks from one state into several other states, whereas a join transition joins a number of states to one state. Certainly, fork and join transitions also carry weights from the underlying bisemiring. Furthermore, weighted branching automata have an initial and a final weight function which determine at which states and by which weights the system is allowed to be entered and left. In a weighted branching automaton, the parallel composition of n runs is realized as follows: in one state a fork transition branches into n states, from these n states the n runs are executed, and, finally, the n terminating states of these runs are joined by a join transition to one state. Hence, parallel runs are balanced in the sense that they start with a fork and end with a join transition. This is quite different to tree automata. Depending on the kind of runs which are allowed to be composed in parallel, two possible running modes of a weighted branching automaton are defined. The cascade branching mode is closer to the machine level. Here, the branching into several parallel sub-processes can be done in cascades. On the other hand, the maximally branching mode is closer to the algebraic structure of sequential-parallel posets. When branching, a process branches at once into all possible subprocesses. These two different notions of runs result in two kinds of behaviors: the cascade branching behavior, or C-behavior for short, and the maximally branching behavior, or M-behavior. The C-runs are the runs Lodaya and Weil [LW00] defined for their branching automata. They did not consider something like the M-mode. Nevertheless, the M-runs resemble the runs they define for branching automata over term algebras with an additional series operation [LW01].

We want to characterize the behaviors of weighted branching automata in the style of Kleene's theorem [Kle56] stating the equivalence of recognizable and rational sets of words. Weighted automata over words generate functions from the free monoid of all finite words into the semiring in consideration. To develop a concept of rationality for those functions it turned out to be convenient to consider these functions as formal power series in non-commuting variables. This allowed Schützenberger [Sch61b] to show that a formal power series is rational if and only if it is the behavior of a weighted word automaton. We follow these lines and introduce in Section 3.2 formal power series over sequential-parallel posets with values in a bisemiring which we call sequential-parallel series or sp-series. Moreover, we define rational operations; this results in the classes of rational and of sequential-rational sp-series. The rational operations comprise sum, scalar products, sequential product and iteration, and parallel product and iteration, whereas the sequential-rational operations exclude the parallel iteration. 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 [LW00] where the number of parallel actions is bounded for all elements of the language uniformly. Since in a parallel system the width corresponds to the number of independent processes, this restriction seems natural to us. Therefore, a lot of our results hold for weighted branching automata generating sp-series with support of bounded width. However, in [LW98] Lodaya and Weil gave another notion of rationality replacing parallel iteration by ξ-substitution and a restricted ξ-exponentiation borrowed from the generalized rational expressions of Thatcher and Wright [TW68]. In [LW01], they refined the restriction of the ξ-exponentiation again and could prove the coincidence of these generalized rational languages with those languages accepted by branching automata. However, this notion of rationality – closer to rationality of tree series – is beyond the scope of this dissertation.

Chapters 4, 5, and 6 are devoted to the proof of a theorem in the spirit of Kleene and Schützenberger. In Chapter 4, the closure of regular sp-series, *i.e.* those recognized by a weighted branching automaton, under rational operations is shown. Except for parallel product and iteration these closure properties hold true both for cascade and maximally branching mode. In addition, a result is given stating that every weighted branching automaton can be turned into a normalized one with the same behavior. Normalized automata can be entered in one state by weight 1 only, and dually for final states. All proofs given for normalization and closure under rational operations are constructive. The concluding result of Chapter 4 states that every rational sp-series is C-regular. To prove the converse of this statement we restrict ourselves to series of bounded width as indicated above. The property of bounded width is one of the series and not of the automaton. However, in order to show that regular sp-series of bounded width are sequential-rational we have to construct a sequential-rational expression from the given weighted branching automaton. For this, a hierarchy of the parallel sub-processes is necessary. This hierarchy can be described by a depth of a run, a notion due to Kuske [Kus03] developed in the context of sp-languages. If the depths of all runs of a weighted branching automaton are uniformly bounded then the automaton is said to be of bounded depth. It turns out that the notion of bounded depth reflects the bounded width of the behavior, *i.e.* every regular sp-series of bounded width can be recognized by a weighted branching automaton of bounded depth. This was already shown for sequential-parallel languages by Kuske [Kus03]. However, for sp-series over bisemirings which allow a non-trivial additive composition of zero the construction of Kuske does not succeed. We give a much more involved construction to prove the result for arbitrary bisemirings. Next, the equivalence of the concepts of bounded depth and fork acyclicity is shown. Fork acyclic automata were introduced by Lodaya and Weil [LW00]. Last but not least, we prove the decidability of empty support and of bounded width for regular sp-languages in case the bisemiring is positive, *i.e.* the bisemiring does not allow a non-trivial decomposition of zero, neither additive nor multiplicative. All this is subject of Chapter 5. Now, we can prove the coincidence of sequential-rational sp-series and C-behaviors of weighted branching automata of bounded depth. This is done in Chapter 6. Moreover, every sequential-rational sp-series is the C-behavior of some weighted branching automaton of bounded depth which is normalized and forks always into two branches only. By these results the Kleene-like characterization given by Lodaya and Weil for sp-languages [LW00] is generalized to sp-series over arbitrary bisemirings.

In Chapter 7 we discuss the two different notions of C- and M-regularity. It is easy to see that M- and C-regular sp-series do not coincide for unbounded width. Again, we restrict ourselves to sp-series of bounded width. Firstly, we give a construction turning a weighted branching automaton $\mathcal{A}$ of bounded depth into another one $\mathcal{A}'$ such that the C-behavior of $\mathcal{A}'$ is the same as the M-behavior of $\mathcal{A}$. Hence, in the case of bounded width every M-behavior, which is closer to the algebraic structure of sequential-parallel posets, can be realized as a C-behavior, closer to the machine level. The other way round, *i.e.* to simulate C-behaviors by M-behaviors, more difficulties arise. An example is given showing that this is not possible in general. Restrictions have to be imposed on the bisemiring. Two classes of bisemirings are considered: doubled semirings, where sequential and parallel product coincide, and distributive bisemirings, where the sequential product distributes over the parallel one. For both classes we show that every sequential-rational sp-series is M-regular, and, therefore, M-regular and C-regular sp-series of bounded width coincide for these bisemirings. *En passant*, we prove that over distributive bisemirings each weighted branching automaton of bounded depth allows an equivalent automaton with the same behavior where all fork and join transitions have weight 1 only.

The M-behavior plays a prominent role in Chapter 8. This chapter is concerned with the closure of regular sp-series under the sequential Hadamard product. The Hadamard product of two series is defined as a pointwise product and is the generalization of the intersection of two languages. Lodaya and Weil [LW00, Thm. 4.6] noted for regular sp-languages that the closure under intersection can be shown by the classical construction. This works when considering the product algebra of the recognizing finite sp-algebras, *i.e.* by turning to an algebraic characterization. However, not every regular sp-language is recognizable, as already noted by Lodaya and Weil [LW00]. But in order to recognize the Hadamard product by an automaton we would have to construct a product automaton simulating two automata simultaneously. As we will show this construction cannot be generalized straightforwardly in the C-running mode. This is because the branching structure of the automaton does not reflect the associativity of the parallel product. Therefore, we have to consider the M-running mode. For the product automaton recognizing the Hadamard product it is necessary to shift weights over fork and join transitions away to the different depth levels of the automaton. Therefore, it is obvious that sequential and parallel multiplication of the bisemiring cannot be absolutely independent of each other. We show the closure of M-regular sp-series under Hadamard product in the case of idempotent commutative doubled semirings. The commutativity is not surprising because it is already necessary in the case of words. However, idempotency seems to come as a surprise. As we will see, it is due to auto-concurrency. When we restrict ourselves to sp-posets which do not allow a direct auto-concurrency we can drop this restriction. For idempotent, commutative, distributive bisemirings M-regular sp-series are not closed under Hadamard product anymore. Nevertheless, we explore the behavior of the product automaton over the bisemiring $(\mathbb{N} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$ which is of this type. In this case, the behavior of the product automaton is less than or equal to the Hadamard product.

The last chapter is concerned with some basic connections between series and languages, especially with characteristic series and supports. A first result states that the support of a regular sp-series is regular if the underlying bisemiring is positive. Then we give an example of a regular sp-language whose characteristic series is not regular over the doubled semiring of the natural numbers. This differs completely from the situation of word series where the characteristic series of a regular language is always regular [BR88, Prop. 2.1]. But if we choose the bisemiring idempotent and impose another small restriction, characteristic series of regular sp-languages are regular. Last but not least, regular series over idempotent bisemirings stay regular when restricted to regular languages.

As we already noted above, a rich field of research concerning series over other structures than words was opened over the last years. One subject very close to ours are the series over Mazurkiewicz traces examined by Droste and Gastin [DG99, DG00]. Last but not least, traces are an important model for concurrency which is determined by a global independence relation on the alphabet. Droste and Gastin considered trace series with values in a semiring. The automata they used are weighted word automata with an additional I-diamond property which ensures that equivalent words, with respect to the independence relation, have the same weight. Naturally, this way the weight operations for sequential and parallel composition cannot be handled differently. In [DG99], Droste and Gastin proved a Schützenberger theorem for commutative semirings. Moreover, they also considered aperiodic formal power series over traces [DG00].

A tremendous amount of literature was published on tree series. Berstel and Reutenauer [BR82] showed that recognizable formal tree series over fields arise as components of unique solutions of certain systems of equations over polynomials. Later on, Bozapalidis *et al.* [BLB83, BA89, Boz91] gave a characterization of recognizable formal tree series over fields by matrix representation and their syntactic algebra following a line of research initiated by Reutenauer [Reu80, BR88] for word series. Later on, Bozapalidis [Boz94] gave a characterization of representable tree series by finitely generated stable semimodules for an arbitrary semiring $\mathbb{K}$. Kuich [Kui97a] proved a Kleene-like theorem for recognizable formal tree series by using fixed point theory on complete partial orders. Therefore, the underlying semiring has to be commutative, complete, naturally ordered, and continuous. Recently, a generalization of Kuich's result was achieved by Bloom and Ésik [BÉ03] who showed a Kleene-theorem for commutative Conway-semirings. Kuich [Kui00] also showed a Kleene-like theorem for series over Σ-algebras. On the other hand, Droste, Vogler, and Pech [DPV04] proved by elementary automata constructions a Kleene-type result for tree series over commutative semirings. Pech [Pec03a, Pec03b] proposed for tree series over commutative semirings a new technique of proving the Kleene-type theorem on another higher semantic level and by transforming this result later on back to the level of formal tree series by a natural abstraction map. The higher semantic level is modeled by weighted tree languages, these are multisets of trees whose nodes are equipped with weights from the semiring. As already indicated, runs in weighted tree automata are different from those of branching automata because they are built by top-catenation of runs by a transition. If *e.g.* the function symbol is binary, the corresponding operation takes three arguments: two runs and the transition. The weights of these three arguments are multiplied to get the weight of the extended run. Since sequential-parallel posets can be seen as terms modulo associativity and commutativity of the parallel product we could try to transfer the top-catenation composition of runs to automata on sp-posets. But this would yield a model where parallel and sequential composition are dealt with in a uniform manner which we consider inappropriate. However, in [Kui97a] Kuich introduced distributive multioperator monoids or DM-monoids for short. A DM-monoid is a commutative monoid with additional operations which distribute over the monoid operation. Moreover, the unit of the monoid is annihilating for the other operations. Kuich noticed that the distributive operations of the DM-monoid can be lifted to series over trees with values in a DM-monoid. In [Kui99], he considered automata on DM-monoids and showed that behaviors of those automata and the least solutions of linear systems of equations are equivalent mechanisms if the underlying DM-monoid is continuous. Bisemirings can be seen as DM-monoids even if not necessarily as continuous DM-monoids. However, the automata of Kuich [Kui99] resemble top-down tree automata. Even if we would consider sp-posets as terms and bisemirings as DM-monoids it is by far not understood how our branching automata compare to those automata of Kuich. It may be interesting to explore this connection furthermore.

By combining the phenomena of weights and concurrency by the model of weighted branching automata we hope both to enrich the area of concurrency and to popularize the idea of composing weights differently with respect to the underlying composition of actions.

2 Sequential-Parallel Posets and Bisemirings

2.1 Sequential-Parallel Posets

Words, *i.e.* elements of the free monoid $\Sigma^\star$ over a finite alphabet Σ, are used to model the executions of a sequential system. Words can also be seen as linearly ordered sets. In order to cover concurrency one can turn to partial orders instead of linear ones.

A *partially ordered set* $(V, \leq)$, or *poset* for short, is a set V equipped with an ordering relation $\leq$, *i.e.* a reflexive, antisymmetric, and transitive binary relation on V. The empty poset $(\emptyset, \emptyset)$ is denoted by ε. A set $A \subseteq V$ is an *anti-chain* of $(V, \leq)$ if the elements of A are mutually incomparable. From now on, any poset is assumed to be finite. Then the *width* of a poset $t = (V, \leq)$ is the maximal cardinality of an anti-chain in t, *i.e.* $\mathrm{wd}(t) = \max\{|A| \mid A \subseteq V \text{anti-chain}\}$.

Now let Σ be a finite alphabet. A Σ-*labeled poset* $(V, \leq, \tau)$ is a finite poset $(V, \leq)$ equipped with a labeling function $\tau : V \longrightarrow \Sigma$. Again, $\varepsilon = (\emptyset, \emptyset, \emptyset)$ is the empty Σ-labeled poset. Now, we define the sequential and the parallel product of two Σ-labeled posets $t_1 = (V_1, \leq_1, \tau_1)$ and $t_2 = (V_2, \leq_2, \tau_2)$ (cf. Figure 2.1). We assume $V_1 \cap V_2 = \emptyset$ (otherwise take disjoint copies of V_1 and V_2, respectively). The *sequential product* $t_1 \cdot t_2$ of t_1 and t_2 is the Σ-labeled poset

$$(V_1 \cup V_2, \leq_1 \cup (V_1 \times V_2) \cup \leq_2, \tau_1 \cup \tau_2).$$

Hence, in $t_1 \cdot t_2$ every element of V_1 is less than every element of V_2. Graphically, t_2 is put on top of t_1. The *parallel product* $t_1 \parallel t_2$ is defined as

$$(V_1 \cup V_2, \leq_1 \cup \leq_2, \tau_1 \cup \tau_2),$$

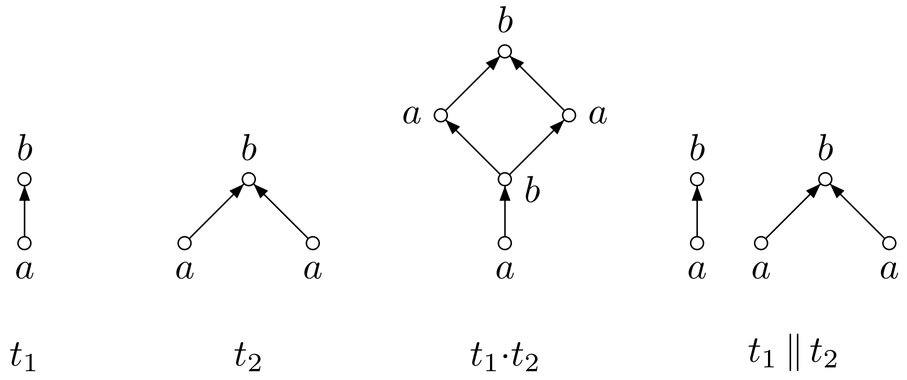


Figure 2.1: Sequential and parallel product.

i.e. the two partial orders are put side by side. Thus, every element of V_1 is incomparable to every element of V_2. Obviously, both the sequential and the parallel product are associative. Moreover, the parallel product is also commutative. Note that there is no connection between both products. In the sequel, we will not further mention the condition $V_1 \cap V_2 = \emptyset$ when applying one of the two products. In fact we consider isomorphism classes of Σ-labeled posets, also called pomsets in the literature [Gis88].

$\mathrm{SP}(\Sigma)$ 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 called *sequential-parallel posets*[1] over Σ or *sp-posets* for short. Thus, any sp-poset can be obtained from the singletons by applying the sequential and parallel product a finite number of times. Hence, any sp-poset is finite. We write SP instead of $\mathrm{SP}(\Sigma)$ if Σ is clear from the context. Note that SP does not comprise the empty poset ε. We put $\mathrm{SP}^1 = \mathrm{SP} \cup \{\varepsilon\}$. But from now on, we usually consider non-empty posets only.

The sp-posets may be characterized by a forbidden sub-structure. Consider the poset $(N, \leq_N)$ shown in Figure 2.2. A poset $(V, \leq)$ is *N-free* if $(N, \leq_N)$ cannot be embedded in $(V, \leq)$, *i.e.* there is no order-isomorphism from $(N, \leq_N)$ to a subposet of $(V, \leq)$.

Lemma 2.1 ([Gra81, Gis88]). *A Σ-labeled poset $(V, \leq, \tau)$ is an sp-poset if and only if $(V, \leq)$ is N-free.*

[1] Called "series-parallel posets" in [LW00].

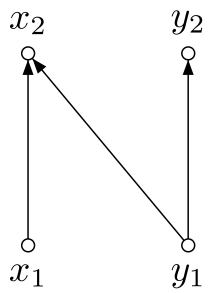


Figure 2.2: The poset $(N, \leq_N)$.

Note that this characterization is independent of the concrete labeling τ of the poset. The poset $(N, \leq_N)$ models some kind of message passing. Once more, consider Figure 2.2. Indeed, $x_1 \leq x_2$ and $y_1 \leq y_2$ may be seen as two parallel processes between which a message is passed from y_1 to x_2. Thus, $y_1 \leq x_2$. In the introduction we already noted that this type of concurrency is not covered by sp-posets.

$(\mathrm{SP}(\Sigma), \cdot, \parallel)$ is an algebra in the sense of universal algebra (see [Wec92] for an overview). Naturally, it may be generalized to the notion of an sp-algebra as already observed by Lodaya and Weil [LW00]. An *sp-algebra* $(S, \cdot, \parallel)$ is a set S with two binary associative operations, called sequential and parallel product, where $\parallel$ is also commutative. Clearly, $(\mathrm{SP}(\Sigma), \cdot, \parallel)$ is an sp-algebra. It is isomorphic to the free sp-algebra over the set Σ in the variety of sp-algebras [LW00].

An element $e \in S$ is an *identity* (or a *neutral element*) if it acts as a unit both for the sequential and the parallel product, *i.e.* $s \cdot e = e \cdot s = s \parallel e = s$ for all $s \in S$. Existence assumed it is unique and usually denoted by 1. If S has an identity we call S an *sp-algebra with identity*. For instance, $(\mathrm{SP}^1(\Sigma), \cdot, \parallel)$ is the sp-algebra with identity of all (even empty) sp-posets labeled by elements of Σ with identity being the empty poset ε. According to [Gra81] and [BÉ96], sp-algebras with identity are also called *double monoids* or *bimonoids*, respectively.

A *homomorphism* φ from an sp-algebra S to an sp-algebra T is a mapping $\varphi : S \to T$ preserving both products, *i.e.* $\varphi(x \cdot y) = \varphi(x) \cdot \varphi(y)$ and $\varphi(x \parallel y) = \varphi(x) \parallel \varphi(y)$ for all $x, y \in S$. A *congruence* on an sp-algebra S is an equivalence relation $\sim$ on S compatible with both products, *i.e.* $x \sim y$ implies $x \cdot z \sim y \cdot z$, $z \cdot x \sim z \cdot y$, and $x \parallel z \sim y \parallel z$ for all $x, y, z \in S$. Obviously, the quotient $S/\!\sim$ is equipped with the structure of an sp-algebra by defining the products of the congruence classes representative-like.

Since $\mathrm{SP}(\Sigma)$ is freely generated, we can define two decompositions of an sp-poset. The sp-poset t is *sequential* if there are no $u, v \in \mathrm{SP}$ such that $t = u \parallel v$, *i.e.* t cannot be written as a parallel product. Dually, t is *parallel* if it cannot be written as a sequential product $t = u \cdot v$ for some $u, v \in \mathrm{SP}$. The only sp-posets that are both sequential and parallel are the singleton posets which are identified with the elements of Σ. Every $t \in \mathrm{SP}$ not being a singleton is either sequential or parallel. Since $\mathrm{SP}(\Sigma)$ is free over Σ, the semigroup $(\mathrm{SP}(\Sigma), \cdot)$ is freely generated by the parallel sp-posets, and the commutative semigroup $(\mathrm{SP}(\Sigma), \parallel)$ is freely generated by the sequential sp-posets. Note that in both cases the set of generators is infinite. More precisely, every $t \in \mathrm{SP}(\Sigma)$ admits a factorization $t = t_1 \cdot \ldots \cdot t_m$ where $m \geq 1$ and each $t_i \in \mathrm{SP}$ $(i = 1, \ldots, m)$ is parallel. This factorization is unique up to associativity of the sequential product and is called the *sequential decomposition* of t. Similarly, each $t \in \mathrm{SP}(\Sigma)$ admits a factorization $t = t_1 \parallel \ldots \parallel t_n$ with $n \geq 1$ and $t_i \in \mathrm{SP}$ $(i = 1, \ldots, n)$ being sequential. Again, this factorization is unique up to associativity and commutativity of $\parallel$ and is referred to as the *parallel decomposition* of t. The number of factors in the decomposition is the *length* of the decomposition.

Let $t = (V, \leq, \tau)$ be an sp-poset. We call $|t| = |V|$ the *size* of t. Now, assume $t \neq a \in \Sigma$. Then either $t = t_1 \cdot \ldots \cdot t_m$ with $m \geq 2$ or $t = t_1 \parallel \ldots \parallel t_n$ with $n \geq 2$. Clearly, $|t_i| < |t|$ for every factor t_i either in the sequential or in the parallel decomposition. Now each factor of the decomposition not being a letter may be factorized again. Since the size of the factors decreases strictly this process of decomposing cannot go on forever: after a finite number of steps a factorization having only letters as factors is reached. Consequently, one proves a property $\mathcal{P}$ for all sp-posets $t \in \mathrm{SP}(\Sigma)$ using the following induction on the structure of t:

- one shows that $\mathcal{P}$ holds true for all letters of Σ;
- one shows that if $\mathcal{P}$ holds for all the factors of the sequential or parallel decomposition of the sequential or parallel sp-poset t, respectively, then it also holds true for t.

Hence, for instance the width of t can be computed by an induction process:

- if $t = a \in \Sigma$ then $\mathrm{wd}(t) = 1$,
- if $t = t_1 \cdot \ldots \cdot t_m$ then $\mathrm{wd}(t) = \max\{\mathrm{wd}(t_1), \ldots, \mathrm{wd}(t_m)\}$,
- if $t = t_1 \parallel \ldots \parallel t_n$ then $\mathrm{wd}(t) = \sum_{i=1}^{n} \mathrm{wd}(t_i)$.

Remark 2.2. A *semiring* $(K, \oplus, \circ, 0, 1)$ is a set K equipped with two binary operations $\oplus$ and $\circ$, called addition and multiplication, such that $(K, \oplus, 0)$ is a commutative monoid, $(K, \circ, 1)$ is a monoid, $\circ$ distributes over $\oplus$, and 0 is absorbing for the multiplication, *i.e.* $0 \circ k = k \circ 0 = 0$ for all $k \in K$. If the multiplication is commutative we speak of a *commutative semiring*. Thus $(K, \circ, \oplus)$ is an sp-algebra for every semiring $(K, \oplus, \circ, 0, 1)$, and $(K, \oplus, \circ)$ is an sp-algebra for every commutative semiring. Hence, one can define the width of sp-posets also by a homomorphism from $(\mathrm{SP}, \cdot, \parallel)$ to the polar semiring $(\mathbb{N}, \max, +)$ by mapping each letter to 1. If we consider an sp-algebra homomorphism from $(\mathrm{SP}, \cdot, \parallel)$ into $(\mathbb{N}, +, \max)$ with $a \mapsto 1$ for all $a \in \Sigma$ one gets the *height* of an sp-poset t, *i.e.* the cardinality of the longest chain in t.

A subset $L \subseteq \mathrm{SP}$ is called a *language* of sp-posets, or an *sp-language* for short. An sp-language L is of *bounded width* (or *width-bounded*) if there is some $n \in \mathbb{N}$ such that $\mathrm{wd}(t) \leq n$ for all $t \in L$.

Example 2.3. The set of all sp-posets not using the parallel product at all is an sp-language that can be identified with the free semigroup Σ^+ over Σ. Obviously, Σ^+ is of bounded width with a uniform bound of 1. Dually, the set of all sp-posets not using the sequential product may be identified with the free commutative semigroup $\Sigma^{\boxplus}$. This sp-language is not of bounded width. Now let $\Sigma = \{a, b\}$. We define two sp-languages L_1 and L_2 as being the least sets such that

- $a \in L_1$ and if $t \in L_1$ then $a \cdot (a \parallel t) \cdot b \in L_1$,
- $a \in L_2$ and if $t \in L_2$ then $a \cdot t \cdot (b \parallel b) \in L_2$.

Whereas L_1 is not of bounded width, L_2 is width-bounded.

Remark 2.4. In terms of computer science one can think of $\mathrm{wd}(t)$ as the minimal number of processors necessary to realize t. Hence, an sp-language of bounded width can be executed by a finite number of processors. On the contrary, languages of unbounded width cannot be realized with a finite number of processors only.

2.2 Bisemirings

The natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$ together with addition and multiplication are one of the oldest algebraic structures we are used to. This structure is a semiring, informally a ring lacking the inverse elements with respect to addition (cf. Remark 2.2). But semiring theory was not a very prominent subject of mathematics for a long time. Only in the last decades an algebraic theory of semirings was developed, for an overview and introduction see [HW99, Gol99]. Last but not least, theoretical computer science put semirings on the agenda again. As it was noticed by Schützenberger [Sch61b] and Eilenberg [Eil74], semirings are ideally suited for introducing quite a number of models for weights (or costs, or multiplicities) in finite automata, and to describe the behaviors of such machines. In weighted finite automata, weights are multiplied by semiring multiplication along a run and then the weights of all runs with the same label are summed up using semiring addition.

Because we want to model concurrency we will introduce systems with a sequential and a parallel composition later on. If weights should be considered in those systems multiplication of weights may depend on the kind of composition used. Therefore, we need a structure equipped with two multiplications instead of only one as in semirings. This leads to the following definition.

Definition 2.5. 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:

1. $(K, \oplus, 0)$ is a commutative monoid,
2. $(K, \circ, 1)$ is a monoid,
3. $(K, \diamond)$ is a commutative semigroup,

4. both $\circ$ and $\diamond$ distribute over $\oplus$, *i.e.* for all $k, l, m \in K$ and $* \in \{\circ, \diamond\}$

$$\begin{aligned} k * (l \oplus m) &= (k * l) \oplus (k * m) \text{ and} \\ (l \oplus m) * k &= (l * k) \oplus (m * k), \end{aligned}$$

5. 0 is absorbing for $\circ$ and $\diamond$, *i.e.* $k \circ 0 = 0 \circ k = k \diamond 0 = 0$ for all $k \in K$.

Thus, a bisemiring is built from two semiring structures having the same domain K and the same additive structure. More precisely, $(K, \oplus, \circ, 0, 1)$ is a semiring, and $(K, \oplus, \diamond, 0)$ is almost a semiring, only the parallel multiplication $\diamond$ does not have to admit a unit. Note that the parallel multiplication of a bisemiring is always commutative, namely to model the composition of weights of two concurrent processes.[2]

A bisemiring is *commutative* if the sequential multiplication $\circ$ is commutative. It is *idempotent* if the addition is idempotent, *i.e.* $k \oplus k = k$ for all $k \in K$. In general there is no connection between the two products of a bisemiring. But if the sequential multiplication distributes over the parallel one, *i.e.* $k \circ (l \diamond m) = (k \circ l) \diamond (k \circ m)$ and $(l \diamond m) \circ k = (l \circ k) \diamond (m \circ k)$ for all $k, l, m \in K$, we call the bisemiring *distributive*.

Let $\mathbb{K} = (K, \oplus, \circ, 0, 1)$ be a semiring and define $k \diamond l = 0$ for all $k, l \in K$. Then $(K, \oplus, \circ, \diamond, 0, 1)$ is a bisemiring, even a distributive one. If the semiring $\mathbb{K}$ is commutative, the structure $(K, \oplus, \circ, \circ, 0, 1)$ also is a bisemiring. Here, sequential and parallel multiplication coincide. Especially, the *Boolean bisemiring* $\mathbb{B} = (\{0, 1\}, \vee, \wedge, \wedge, 0, 1)$ with $1 \vee 1 = 1$ is of this form.

A *bisemiring homomorphism* is a mapping $h : \mathbb{K} \to \mathbb{K}'$ between two bisemirings $\mathbb{K}$ and $\mathbb{K}'$ such that $h(k * l) = h(k) * h(l)$ for $* \in \{\oplus, \circ, \diamond\}$, $h(0) = 0$, and $h(1) = 1$.

Below we collect some examples of bisemirings.

Example 2.6. The structure $\mathbb{T} = (\mathbb{N} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$ is the bisemiring that we referred to in the introduction. It is called the *tropical bisemiring*[3]. Here, 0 is the unit for the sequential multiplication $+$ and

[2] The notion "bisemiring" was first used in [KM03]. Recently, Sen, Ghosh, and Ghosh [SGG04] used it in a different meaning. There, an algebraic structure $(S, +, \cdot, \times)$ is a bisemiring if $(S, +, \cdot)$ and $(S, \cdot, \times)$ are semirings, *i.e.* the operation $\cdot$ plays one time the role of multiplication and the other time the role of addition.

[3] This notion goes back to the *tropical semiring* $(\mathbb{N} \cup \{+\infty\}, \min, +, +\infty, 0)$. According to I. Simon the name was suggested by Ch. Choffrut.

$+\infty$ is the absorbing zero. The parallel multiplication max admits also a unit, namely 0. $\mathbb{T}$ is idempotent, commutative, and distributive.

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 $t = t_1 \cdot \ldots \cdot t_m \in \mathrm{SP}$ we put $\mathrm{time}(t) = \mathrm{time}(t_1) + \ldots + \mathrm{time}(t_m)$, and for $t = t_1 \parallel \ldots \parallel t_n \in \mathrm{SP}$ we put $\mathrm{time}(t) = \max\{\mathrm{time}(t_1), \ldots, \mathrm{time}(t_n)\}$. Hence, $\mathrm{time} : (\mathrm{SP}, \cdot, \parallel) \to (\mathbb{N} \cup \{+\infty\}, +, \max)$ is an sp-algebra homomorphism and can be interpreted as the duration time of an sp-poset t. In Example 3.4, we will present an automaton that computes the minimal execution time of an sp-poset using the semiring $\mathbb{T}$.

If we extend the domain of the bisemiring $\mathbb{T}$ to $\mathbb{Z}$, $\mathbb{Q}$, or $\mathbb{R}$ (united with $+\infty$) we obtain three other bisemirings $\mathbb{T}_Z$, $\mathbb{T}_Q$, and $\mathbb{T}_R$ which we refer to as the $\mathbb{Z}$-, $\mathbb{Q}$-, and $\mathbb{R}$-*tropical bisemiring*, respectively. Another descendant is the bisemiring $\mathbb{T}_{R^{\geq 0}} = (\mathbb{R}^{\geq 0} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$[4]. One may also extend $\mathbb{T}$ to $(\mathbb{N} \cup \{-\infty, +\infty\}, \min, +, \max, +\infty, 0)$ where we have to put $(-\infty) + (+\infty) = +\infty$ (recall that $+\infty$ is absorbing with respect to $+$). Now, the parallel multiplication max allows the unit $-\infty$.

Example 2.7. We define a binary operation nax for $k, l \in \mathbb{N} \cup \{-\infty\}$ as follows:

$$\mathrm{nax}(k, l) = \begin{cases} \max(k, l) & \text{if } k, l \neq -\infty, \\ -\infty & \text{otherwise.} \end{cases}$$

The bisemiring $\mathbb{P} = (\mathbb{N} \cup \{-\infty\}, \max, +, \mathrm{nax}, -\infty, 0)$ is called the *polar bisemiring*. It is idempotent, commutative, and distributive. Also $\mathbb{P}$ may be used to compute the duration time of an sp-poset t. But as we will see in Example 3.4, this time the maximal execution time of t in an automaton would be computed. As in Example 2.6 one may also consider the $\mathbb{Z}$-, $\mathbb{Q}$-, and $\mathbb{R}$-polar bisemirings.

Example 2.8. $\mathbb{F} = (\mathbb{R} \cup \{-\infty, +\infty\}, \max, \min, +, -\infty, +\infty)$ is called the *flow rate* or *capacity bisemiring*. Here $(-\infty) + (+\infty) = -\infty$. $\mathbb{F}$ is idempotent, commutative, but not distributive.

Let $a \in \Sigma$ be interpreted as a channel or pipe with some capacity $\mathrm{cap}(a)$. Let $\mathrm{cap}(a) = -\infty$ if a is blocked and there is no throughput at all. Then any $t \in \mathrm{SP}$ may be seen as a sequential-parallel channel system. For

[4] Here $\mathbb{R}^{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\}$.

$t = t_1 \cdot \ldots \cdot t_m \in \mathrm{SP}$ we define $\mathrm{cap}(t) = \min\{\mathrm{cap}(t_1), \ldots, \mathrm{cap}(t_m)\}$, *i.e.* the capacity of a sequence of channels is determined by a sub-channel of minimal capacity. For any $t = t_1 \parallel \ldots \parallel t_n \in \mathrm{SP}$ we put $\mathrm{cap}(t) = \mathrm{cap}(t_1) + \cdots + \mathrm{cap}(t_n)$, *i.e.* the flow rates of several channels used in parallel are summed up. Thus, $\mathrm{cap} : (\mathrm{SP}, \cdot, \parallel) \to (\mathbb{R} \cup \{-\infty, +\infty\}, \min, +)$ is an sp-algebra homomorphism and $\mathrm{cap}(t)$ determines the capacity of channel system t.

Example 2.9. Very similar to the last example is the *environmental bisemiring* $\mathbb{E} = (\mathbb{R}^{\geq 0} \cup \{+\infty\}, \min, \max, +, +\infty, 0)$. Suppose $a \in \Sigma$ stands for a production step in a fabrication process and $\mathrm{poll}(a)$ is the pollution (or the noise etc.) caused by a. If a cannot be performed we put $\mathrm{poll}(a) = +\infty$. For $t = t_1 \cdot \ldots \cdot t_m \in \mathrm{SP}$ we are interested in the maximal threshold of the pollution caused by this sequence of production steps. Thus, $\mathrm{poll}(t) = \max\{\mathrm{poll}(t_1), \ldots, \mathrm{poll}(t_m)\}$. For $t = t_1 \parallel \ldots \parallel t_n \in \mathrm{SP}$ the reached threshold is the sum of the single thresholds. Hence, $\mathrm{poll}(t) = \mathrm{poll}(t_1) + \cdots + \mathrm{poll}(t_n)$. Thus, $\mathrm{poll} : (\mathrm{SP}, \cdot, \parallel) \to (\mathbb{R}^{\geq 0} \cup \{+\infty\}, \max, +)$ yields the maximal threshold reached by the fabrication process t. If in a finite-state system several executions of t are possible, then we would gain the minimal pollution threshold that may be caused by t.

For the next two examples we do not have such an intuitive interpretation as for the ones given above.

Example 2.10. $(\mathbb{R}^{>0} \cup \{+\infty\}, \min, \cdot, +, +\infty, 1)$ where $\cdot$ denotes the usual multiplication is an idempotent, commutative, and distributive bisemiring that does not allow a unit for the parallel multiplication.

Example 2.11. Let M be a set and $\mathfrak{P}(M)$ the power set of M. Then $(\mathfrak{P}(M), \cup, \cap, \diamond, \emptyset, M)$ is a bisemiring if we define for $A, B \subseteq M$:

$$A \diamond B = \begin{cases} A \cup B & \text{if } A, B \neq \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$$

Similarly, $(\mathfrak{P}(M), \cap, \cup, *, M, \emptyset)$ with

$$A * B = \begin{cases} A \cap B & \text{if } A, B \neq M, \\ M & \text{otherwise} \end{cases}$$

is a bisemiring.

It is not surprising that the sequential and parallel product of sp-posets lifted to sp-languages give again rise to a bisemiring.

Example 2.12. Let Σ be a finite alphabet and $\mathrm{SP}^1 = \mathrm{SP} \cup \{\varepsilon\}$. Then the set of *non-proper sp-languages* $(\mathfrak{P}(\mathrm{SP}^1), \cup, \cdot, \|, \emptyset, \{\varepsilon\})$ constitutes a bisemiring. Here the multiplications $\cdot$ and $\|$ are defined elementwise, *i.e.* $L_1 \cdot L_2 = \{t_1 \cdot t_2 \mid t_1 \in L_1, t_2 \in L_2\}$ and similar for $L_1 \| L_2$. Note that this bisemiring is neither commutative nor distributive, but idempotent. The parallel multiplication $\|$ admits the same unit $\{\varepsilon\}$ as the sequential one.

The next example gives a quite general construction of bisemirings.

Example 2.13. Let $(L, \vee, \wedge)$ be a lattice and $(L, \cdot, 1)$ a monoid such that

$$a \cdot (b \vee c) = (a \cdot b) \vee (a \cdot c)$$
$$(a \vee b) \cdot c = (a \cdot c) \vee (b \cdot c)$$

for all $a, b, c \in L$. Then L is called a *lattice-ordered monoid* or *l-monoid* for short (cf. [Bir73, Chap. XIV,§ 4]). A *zero* of an l-monoid L is an element $0 \in L$ such that $0 \leq a$ (where $\leq$ is the order of the lattice) and $0 \cdot a = a \cdot 0 = 0$ for all $a \in L$ (cf. [Bir73, Chap. XIV,§ 1]). Now, assume L to be a distributive lattice. Then $(L, \vee, \cdot, \wedge, 0, 1)$ is an idempotent bisemiring.

We give two instances of such l-monoid bisemirings. Let $(M, \cdot, 1)$ be an arbitrary monoid. Then $(\mathfrak{P}(M), \cup, \cdot, \cap, \emptyset, \{1\})$ is a bisemiring where the sequential product is the monoid operation lifted to subsets of M.

Let $(R, +, \cdot, 0, 1)$ be a ring and U_R the collection of all additive subgroups of R. For $X, Y \in U_R$ we define $X \vee Y = \langle X \cup Y \rangle_+$ as the least additive subgroup of R containing $X \cup Y$, and

$$X \cdot Y = \left\{ \sum_{i=1}^{n} x_i y_i \;\middle|\; x_i \in X, y_i \in Y, n \in \mathbb{N}^{>0} \right\}.$$

Obviously, $X \vee Y$, $X \cdot Y$, and $X \cap Y$ are additive subgroups of R. Then $(U_R, \vee, \cap)$ is a distribute lattice with $\{0\}$ as zero. Moreover, the additive subgroup $\langle 1 \rangle_+$ generated by 1 is neutral for subgroup multiplication, and subgroup multiplication distributes over $\vee$. We omit the technical details of the proof. Hence, $(U_R, \vee, \cdot, \cap, \{0\}, \langle 1 \rangle_+)$ is a bisemiring.

The following example of matrices over a distributive lattice is due to [SGG04].

Example 2.14. Let $(L, \vee, \wedge)$ be a distributive lattice with least element 0 and greatest element 1, and let $M_n(L)$ be the set of all $n \times n$-matrices over L. Define for any $A = (a_{ij}), B = (b_{ij}) \in M_n(L)$

$$
\begin{aligned}
A \vee B &= (a_{ij} \vee b_{ij}) \ , \\
A \wedge B &= (a_{ij} \wedge b_{ij}) \ , \\
A \cdot B &= \left(\bigvee_{k=1}^{n} (a_{ik} \wedge b_{kj}) \right) .
\end{aligned}
$$

Furthermore, let $\mathbf{0} = (0_{ij})$ and $\mathbf{1} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix}$. Then $(M_n(L), \vee, \cdot, \wedge, \mathbf{0}, \mathbf{1})$ is an idempotent bisemiring.

We may generalize this example even to matrices over commutative semirings. Let $(R, +, \cdot, 0, 1)$ be a commutative semiring, *i.e.* a semiring with a commutative multiplication. Let $\oplus$ and $\odot$ be the pointwise addition and multiplication of matrices, and let $\circ$ be the usual matrix multiplication. With matrices $\mathbf{0}$ and $\mathbf{1}$ defined as above, the structure $(M_n(R), \oplus, \circ, \odot, \mathbf{0}, \mathbf{1})$ is a bisemiring.

Another quite general example of a bisemiring is that of binary relations.

Example 2.15. Let M be a set and R_M the set of binary relations on M. Let $\circ$ denote the usual relational product, *i.e.* for $A, B \in R_M$

$$
A \circ B = \{(a, b) \in M^2 \mid \exists\, c \in M : (a, c) \in A \text{ and } (c, b) \in B\} \ .
$$

By Δ we denote the diagonal relation: $\Delta = \{(a, a) \mid a \in M\}$. Then $(R_M, \cup, \circ, \cap, \emptyset, \Delta)$ is an idempotent bisemiring. Note that for M a finite set with n elements and L the Boolean lattice with two elements, the bisemiring $(M_n(L), \vee, \cdot, \wedge, \mathbf{0}, \mathbf{1})$ from Example 2.14 is isomorphic to $(R_M, \cup, \circ, \cap, \emptyset, \Delta)$.

We give an interpretation of the bisemiring of binary relations. Let Σ be a finite set of atomic situations or conditions, and M an arbitrary set of ports or agents. Then a binary relation $R \subseteq M \times M$ states which ports are connected such that, *e.g.*, messages or signals could be passed. Next, a condition $a \in \Sigma$ allows certain connections $\mathrm{link}(a) \subseteq M \times M$ between ports. Now, some conditions may emerge sequentially and others at the same time. Hence, any $t \in \mathrm{SP}$ can be understood as a complex condition or situation. For $t = t_1 \cdot \ldots \cdot t_m$ we put $\mathrm{link}(t) = \mathrm{link}(t_1) \circ \cdots \circ \mathrm{link}(t_m)$. Hence, $\mathrm{link}(t)$ gives us the connected ports after a sequence of conditions occurred. On the other hand, if $t = t_1 \parallel \ldots \parallel t_n$ then $\mathrm{link}(t) = \mathrm{link}(t_1) \cap \cdots \cap \mathrm{link}(t_n)$, *i.e.* only those port connections are still enabled that satisfy all conditions $t_1, \ldots, t_n$ simultaneously. Thus, $\mathrm{link} : (\mathrm{SP}, \cdot, \parallel) \to (R_M, \circ, \cap)$ is an sp-algebra homomorphism determining the set of connected ports after the occurrence of a complex condition t.

The last two examples will present a non-idempotent bisemiring and a finite bisemiring.

Example 2.16. Let $m, n \in \mathbb{N}$ with $m \neq 0$. We define two binary operations $\cdot_m$ and $\cdot_n$ on $\mathbb{Q}$ for all $x, y \in \mathbb{Q}$ by

$$\begin{aligned} x \cdot_m y &= m \cdot x \cdot y, \text{ and} \\ x \cdot_n y &= n \cdot x \cdot y \end{aligned}$$

where $\cdot$ denotes the usual multiplication. Both operations are associative and commutative. Moreover, they distribute over the usual addition:

$$\begin{aligned} x \cdot_m (y + z) &= m \cdot x \cdot (y + z) \\ &= m \cdot x \cdot y + m \cdot x \cdot z \\ &= (x \cdot_m y) + (x \cdot_m z). \end{aligned}$$

Thus, $(\mathbb{Q}, +, \cdot_m, \cdot_n, 0, \frac{1}{m})$ is a bisemiring. It is commutative, but neither idempotent nor distributive.

Example 2.17. Let $n \in \mathbb{N}$ with $n \geq 1$ and put $[n] = \{0, \ldots, n\}$. The structure $([n], \min, \dagger, \max, n, 0)$ with

$$x \dagger y = \begin{cases} x + y & \text{if } a + b \leq n, \\ n & \text{otherwise} \end{cases}$$

is an idempotent, commutative, distributive, and finite bisemiring.

We conclude this section by stating two constructions by which new bisemirings can be obtained from given ones. Since bisemirings are defined by universally quantified equations they constitute a variety in the sense of universal algebra (cf. [Wec92]). By Birkhoff's characterization, the class of bisemirings is closed under subalgebras, homomorphic images, and direct products. These closure properties imply the following two lemmata.

Lemma 2.18. *Let I be an index set and $\Big((K_i, \oplus_i, \circ_i, \Diamond_i, 0_i, 1_i)\Big)_{i \in I}$ a family of, not necessarily different, bisemirings, and let $K = \prod_{i \in I} K_i$ be the product set of the set family $(K_i)_{i \in I}$.*
The structure $\mathbb{K} = (K, \oplus, \circ, \Diamond, 0, 1)$ is defined componentwise:

- *for all $k = (k_i)_{i \in I}, l = (l_i)_{i \in I} \in K$ and $* \in \{\oplus, \circ, \Diamond\}$:*

$$a * b = (a_i *_i b_i)_{i \in I},$$

- $0 = (0_i)_{i \in I}$, $1 = (1_i)_{i \in I}$.

Then $\mathbb{K}$ is a bisemiring called the direct product of the bisemirings $\mathbb{K}_i$ *$(i \in I)$.*

Let $\mathbb{K}$ be a bisemiring. An equivalence relation $\sim$ on $\mathbb{K}$ which is compatible with the three operations of $\mathbb{K}$ is a *congruence relation* of $\mathbb{K}$. More precisely, any congruence $\sim$ satisfies for all $k, l, m \in K$:

- $k \sim l$ implies $(k \oplus m) \sim (l \oplus m)$,
- $k \sim l$ implies $(k \circ m) \sim (l \circ m)$ and $(m \circ k) \sim (m \circ l)$, and
- $k \sim l$ implies $(k \Diamond m) \sim (l \Diamond m)$.

We denote the congruence class of some $k \in K$ by $[k]_\sim$. Let $K/\!\sim$ be the set of all congruence classes and define $[k]_\sim *_\sim [l]_\sim = [k * l]_\sim$ for all $k, l \in K$ and $* \in \{\oplus, \circ, \Diamond\}$.

Lemma 2.19. *Let $\sim$ be a congruence on the bisemiring $\mathbb{K}$. Then $\mathbb{K}/\!\sim$ $= (K/\!\sim, \oplus_\sim, \circ_\sim, \Diamond_\sim, [0]_\sim, [1]_\sim)$ is a bisemiring, and is called the* quotient bisemiring *of $\mathbb{K}$ with respect to $\sim$.*

Example 2.20. Consider the tropical bisemiring $\mathbb{T}$ from Example 2.6. Let $n \in \mathbb{N}$ with $n \geq 1$. We define $\sim$ as follows: for $m < n$ we put $[m]_\sim = \{m\}$, for $m \geq n$ and $m = +\infty$ we define $[m]_\sim = \{k \mid k \geq n \text{ or } k = +\infty\}$. It is an easy exercise to check that $\sim$ is a congruence on the tropical bisemiring. Now $\mathbb{T}/\sim$ is isomorphic to the bisemiring of Example 2.17.

In Chapter 3 we will see another construction to obtain a bisemiring from a given one $\mathbb{K}$: the formal power series over SP^1 and $\mathbb{K}$.

Last but not least, we will mention a negative result. Let $\mathbb{K}$ be a bisemiring and $n \in \mathbb{N}$ with $n \geq 2$. Consider the structure $(M_n(K), \oplus, \circ, \diamond, \mathbf{0}, \mathbf{1})$ of $n \times n$-matrices with entries from K. Here, $\oplus$ is defined elementwise, $\circ$ and $\diamond$ are defined as usual matrix multiplications (by the underlying sequential or parallel multiplication of $\mathbb{K}$, respectively), $\mathbf{0}$ is the zero matrix, and $\mathbf{1}$ the unit matrix. Unfortunately, in general this structure is not a bisemiring because it lacks the commutativity of the parallel multiplication.

3 Weighted Branching Automata and Formal Series

3.1 Weighted Branching Automata and their Behavior

In classical automata theory, finite-state devices were introduced to recognize word languages, *i.e.* subsets of the free monoid $\Sigma^\star$ over a finite alphabet Σ (for an overview see [HMU00, KN01]). Later on, weighted automata over words with weights from a semiring were considered ([SS78, KS86, BR88, Kui97b]). On the other hand, Lodaya and Weil [LW00] presented finite-state devices to recognize languages of sp-posets. With the following model of weighted branching automata we will fuse the concepts of classical weighted automata and of the branching automata of Lodaya and Weil.

We fix a finite alphabet Σ and a bisemiring $\mathbb{K}$. If Q is a set and $m \in \mathbb{N}$, we denote by $\mathfrak{P}_m(Q)$ the collection of all subsets of Q with cardinality m.

Definition 3.1. A *weighted branching automaton*[1], or a *wba* for short, over the alphabet Σ and with weights from the bisemiring $\mathbb{K}$ is a sextuple $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ where

- Q is a finite set of *states*,
- $\mu_{\text{seq}} : Q \times \Sigma \times Q \longrightarrow \mathbb{K}$ is the *sequential transition function*,
- $\mu_{\text{fork}} = \{\mu^m_{\text{fork}} : Q \times \mathfrak{P}_m(Q) \longrightarrow \mathbb{K} \mid m = 2, \ldots, |Q|\}$ is the family of *fork transition functions*,

[1] In [KM03] the notion "branching automaton with costs" was used. However, the notion "weighted automaton" is much more common and, therefore, we use the latter term.

- $\mu_{\text{join}} = \{\mu^m_{\text{join}} : \mathfrak{P}_m(Q) \times Q \longrightarrow \mathbb{K} \mid m = 2, \ldots, |Q|\}$ is the family of *join transition functions*,
- $\lambda, \gamma : Q \longrightarrow \mathbb{K}$ are the *initial* and the *final weight function*, respectively.

Definition 3.2. A *weighted binary branching automaton* (*wbba* for short) is a wba $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ such that for $m > 2$ the functions μ^m_{fork} and μ^m_{join} are identical zero, *i.e.* are mapping every argument to $0 \in K$.

Notation. Usually, in classical automata theory we speak of transitions instead of transition functions. Here, we will do so only if the weight of the transition is distinct from zero. More precisely, we write $p \xrightarrow{a}_k q$ if $\mu_{\text{seq}}(p, a, q) = k \neq 0$ and call it a *sequential transition* from p to q with action a and weight k; if it only matters that the weight is distinct from 0, we write $p \xrightarrow{a} q$. Similarly, we write $p \rightarrow_k \{p_1, \ldots, p_m\}$ and $p \rightarrow \{p_1, \ldots, p_m\}$ if $\mu^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) = k \neq 0$. In the same way, $\{q_1, \ldots, q_m\} \rightarrow_l q$ and $\{q_1, \ldots, q_m\} \rightarrow q$ mean $\mu^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) = l \neq 0$. In these cases we speak of a *fork transition* from p to $\{p_1, \ldots, p_m\}$ with weight k and of a *join transition* from $\{q_1, \ldots, q_m\}$ to q with weight l, respectively. The integer m is called the *arity* of the fork and the join transition, respectively. A state $q \in Q$ is an *initial state* if $\lambda(q) \neq 0$. Dually, q is a *final state* if $\gamma(q) \neq 0$.

Intuitively, if in classical automata theory there is no transition $p \xrightarrow{a} q$ we put here $\mu_{\text{seq}}(p, a, q) = 0$, and similarly for fork and join transitions. Once again, if we speak of a transition in our setting it has always a weight distinct from 0. The initial and final weights may be interpreted as the price for entering and leaving the automaton.

There are two differences between our wba and the branching automata as defined by Lodaya and Weil [LW00]. Firstly, we allow weights for transitions, for initial and final states. Secondly, Lodaya and Weil allowed a forking into and a joining from multisets of states. We favorite sets instead of multisets. We do this mainly for technical reasons. We will point out this later on.

Notation. A weighted branching automaton $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ can be graphically represented by a labeled graph[2] $\mathcal{G}(\mathcal{A})$ as follows (cf. Figure 3.1):

[2]Here we do not speak of a graph in the strong sense of mathematical graph theory but more in the sense of a drawing.

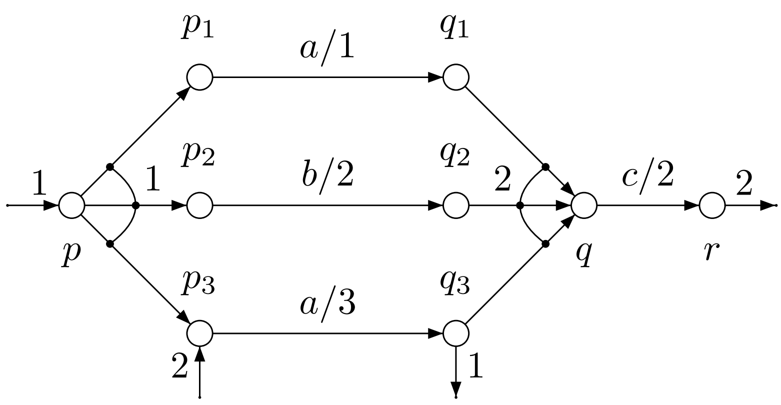


Figure 3.1: Graphical representation of wba.

- the set of vertices of $\mathcal{G}(\mathcal{A})$ is Q,
- for every letter $a \in \Sigma$ and every pair (p, q) of vertices in Q with $\mu_{\text{seq}}(p, a, q) \neq 0$ there is an oriented edge in $\mathcal{G}(\mathcal{A})$ going from p to q and labeled by the pair $a/\mu_{\text{seq}}(p, a, q) \in \Sigma \times K$,[3]
- for every $p \in Q$ and $P \subseteq Q$ with $\mu_{\text{fork}}^{|P|}(p, P) \neq 0$ there are oriented edges (p, p_i) for all $p_i \in P$ (with no label) and a semi-circle connecting these edges labeled by $\mu_{\text{fork}}^{|P|}(p, P) \in K$,
- for every $p \in Q$ and $P \subseteq Q$ with $\mu_{\text{join}}^{|P|}(P, p) \neq 0$ there are oriented edges (p_i, p) for all $p_i \in P$ and a semi-circle connecting these edges labeled by $\mu_{\text{join}}^{|P|}(P, p) \in K$,
- for every $p \in Q$ with $\lambda(p) \neq 0$ there is an in-going arrow to p labeled by $\lambda(p) \in K$,
- for every $p \in Q$ with $\gamma(p) \neq 0$ there is an out-going arrow of p labeled by $\gamma(p) \in K$.

[3] Sometimes we will put the weight of the transition not beside the action but under the arrow because of lack of space, cf. Figure 3.5.

In Figure 3.1 the states p and p_3 are initial with entry weights of 1 and 2, respectively, as r and q_3 are final with weights 2 and 1. There are four sequential transitions, *e.g.* from p_1 to q_1 with label a and weight 1, one fork transition $p \rightarrow_1 \{p_1, p_2, p_3\}$, and one join transition $\{q_1, q_2, q_3\} \rightarrow_2 q$.

3.1.1 The Behavior of Weighted Branching Automata

In order to describe the behavior of a wba, *i.e.* to calculate the weight of an sp-poset t in a wba $\mathcal{A}$, we have to introduce the notion of a run. Two kinds of runs will be defined: one does and the other does not allow a branching in cascades. The first one is closer to the machine level, the second one closer to the maximal decompositions of sp-posets. From these two notions of runs two, possibly different, behaviors of a wba will result. In Chapter 7, we will see that the expressive power of these two concepts is in general not the same.

Recall that in classical word automata a run is a sequence of transitions $(t_i)_i$ such that t_{i+1} starts in the state in which t_i is terminating. This sequence can be seen as a very particular graph with source and sink, and in which vertices are labeled with states and edges are labeled with elements of the alphabet Σ. We generalize those graphs to cover our setting as follows: we consider a set of labeled graphs and define two compositions on them, then we define runs as a subset of this set of graphs demanding the containment of atomic runs and closure under the defined compositions. More formally, we proceed as follows:

Let $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ be a wba over Σ and $\mathbb{K}$. Now let $\mathcal{G}(Q, \Sigma)$ be the set of all labeled directed graphs $G = (V, E, \nu, \eta)$ with $|V| \geq 2$, $E \subseteq V^2$, with a unique source $\text{src}(G)$, a unique sink $\text{sk}(G)$, and with $\nu : V \rightarrow Q$ a total and $\eta : E \dashrightarrow \Sigma$ a partial function. Now we specify two constructions yielding a new element of $\mathcal{G}(Q, \Sigma)$ from given ones.

Let $G_i = (V_i, E_i, \nu_i, \eta_i) \in \mathcal{G}(Q, \Sigma)$ for $i = 1, 2$. If the label of the sink of G_1 is the same as the label of the source of G_2, *i.e.* $\nu_1(\text{sk}(G_1)) = \nu_2(\text{src}(G_2))$, we define the *sequential product* $G = G_1 \cdot G_2$ of G_1 and G_2 as follows (cf. Figure 3.2): G is the disjoint union of both graphs, but the sink of G_1 and the source of G_2 are fused to one vertex with the same label as before.

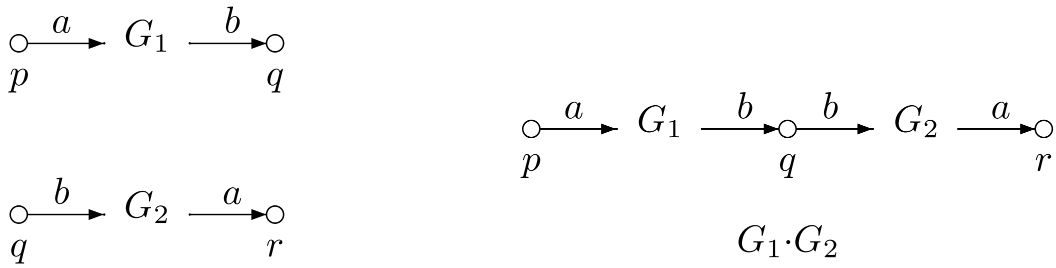


Figure 3.2: Sequential product of G_1 and G_2.

Let $G_i = (V_i, E_i, \nu_i, \eta_i) \in \mathcal{G}(Q, \Sigma)$ and let $p_i = \nu_i(\mathrm{src}(G_i))$ and $q_i = \nu_i(\mathrm{sk}(G_i))$ for $i = 1, \dots, m$. If $p \to \{p_1, \dots, p_m\}$ is a fork and $\{q_1, \dots, q_m\} \to q$ a join transition of $\mathcal{A}$ then

$$\|_{p,q}(G_1, \dots, G_m) = (V, E, \nu, \eta) \in \mathcal{G}(Q, \Sigma)$$

is defined as follows (cf. Figure 3.3):

- $V = V_1 \,\dot\cup\, \dots \,\dot\cup\, V_m \,\dot\cup\, \{u, w\}$,[4]
- $E = E_1 \,\dot\cup\, \dots \,\dot\cup\, E_m \,\dot\cup\, \{(u, \mathrm{src}(G_i)), (\mathrm{sk}(G_i), w) \mid i = 1, \dots, m\}$,
- for $v \in V_i$ put $\nu(v) = \nu_i(v)$ $(i = 1, \dots, m)$, and, furthermore, $\nu(u) = p$, $\nu(w) = q$, and
- $\eta = \eta_1 \,\dot\cup\, \dots \,\dot\cup\, \eta_m$.

This construction is called the *p-q-parallel product* of $G_1, \dots, G_m$.

Note that the sequential product is associative and every p-q-parallel product satisfies a kind of commutativity, *i.e.* for every permutation α of the symmetric group S_m:

$$\|_{p,q}(G_1, \dots, G_m) = \|_{p,q}(G_{\alpha(1)}, \dots, G_{\alpha(m)})$$

in case one side of the equation is defined. But the p-q-parallel products are not associative.

[4] The symbol $\dot\cup$ denotes the disjoint union.

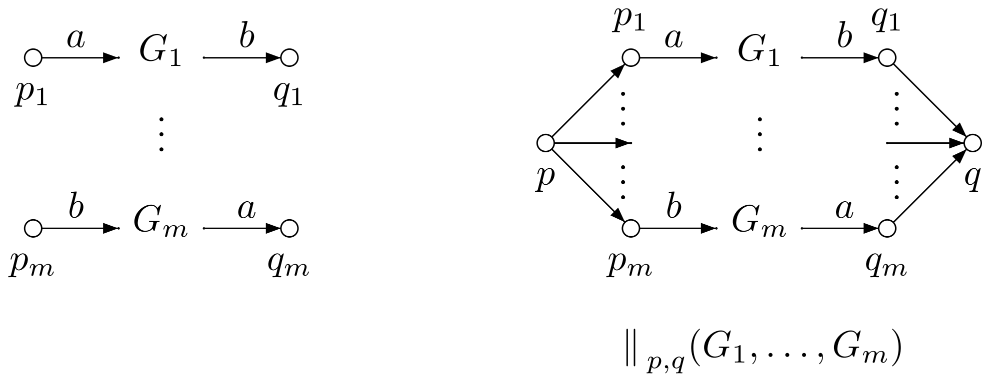


Figure 3.3: The p-q-parallel product of G_1 and G_2.

Now we are ready to define the two notions of runs of a wba $\mathcal{A}$. The graph G with $V = \{v_1, v_2\}$, $E = \{(v_1, v_2)\}$, $\nu(v_i) = p_i$ $(i = 1, 2)$ and $\eta(v_1, v_2) = a$ such that $p_1 \xrightarrow{a} p_2$ is a sequential transition of $\mathcal{A}$ is an element of $\mathcal{G}(Q, \Sigma)$. We refer to such a graph as an *atomic run* of $\mathcal{A}$.

The set of all *cascade branching runs* $\mathcal{R}_C(\mathcal{A})$ is the smallest subset of $\mathcal{G}(Q, \Sigma)$:

- containing all atomic runs, and
- that is closed under sequential product and under all p-q-parallel products for all $p, q \in Q$.

The set of all *maximally branching runs* $\mathcal{R}_M(\mathcal{A})$ is the smallest subset of $\mathcal{G}(Q, \Sigma)$:

- containing all atomic runs,
- closed under sequential product, and
- if $G_1, \ldots, G_m \in \mathcal{R}_M(\mathcal{A})$ are such that G_i is either atomic or a sequential product for all $i = 1, \ldots, m$, then, in case of existence, $\|_{p,q}(G_1, \ldots, G_m) \in \mathcal{R}_M(\mathcal{A})$ for any $p, q \in Q$.

A run from $\mathcal{R}_C(\mathcal{A})$ will be called a *C-run* for short, a run from $\mathcal{R}_M(\mathcal{A})$ an *M-run*.

The difference between the two notions of a run is the restriction for maximally branching runs under which parallel composition is allowed. Whereas for cascade branching runs the factors of a parallel product may be parallel products as well, for maximally branching runs this is forbidden. This is illustrated by Figure 3.4. Here runs are not depicted exactly as they were defined but in the same manner as wba. This way the drawings are easier to understand. But opposed to wba, now states may occur several times. Whereas both graphs are cascade branching runs (assumed all transitions depicted do exist), only the right one is a maximally branching run. Clearly, every maximally branching run is also a cascade branching run but in general not vice versa.

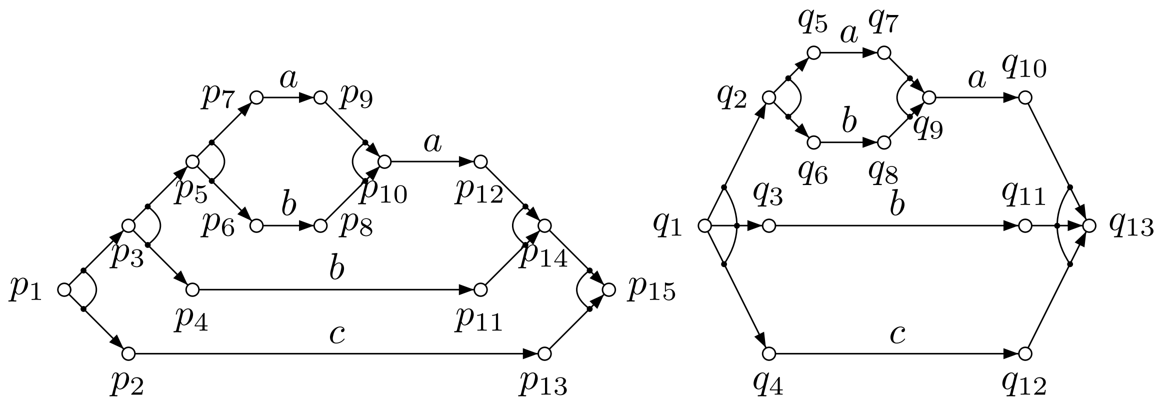


Figure 3.4: Cascade branching and maximally branching run.

The following definitions are both valid for cascade and maximally branching runs. Therefore, we just speak of a run. A run G is *sequential* if G cannot be written as a p-q-parallel product for some $p, q \in Q$. Dually, G is *parallel* if it is not a sequential product. Every atomic run is both sequential and parallel. Similar to sp-posets every run G admits a unique *sequential decomposition* $G = G_1 \cdot \ldots \cdot G_m$ for some $m \geq 1$ such that every G_i is a parallel run ($i = 1, \ldots, m$). On the other hand, every run G has also a unique *parallel decomposition* meaning either G is sequential or $G = \|_{p,q}(G_1, \ldots, G_n)$ for some $p, q \in Q$ and $n \geq 2$. If G is a maximally branching run, then each of the G_i is a sequential run. For G being a cascade branching run the G_i may be sequential or parallel.

By these decompositions we define two functions $\mathbf{lab} : \mathcal{R}(\mathcal{A}) \longrightarrow \mathrm{SP}(\Sigma)$

and $\mathbf{wgt} : \mathcal{R}(\mathcal{A}) \longrightarrow \mathbb{K}$. In the case of classical automata these functions correspond to those which associate to a computation sequence its word and its weight, respectively.

For an atomic run $G : p \xrightarrow{a} q$ we put $\mathbf{lab}(G) = a$ and $\mathbf{wgt}(G) = \mu_{\text{seq}}(p, a, q)$. If $G = G_1 \cdot \ldots \cdot G_m$ is the sequential decomposition of G then

$$\begin{aligned}\mathbf{lab}(G) &= \mathbf{lab}(G_1) \cdot \ldots \cdot \mathbf{lab}(G_m),\\ \mathbf{wgt}(G) &= \mathbf{wgt}(G_1) \circ \ldots \circ \mathbf{wgt}(G_m).\end{aligned}$$

Now let $G = \|_{p,q}(G_1, \ldots, G_n)$ $(n \geq 2)$ be the parallel decomposition of G with $p_i = \nu_i(\text{src}(G_i))$ and $q_i = \nu_i(\text{sk}(G_i))$ for $i = 1, \ldots, n$. Then

$$\mathbf{lab}(G) = \mathbf{lab}(G_1) \parallel \ldots \parallel \mathbf{lab}(G_n),$$

$$\mathbf{wgt}(G) = \mu^n_{\text{fork}}(p, \{p_1, \ldots, p_n\}) \circ \Big[\mathbf{wgt}(G_1) \diamond \ldots \diamond \mathbf{wgt}(G_n)\Big] \circ \mu^n_{\text{join}}(\{q_1, \ldots, q_n\}, q).$$

The weight (or cost) of such a parallel run can be interpreted as follows. Firstly, a weight for branching the process emerges, then the weights for the n sub-processes, and, finally, the weight for joining the sub-processes. These weights occur one after the other and, therefore, are multiplied sequentially. On the other hand, the weights of the n sub-processes are multiplied in parallel.

If a run G has label t we say that G is a run on t. If G is a run on t from p to q ($t \in \text{SP}, p, q \in Q$) then we write $G : p \xrightarrow{t} q$.

From now on, we will often sum up over a set of runs or states. We define the sum to equal zero if the set over which we sum up is empty.

Note that for any $t \in \text{SP}$ there are only finitely many runs G of $\mathcal{A}$ with label t. The weight of some $t \in \text{SP}$ from p to q in $\mathcal{A}$ is given by summing up the weights of all possible runs from p to q with label t:

$$\begin{aligned}\mathbf{wgt}_C(p, t, q) &= \bigoplus_{\substack{G : p \xrightarrow{t} q \\ G \in \mathcal{R}_C(\mathcal{A})}} \mathbf{wgt}(G),\\ \mathbf{wgt}_M(p, t, q) &= \bigoplus_{\substack{G : p \xrightarrow{t} q \\ G \in \mathcal{R}_M(\mathcal{A})}} \mathbf{wgt}(G).\end{aligned}$$

The weight of $t \in \mathrm{SP}$ in $\mathcal{A}$ in cascade and maximally branching mode, respectively, is defined as

$$\begin{aligned}(\mathcal{S}_C(\mathcal{A}), t) &= \bigoplus_{p,q \in Q} \lambda(p) \circ \mathbf{wgt}_C(p, t, q) \circ \gamma(q), \\ (\mathcal{S}_M(\mathcal{A}), t) &= \bigoplus_{p,q \in Q} \lambda(p) \circ \mathbf{wgt}_M(p, t, q) \circ \gamma(q).\end{aligned}$$

Then $\mathcal{S}_C(\mathcal{A}) : \mathrm{SP} \longrightarrow \mathbb{K}$ and $\mathcal{S}_M(\mathcal{A}) : \mathrm{SP} \longrightarrow \mathbb{K}$ are called the *cascade branching behavior* of $\mathcal{A}$, or *C-behavior* for short, and the *maximally branching behavior* of $\mathcal{A}$, or *M-behavior* for short. A function $S : \mathrm{SP} \longrightarrow \mathbb{K}$ is *C-regular* or *M-regular* if there is a wba $\mathcal{A}$ such that $S = \mathcal{S}_C(\mathcal{A})$ or $S = \mathcal{S}_M(\mathcal{A})$, respectively. Equivalently, we say S is *C-recognized* or *M-recognized* by $\mathcal{A}$.

Remark 3.3. For $\mathbb{K} = \mathbb{B}$ the transition functions of a wba $\mathcal{A}$ take values in $\{0, 1\}$ only. Remember that every transition has a weight distinct from 0. Hence, all transitions have weight 1. Similarly, $p \in Q$ is initial (final) iff[5] $\lambda(p) = 1$ ($\gamma(p) = 1$), respectively. Remember that sequential and parallel multiplication in $\mathbb{B}$ are conjunction whereas addition is disjunction. Thus, a run has weight 1 iff all transitions of the run have weight 1. Moreover, $(\mathcal{S}_C(\mathcal{A}), t) = 1$ iff there is a run $G : p \xrightarrow{t} q \in \mathcal{R}_C(\mathcal{A})$ such that p is initial and q final (and similar for $\mathcal{S}_M(\mathcal{A})$). Consequently, wba are a generalization of a sub-class of the branching automata by Lodaya and Weil [LW00] (remember that we use sets instead of multisets for fork and join transitions). Nevertheless, for languages of bounded width wba over the Boolean bisemiring have the same expressive power as the branching automata by Lodaya and Weil. In Chapter 6, we will see that the behaviors of both automata models coincide with the class of sequential-rational sp-languages (cf. Corollary 6.4).

Note that Lodaya and Weil do not define something like an M-behavior for sp-languages. Nevertheless, maximally branching runs have some similarity to runs defined by Lodaya and Weil in [LW01] for branching automata over term algebras with an additional series operation, *i.e.* the parallel product $\parallel$ is understood as a binary term operation. In our notion of M-behavior we consider the maximal parallel decomposition of an sp-poset t, *i.e.* we understand $\parallel$ as a family of m-ary operations for $m \geq 2$.

[5] This term stands for "if and only if".

In Chapter 8, we will make use of the M-behavior of wba, and, in Chapter 7, we will give a comparison of the different notions of regularity. For now we usually deal with C-behaviors of wba. Next, we will give some examples of wba.

3.1.2 Examples of Weighted Branching Automata

Example 3.4. In this example we define a wbba $\mathcal{A}$ whose C-behavior measures the height of an sp-poset t, *i.e.* $(\mathcal{S}_C(\mathcal{A}), t) = \text{height}(t)$ for any sp-poset t. To make this work, we use the tropical bisemiring $(\mathbb{N} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$ from Example 2.6. The automaton has just three states p_0, p_1, p_2. Any of these states can fork into the other states at weight 0; similarly, any two distinct of these states can be joined into the remaining one at weight 0:

$$\mu^2_{\text{fork}}(p_i, \{p_j, p_k\}) = \begin{cases} +\infty & \text{if } |\{i, j, k\}| < 3, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu^2_{\text{join}}(\{p_j, p_k\}, p_i) = \begin{cases} +\infty & \text{if } |\{i, j, k\}| < 3, \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, in any state we can execute any action at weight 1 without changing the state:

$$\mu_{\text{seq}}(p_i, a, p_j) = \begin{cases} 1 & \text{if } i = j, \\ +\infty & \text{otherwise.} \end{cases}$$

Any state is initial and final with $\lambda(p_i) = \gamma(p_i) = 0$ for $i = 0, 1, 2$.

Figure 3.5 depicts a run of $\mathcal{A}$ on the sp-poset $t = (aa \parallel b)(a \parallel bb)$.[6] The weight of this run is evaluated as follows: the run is the sequential product of two "bubbles" whose weights we calculate first. The first "bubble" is the parallel product of an atomic b-run and the sequential aa-run. Since the join and fork transitions involved in this product have weight 0, the

[6] Here $(aa \parallel b)(a \parallel bb)$ abbreviates $((a{\cdot}a) \parallel b){\cdot}(a \parallel (b{\cdot}b))$. That is we drop the sign $\cdot$ to denote the sequential product and agree that the sequential product ties stronger than the parallel one.

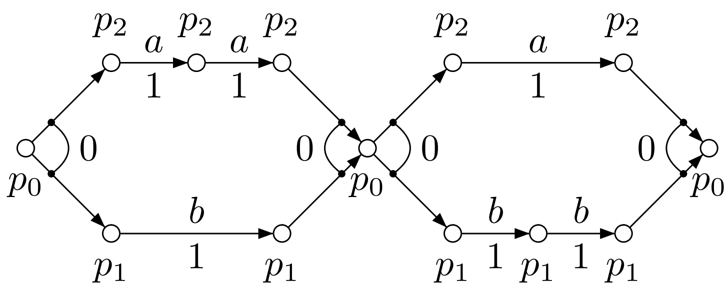


Figure 3.5: A run measuring height and width.

weight of a "bubble" is $0 + \max(1+1, 1) + 0 = 2$. Since this holds for both "bubbles", the total weight is $2 + 2 = 4$ which equals the height of the poset $(aa \parallel b)(a \parallel bb)$. Any permutation of Q gives another run of the same weight, and no further runs on t exist. Hence, $(\mathcal{S}_C(\mathcal{A}), t) = 4 = \text{height}(t)$.

Assumed all the actions to be executed in an sp-poset require one time unit, the automaton given above calculates the execution time. The example can be modified easily to cover the following situation: there is some system in which the execution times of atomic actions depend on the state in which they are executed. One can then construct a wba that calculates the minimal execution time of an sp-poset. If, instead of working in $(\mathbb{N} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$, we work in the polar bisemiring $(\mathbb{N} \cup \{-\infty\}, \max, +, \text{nax}, -\infty, 0)$ of Example 2.7, this automaton would compute the maximal execution time.

Example 3.5. In this example, we present a wbba that measures the width of an sp-poset, *i.e.* $(\mathcal{S}_C(\mathcal{A}), t) = \text{wd}(t)$ for any $t \in \text{SP}$. To this end we take the environmental bisemiring $\mathbb{E} = (\mathbb{R}^{\geq 0} \cup \{+\infty\}, \min, \max, +, +\infty, 0)$ from Example 2.9 and use the automaton from Example 3.4. Consider Figure 3.5 that depicts a run on the sp-poset $(aa \parallel b)(a \parallel bb)$. But this time, the weight of the aa-run is evaluated by $\max(1, 1) = 1$. Hence, the weight of the first "bubble" is $\max(0, 1 + 1, 0) = 2$ and similarly for the second "bubble". Thus, the total weight is $\max(2, 2) = 2$ which equals the width of the poset in question.

Example 3.6. For a Σ-labeled poset $t = (V, \leq, \lambda)$ let $\text{Lin}(t)$ denote the set of all words in Σ^* that label a maximal linearly ordered subset of V. For instance, we have $\mathrm{Lin}\big((aa \parallel b)(a \parallel bb)\big) = \{aaa, aabb, ba, bbb\}$. Next let $\mathcal{W} = (Q, T, \iota, F)$ be a non-deterministic finite automaton[7] that accepts some language $L \subseteq \Sigma^*$. We want to construct a wbba $\mathcal{A}$ such that $(\mathcal{S}_C(\mathcal{A}), t)$ allows us to calculate $\mathrm{Lin}(t) \cap L$ for any sp-poset t (without looking at t again). To make this work, we use the following bisemiring $\mathbb{K}$ of subsets of Σ^*: Let $K = \mathfrak{P}(\Sigma^*) \,\dot\cup\, \{0\}$ where 0 acts as unit with respect to $\oplus$ and absorbing with respect to $\diamond$ and $\circ$. For $A, B \subseteq \Sigma^*$ we then define $A \oplus B = A \diamond B = A \cup B$, $A \circ B = \{u \cdot v \mid u \in A, v \in B\}$. Then $\{\varepsilon\}$ is neutral with respect to sequential multiplication. Now set $Q' = Q \,\dot\cup\, \{\bot_0, \bot_1\}$ and consider the following transition functions:

$$\mu_{\mathrm{seq}}(p, a, q) = \begin{cases} \{a\} & \text{if } (p, a, q) \in T, \\ \emptyset & \text{if } p = q = \bot_i \quad (i = 0, 1), \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu^2_{\mathrm{fork}}(p, \{q, r\}) = \begin{cases} \{\varepsilon\} & \text{if } p \in Q, \{q, r\} = \{p, \bot_0\}, \\ \emptyset & \text{if } p \in \{\bot_0, \bot_1\} = \{q, r\}, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu^2_{\mathrm{join}}(\{q, r\}, p) = \begin{cases} \{\varepsilon\} & \text{if } p \in Q, \{q, r\} = \{p, \bot_0\}, \\ \emptyset & \text{if } p \in \{\bot_0, \bot_1\} = \{q, r\}, \\ 0 & \text{otherwise,} \end{cases}$$

$$\lambda(q) = \begin{cases} \{\varepsilon\} & \text{if } q = \iota, \\ 0 & \text{otherwise,} \end{cases}$$

$$\gamma(q) = \begin{cases} \{\varepsilon\} & \text{if } q \in F, \\ 0 & \text{otherwise.} \end{cases}$$

We claim that for $t \in \mathrm{SP}$ with $\mathrm{Lin}(t) \cap L \neq \emptyset$, we get $(\mathcal{S}_C(\mathcal{A}), t) = \mathrm{Lin}(t) \cap L$, and otherwise $(\mathcal{S}_C(\mathcal{A}), t) = 0$.

The formal proof of this claim is quite tedious. Here we will present the main idea of how $\mathcal{A}$ works only(cf. Figure 3.6). $\mathcal{A}$ simulates the automaton

[7] Recall that $T \subseteq Q \times \Sigma \times Q$ is the set of transitions, $\iota \in Q$ the unique initial state, and $F \subseteq Q$ the set of final states.

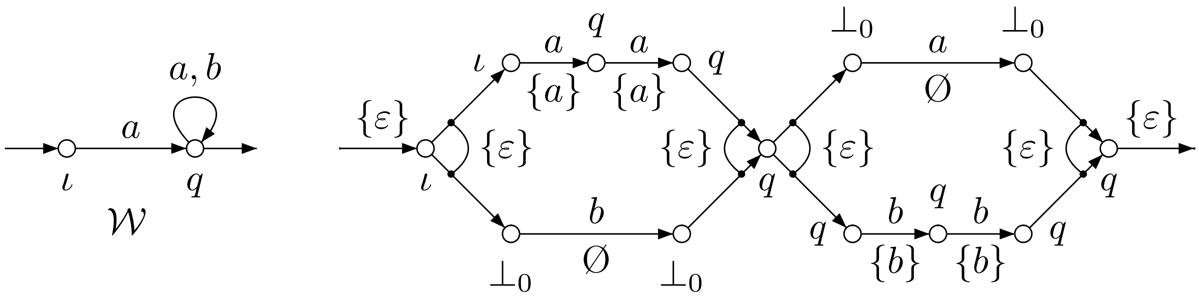


Figure 3.6: A run of $\mathcal{A}$ on $(aa \parallel b)(a \parallel bb)$ with weight $\{aabb\}$.

$\mathcal{W}$ by its sequential transitions: if $w \in \mathrm{SP}$ is actually a word, *i.e.* $\mathrm{wd}(w) = 1$, then w has a run in $\mathcal{W}$ from p to q iff there is the same run in $\mathcal{A}$ from p to q with label w and weight $\{w\}$. Moreover, for every $w \in \Sigma^+$ there is a run in $\mathcal{A}$ from $\perp_0$ to $\perp_0$ and from $\perp_1$ to $\perp_1$ with weight $\emptyset$. Now let $t \in \mathrm{SP}$. $\mathcal{A}$ starts to run on t at the unique initial state ι. Let $t = t_1 \cdot \ldots \cdot t_m$ $(m \geq 1)$ be the sequential decomposition of t and let t_i be the first factor in this decomposition which is not atomic but a parallel product. When $\mathcal{A}$ starts to run at t_i then $\mathcal{A}$ has to branch from some state $p \in Q$. By the definition of μ_{fork}, it branches into p and $\perp_0$ with neutral weight $\{\varepsilon\}$. Now $\mathcal{A}$ can continue in state p with one of the parallel factors of t_i. Altogether this means $\mathcal{A}$ guesses one branch in every fork and continues to simulate $\mathcal{W}$ in this branch. It does the same with every fork. In the branches not simulating $\mathcal{W}$ the definition of the transition functions guarantees that $\mathcal{A}$ may execute any sp-poset with weight $\emptyset$. Hence, a run of $\mathcal{A}$ on t has either a weight $\{w\}$ with $w \in \mathrm{Lin}(t)$ (if the chosen simulated run is indeed a run of $\mathcal{W}$) or has weight 0 (otherwise). Moreover, a run on t is successful, *i.e.* going from ι to some $q \in F$ and having weight $\{w\}$ distinct from 0 iff $w \in L$. Because $\mathcal{A}$ may guess all $w \in \mathrm{Lin}(t)$ and the sum of the bisemiring is mainly identical to the union, $\mathcal{A}$ calculates all $w \in \mathrm{Lin}(t)$ that are in L. Hence,

$$(\mathcal{S}_C(\mathcal{A}), t) = \begin{cases} \mathrm{Lin}(t) \cap L & \text{if } \mathrm{Lin}(t) \cap L \neq \emptyset, \\ 0 & \text{otherwise} \end{cases}$$

for all $t \in \mathrm{SP}$.

As an example consider Figure 3.6. Here $\Sigma = \{a, b\}$. The automaton

$\mathcal{W}$ is depicted on the left hand side. Its recognized language is $L = \{au \mid u \in \Sigma^\star\}$. On the right hand side, a successful run of the associated wbba $\mathcal{A}$ on $t = (aa \parallel b)(a \parallel bb)$ is depicted. The guessed run of $\mathcal{W}$ is on the word $aabb$. Hence, the weight of t at this run is $\{aabb\}$. Moreover, $\mathcal{S}_C(\mathcal{A}, t) = \{aabb, aaa\}$.

Since $\Sigma^\star$ is a regular word language there is a wbba $\mathcal{A}$ with $(\mathcal{S}_C(\mathcal{A}), t) = \mathrm{Lin}(t)$ for all $t \in \mathrm{SP}$.

3.2 Formal Power Series over Sequential-Parallel Posets

To characterize the possible behaviors of wba, 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. [SS78]) and of sp-languages as introduced by Lodaya and Weil [LW00].

A *formal power series over* SP *with values in the bisemiring* $\mathbb{K}$, or an *sp-series* for short, 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 value (S, t) is referred to as the *coefficient* of t in S. The terminology of an sp-series underlines that we are interested in various operations similar to those as defined for series in analysis. The power series is called “formal” because we are not interested in summing up series. The *support* of S is $\operatorname{supp} S := \{t \in \mathrm{SP} \mid (S, t) \neq 0\}$. Formal power series with finite support are called *polynomials*, those whose support is a letter are called *monomials*. The set of all formal power series over SP with values in $\mathbb{K}$ is denoted by $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$, the set of all polynomials is denoted by $\mathbb{K}\langle \mathrm{SP} \rangle$.

Similarly, we can consider formal power series over SP^1 with values in a bisemiring $\mathbb{K}$, *i.e.* functions $S : \mathrm{SP}^1 \longrightarrow \mathbb{K}$. All the above definitions apply also to sp-series over SP^1. We abbreviate the set of all sp-series over SP^1 by $\mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$, and the set of all polynomials by $\mathbb{K}\langle \mathrm{SP}^1 \rangle$. Obviously, any series from $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ can be understood as a series from $\mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$ by putting $(S, \varepsilon) = 0$. Vice versa, every series $S \in \mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$ with $(S, \varepsilon) = 0$ can be seen as a series of $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$.

Now we introduce some operations for sp-series. Let $S, T \in \mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$. We define for all $t \in \mathrm{SP}^1$:

- the *sum* $S + T$ by

$$(S + T, t) := (S, t) \oplus (T, t),$$

- the *scalar products* $k \cdot S$ and $S \cdot k$ for $k \in \mathbb{K}$ by

$$(k \cdot S, t) := k \circ (S, t) \text{ and } (S \cdot k, t) := (S, t) \circ k,$$

- 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 $u, v \in \mathrm{SP}^1$,

- the *parallel product* $S \parallel T$ by

$$(S \parallel T, t) := \bigoplus_{(u,v)\,:\, t = u \parallel v} (S, u) \diamond (T, v)$$

  where we add over all pairs (u, v) such that $t = u \parallel v$ with $u, v \in \mathrm{SP}^1$ (because of the commutativity of $\parallel$ in SP^1 both $(S, u) \diamond (T, v)$ and $(S, v) \diamond (T, u)$ contribute to the sum).

Sum, scalar products, sequential and parallel product are defined likewise for $S, T \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$.

For the next two operations let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$. Then we define for all $t \in \mathrm{SP}$:

- the *sequential iteration* S^+ of an sp-series S by

$$(S^+, t) := \bigoplus_{1 \le m \le |t|} \bigoplus_{t = u_1 \cdot \ldots \cdot u_m} (S, u_1) \circ \ldots \circ (S, u_m)$$

  where we sum up over all possible sequential factorizations of t,

- the *parallel iteration* $S^{\boxplus}$ by

$$(S^{\boxplus}, t) := \bigoplus_{1 \le n \le |t|} \bigoplus_{\substack{(u_1, \ldots, u_n): \\ t = u_1 \,\|\, \ldots \,\|\, u_n}} (S, u_1) \diamond \ldots \diamond (S, u_n)$$

  where the sum extends over all parallel factorizations and any order of the factors.

Collectively, we refer to the defined operations as the *rational operations*. Moreover, all rational operations without the parallel iteration are referred to as the *sequential-rational operations*.

Remark 3.7. If $(\mathrm{SP}^1, \cdot)$ is viewed as the free monoid over the infinite alphabet of all parallel posets, then $S \cdot T$ is the usual definition of a Cauchy product (cf. [SS78]). The same holds true for $S \,\|\, T$ if $(\mathrm{SP}^1, \|\,)$ is seen as the free commutative monoid over the infinite alphabet of all sequential sp-posets.

The series $\mathbb{0}$ is defined by $(\mathbb{0}, t) = 0$ for all $t \in \mathrm{SP}^1$, the series $\mathbb{1}_\varepsilon$ by $\mathbb{1}_\varepsilon(\varepsilon) = 1$ and $\mathbb{1}_\varepsilon(t) = 0$ for any $t \neq \varepsilon$.

Lemma 3.8. $(\mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle, +, \cdot, \|\,, \mathbb{0}, \mathbb{1}_\varepsilon)$ *and* $(\mathbb{K}\langle \mathrm{SP}^1 \rangle, +, \cdot, \|\,, \mathbb{0}, \mathbb{1}_\varepsilon)$ *are bi-semirings.*

Proof. Because of Remark 3.7 the proof of associativity for $+, \cdot$, and $\|$ is by standard arguments. It is clear that $+$ is commutative. For the sake of completeness we include the argument for commutativity of $\|\,$. For

$S, T \in \mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$ and $t \in \mathrm{SP}^1$ we have:

$$\begin{aligned}(S \parallel T, t) &= \bigoplus_{(u,v)\,:\, t=u \parallel v} (S,u) \diamond (T,v) \\ &= \bigoplus_{(u,v)\,:\, t=u \parallel v} (T,v) \diamond (S,u) \\ &= \bigoplus_{(v,u)\,:\, t=v \parallel u} (T,v) \diamond (S,u) \\ &= (T \parallel S, t).\end{aligned}$$

It is easy to check that $\cdot$ and $\parallel$ distribute over $+$. Clearly, $\mathbb{0} + S = S$, $\mathbb{0} \cdot S = S \cdot \mathbb{0} = S \parallel \mathbb{0} = \mathbb{0}$ and $\mathbb{1}_\varepsilon \cdot S = S \cdot \mathbb{1}_\varepsilon$ for all $S \in \mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle$. Hence, $(\mathbb{K}\langle\langle \mathrm{SP}^1 \rangle\rangle, +, \cdot, \parallel, \mathbb{0}, \mathbb{1}_\varepsilon)$ is a bisemiring. Since $+, \cdot$, and $\parallel$ preserve finite supports, $(\mathbb{K}\langle \mathrm{SP}^1 \rangle, +, \cdot, \parallel, \mathbb{0}, \mathbb{1}_\varepsilon)$ is also a bisemiring. $\square$

Note. $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ and $\mathbb{K}\langle \mathrm{SP} \rangle$ do not carry the structure of a bisemiring because the unit $\mathbb{1}_\varepsilon$ of the sequential product is missing.

Remark 3.9. In Definition 3.1 of a wba $\mathcal{A}$ and in the definition of the behavior of $\mathcal{A}$ we have allowed neither ε-transitions nor a run of $\mathcal{A}$ on ε. Therefore, the behavior of a wba $\mathcal{A}$ is always an element of $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$. Hence, we deal with $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ in the sequel.

Definition 3.10. (a) The class $\mathbb{K}^{\mathrm{s-rat}}\langle\langle \mathrm{SP} \rangle\rangle$ of *sequential-rational sp-series*[8] over Σ with values in $\mathbb{K}$ is the smallest class $\mathcal{C}$ of $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ such that

- all monomials are in $\mathcal{C}$, and
- $\mathcal{C}$ is closed under all sequential-rational operations.

(b) The class $\mathbb{K}^{\mathrm{rat}}\langle\langle \mathrm{SP} \rangle\rangle$ of *rational sp-series* over Σ with values in $\mathbb{K}$ is the smallest class $\mathcal{D}$ of $\mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ such that

- all monomials are in $\mathcal{D}$, and
- $\mathcal{D}$ is closed under all rational operations.

[8] Called "series-rational" in [KM03].

Another way to define rational and sequential-rational sp-series is by *rational* and *sequential-rational expressions*. They and their associated sp-series are defined inductively as follows:

1. if $a \in \Sigma$ then a is a rational and a sequential-rational expression and $[\![\, a \,]\!] = S$ with $(S, a) = 1$ and $(S, t) = 0$ otherwise;
2. if E is a rational (sequential-rational) expression and $k \in \mathbb{K}$ then $k{\cdot}E$ and $E{\cdot}k$ are rational (sequential-rational) expressions, and $[\![\, k{\cdot}E \,]\!] = k{\cdot}[\![\, E \,]\!]$, $[\![\, E{\cdot}k \,]\!] = [\![\, E \,]\!]{\cdot}k$;
3. if E_1, E_2 are rational (sequential-rational) expressions then $E_1 + E_2$, $E_1{\cdot}E_2$, and $E_1 \parallel E_2$ are rational (sequential-rational) expressions, and $[\![\, E_1 + E_2 \,]\!] = [\![\, E_1 \,]\!] + [\![\, E_2 \,]\!]$, $[\![\, E_1{\cdot}E_2 \,]\!] = [\![\, E_1 \,]\!]{\cdot}[\![\, E_2 \,]\!]$, $[\![\, E_1 \parallel E_2 \,]\!] = [\![\, E_1 \,]\!] \parallel [\![\, E_2 \,]\!]$;
4. if E is a rational (sequential-rational) expression then E^+ is a rational (sequential-rational) expression, and $[\![\, E^+ \,]\!] = [\![\, E \,]\!]^+$;
5. if E is a rational expression then $E^{\boxplus}$ is a rational expression, and $[\![\, E^{\boxplus} \,]\!] = [\![\, E \,]\!]^{\boxplus}$.

Now, an sp-series S is rational (sequential-rational) if $[\![\, E \,]\!] = S$ for some rational (sequential-rational) expression E.

Note. Already in the theory of sp-languages the parallel iteration causes severe problems [LW00]. Smoother results are obtained if one does not allow the parallel iteration [LW00]. This restriction seems natural to us because of the boundedness of the number of independent processes in a parallel system.

Remark 3.11. Σ^+ can be identified with those sp-posets of $\mathrm{SP}(\Sigma)$ built of singletons by the use of sequential product only. If $\mathbb{K} = (K, \oplus, \circ, \diamond, 0, 1)$, then $\mathbb{K}' = (K, \oplus, \circ, 0, 1)$ is a semiring. Thus, a formal power series $S' : \Sigma^+ \to \mathbb{K}'$ may be identified with the series $S : \mathrm{SP}(\Sigma) \to \mathbb{K}$ where

$$(S, t) = \begin{cases} (S', t) & \text{if } t \in \Sigma^+, \\ 0 & \text{otherwise.} \end{cases}$$

The rational formal power series over Σ^+ and $\mathbb{K}'$ are the smallest class of all formal power series containing the monomials and closed under sum, sequential product and sequential iteration. It is denoted by $\mathbb{K}'^{\mathrm{rat}}\langle\langle\Sigma^+\rangle\rangle$. Hence, $\mathbb{K}^{\mathrm{s-rat}}\langle\langle \mathrm{SP}(\Sigma)\rangle\rangle$ is a generalization of $\mathbb{K}'^{\mathrm{rat}}\langle\langle\Sigma^+\rangle\rangle$.[9]

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 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 because f is non-erasing, *i.e.* $|t| \leq |f(t)|$ for all $t \in \mathrm{SP}(\Sigma_1)$ because $\varepsilon \notin \mathrm{SP}(\Sigma_2)$.

Proposition 3.12. *Let $h : \mathbb{K} \to \mathbb{K}'$ be a bisemiring homomorphism and $f : \mathrm{SP}(\Sigma_1) \to \mathrm{SP}(\Sigma_2)$ an sp-algebra homomorphism. For any $k \in \mathbb{K}$ and $S \in \mathbb{K}\langle\langle \mathrm{SP}(\Sigma_1)\rangle\rangle$, $\overline{h}(k\cdot S) = h(k)\cdot\overline{h}(S)$, and $\overline{h}(S\cdot k) = \overline{h}(S)\cdot h(k)$. Further, $\overline{h}$ commutes with all other rational operations, and $\overleftarrow{f}$ commutes with all rational operations. In particular, $\overline{h}$ and $\overleftarrow{f}$ preserve sequential-rationality and rationality.*

Proof. The proof for $\overline{h}$ is straightforward. Obviously, $\overleftarrow{f}$ commutes with sum and scalar products. Because of Remark 3.7 it is folklore to show that $\overleftarrow{f}$ commutes with the sequential product. We include the argument for the sake of completeness. Let $t \in \mathrm{SP}(\Sigma_2)$, and $S, T \in \mathbb{K}\langle\langle \mathrm{SP}(\Sigma_1)\rangle\rangle$. Then we have:

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

On the other hand:

$$\begin{aligned}(\overleftarrow{f}(S)\cdot\overleftarrow{f}(T), t) &= \bigoplus_{u',v':\ u'\cdot v'=t} (\overleftarrow{f}(S), u') \circ (\overleftarrow{f}(T), v')\\ &= \bigoplus_{u',v':\ u'\cdot v'=t} \Big[\bigoplus_{u\in f^{-1}(u')} (S, u)\Big] \circ \Big[\bigoplus_{v\in f^{-1}(v')} (T, v)\Big]\end{aligned}$$

[9] Usually, $\mathbb{K}'\langle\langle\Sigma^\star\rangle\rangle$ and $\mathbb{K}'^{\mathrm{rat}}\langle\langle\Sigma^\star\rangle\rangle$ are considered. Then $\mathbb{K}\langle\langle \mathrm{SP}^1\rangle\rangle$ and $\mathbb{K}^{\mathrm{s-rat}}\langle\langle \mathrm{SP}^1\rangle\rangle$ would be the appropriate generalizations. But we restrict ourselves to SP as noticed above.

$$= \bigoplus_{u',v' \colon u' \cdot v' = t} \quad \bigoplus_{\substack{u \in f^{-1}(u') \\ v \in f^{-1}(v')}} (S,u) \circ (T,v)$$

$$= \bigoplus_{u,v \colon f(u) \cdot f(v) = t} (S,u) \circ (T,v).$$

Since $f : \mathrm{SP}(\Sigma_1) \to \mathrm{SP}(\Sigma_2)$ is a homomorphism, $\{(u,v) \mid f(u \cdot v) = t\} = \{(u,v) \mid f(u) \cdot f(v) = t\}$. Thus, $\overleftarrow{f}$ commutes with the sequential product. Similarly, one shows that $\overleftarrow{f}$ commutes with the parallel product, the sequential iteration, and the parallel iteration. $\square$

We consider as a special case the Boolean bisemiring $\mathbb{B}$. An *sp-language* L is a subset of $\mathrm{SP}(\Sigma)$. Any sp-language $L \subseteq \mathrm{SP}$ can be identified with its *characteristic series* $\mathbb{1}_L$ over $\mathbb{B}$ where $\mathbb{1}_L : \mathrm{SP} \to \mathbb{B}$ and $(\mathbb{1}_L, t) = 1$ iff $t \in L$. Hence, $\operatorname{supp} \mathbb{1}_L = L$. The operations introduced above can now be seen as operations of sp-languages. Let $L, L' \subseteq \mathrm{SP}$. We define:

- union or sum: $L \cup L'$,
- sequential product: $L \cdot L' := \{t \cdot t' \mid t \in L, t' \in L'\}$,
- parallel product: $L \parallel L' := \{t \parallel t' \mid t \in L, t' \in L'\}$,
- sequential iteration: $L^{+} := \{t_1 \cdot \ldots \cdot t_m \mid m > 0, t_i \in L\}$,
- parallel iteration: $L^{\boxplus} := \{t_1 \parallel \ldots \parallel t_n \mid n > 0, t_i \in L\}$.

Then $\mathrm{char} : \mathfrak{P}(\mathrm{SP}) \to \mathbb{B}\langle\langle \mathrm{SP} \rangle\rangle : L \mapsto \mathbb{1}_L$ and $\mathrm{supp} : \mathbb{B}\langle\langle \mathrm{SP} \rangle\rangle \to \mathfrak{P}(\mathrm{SP}) : S \mapsto \mathrm{supp}(S)$ are bijections preserving the operations defined above. Therefore, the theory of sp-series is a generalization of the theory of sp-languages as investigated by Lodaya and Weil [LW00]. This bijection maps the class $\mathbb{B}^{\mathrm{s-rat}}\langle\langle \mathrm{SP} \rangle\rangle$ to the class of *sequential-rational sp-languages*[10] $\mathrm{SP}^{s-rat}(\Sigma)$, *i.e.* the least class $\mathcal{C}$ of subsets of $\mathrm{SP}(\Sigma)$ such that

- $\emptyset$ and all singletons are in $\mathcal{C}$, and
- $\mathcal{C}$ is closed under union, sequential and parallel product, and under sequential iteration.

[10] Called "series-rational languages" in [LW00].

Since the iteration of the parallel product is not allowed in the construction of a sequential-rational sp-language we have immediately that any $L \in \mathrm{SP}^{s-rat}$ is width-bounded [LW00]. For cardinality reasons there are width-bounded sp-languages which are not sequential-rational. Or consider any non-regular word language like $L = \{a^n b^n \mid n \geq 1\}$ which has uniform width of 1 but which is not sequential-rational.

We call an sp-series $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ *width-bounded* if $\operatorname{supp} S$ has bounded width.

Proposition 3.13. *Any sequential-rational sp-series has bounded width.*

Proof. Note the following relations for $S, T \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ and $k \in \mathbb{K}$:

- $\operatorname{supp}(S + T) \subseteq \operatorname{supp} S \cup \operatorname{supp} T$,
- $\operatorname{supp}(k \cdot S) \subseteq \operatorname{supp} S$ and $\operatorname{supp}(S \cdot k) \subseteq \operatorname{supp} S$,
- $\operatorname{supp}(S \cdot T) \subseteq (\operatorname{supp} S) \cdot (\operatorname{supp} T)$,
- $\operatorname{supp}(S \parallel T) \subseteq (\operatorname{supp} S) \parallel (\operatorname{supp} T)$, and
- $\operatorname{supp}(S^+) \subseteq \big(\operatorname{supp} S\big)^+$.

Now, let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be sequential-rational. Due to the relations given above there is a sequential-rational language $L \subseteq \mathrm{SP}$ with $\operatorname{supp} S \subseteq L$. Since L is of bounded width, $\operatorname{supp} S$ is of bounded width. □

As for sp-languages the converse of Proposition 3.13 is not true.

4 First Closure Properties of Regular Sequential-Parallel Series

In this chapter, we show the closure of C-regular sp-series under rational operations. Most of these closure properties are also valid for M-regular sp-series. The main theorem of this chapter states that every rational sp-series is C-regular.

Notation. In the sequel, many proofs are the same for C-regular and M-regular sp-series. Therefore, we speak in these proofs just of "a run" and of "the behavior". If we do so we mean either uniformly C-runs, C-behavior etc. or M-runs, M-behavior etc. The point is that the argument remains the same in cascade branching and maximally branching mode.

4.1 Closure under Sum and Scalar Products

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

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

$$\mu_{\text{seq}}(p, a, q) = \begin{cases} \mu_{i_{\text{seq}}}(p, a, q) & \text{if } p, q \in Q_i \ (i = 1, 2), \\ 0 & \text{otherwise,} \end{cases}$$

and similarly for $\mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma$.

Then a run of $\mathcal{A}$ is either completely in $\mathcal{A}_1$ or in $\mathcal{A}_2$. Thus, $\mathcal{A}$ recognizes $S_1 + S_2$. □

Proposition 4.2. *Let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be a C-regular (M-regular) sp-series and $k \in \mathbb{K}$. Then both $k \cdot S$ and $S \cdot k$ are C-regular (M-regular),respectively.*

Proof. Let S be recognized by the wba $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$. We define $\mathcal{A}'$ being equal to $\mathcal{A}$ with the exception of the initial weight function which we put

$$\lambda'(p) = k \circ \lambda(p)$$

for all $p \in Q$. Then $\mathcal{A}'$ recognizes the sp-series $k{\cdot}S$.

A similar proof holds for the series $S{\cdot}k$ changing the final weights instead of the initial ones. □

Since the constructions in the proofs of Proposition 4.1 and 4.2 do not introduce fork or join transitions of new arity we get:

Corollary 4.3. *Let $S, S_1, S_2 \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be C-recognized (M-recognized) by some wbba and let $k \in \mathbb{K}$. Then $k{\cdot}S$, $S{\cdot}k$, and $S_1 + S_2$ are C-recognized (M-recognized) by a wbba.*

4.2 Normalization

In the sequel, we will use branching automata with restricted possibilities to enter and to leave the automaton. A wba $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ is called *initial-state-normalized* if there is one unique initial state $\mathfrak{i}$ only with $\lambda(\mathfrak{i}) = 1$, $\mu_{\text{seq}}(p, a, \mathfrak{i}) = 0$, and

$$\mu^m_{\text{join}}(\{p_1, \ldots, p_m\}, \mathfrak{i}) = \mu^m_{\text{fork}}(p_1, \{\mathfrak{i}, p_2, \ldots, p_m\}) = 0$$

for all $m = 2, \ldots, |Q|$, $p, p_1, \ldots, p_m \in Q$ and $a \in \Sigma$. The wba $\mathcal{A}$ is *final-state-normalized* if there is one unique final state $\mathfrak{f}$ only with $\gamma(\mathfrak{f}) = 1$, $\mu_{\text{seq}}(\mathfrak{f}, a, q) = 0$, and

$$\mu^m_{\text{fork}}(\mathfrak{f}, \{q_1, \ldots, q_m\}) = \mu^m_{\text{join}}(\{\mathfrak{f}, q_2, \ldots, q_m\}, q) = 0$$

for all $m = 2, \ldots, |Q|$, $q, q_1, \ldots, q_m \in Q$ and $a \in \Sigma$. If $\mathcal{A}$ is both initial- and final-state-normalized then $\mathcal{A}$ is said to be *normalized*.

Proposition 4.4. *Let $\mathcal{A}$ be a wba over Σ with weights from $\mathbb{K}$. Then there is a normalized wba with the same behavior as $\mathcal{A}$.*

Proof. We show how to transform $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ into an initial-state-normalized wba. The wba $\mathcal{A}_I = (Q_I, \mu_{I_{\text{seq}}}, \mu_{I_{\text{fork}}}, \mu_{I_{\text{join}}}, \lambda_I, \gamma_I)$ is defined as follows:

- $Q_I = Q \mathbin{\dot\cup} \{\mathfrak{i}\}$,
- $\mu_{I_{\text{seq}}}(p, a, q) = \begin{cases} \mu_{\text{seq}}(p, a, q) & \text{if } p, q \in Q, \\ \bigoplus_{r \in Q} [\lambda(r) \circ \mu_{\text{seq}}(r, a, q)] & \text{if } p = \mathfrak{i}, q \in Q, \\ 0 & \text{otherwise,} \end{cases}$
- for any $m \in \{2, \ldots, |Q_I|\}$:

$$\mu_{I_{\text{fork}}}^{m}(p, \{p_1, \ldots, p_m\})$$

$$= \begin{cases} \mu_{\text{fork}}^{m}(p, \{p_1, \ldots, p_m\}) & \text{if } p, p_1, \ldots, p_m \in Q, \\ \bigoplus_{r \in Q} [\lambda(r) \circ \mu_{\text{fork}}^{m}(r, \{p_1, \ldots, p_m\})] & \text{if } p = \mathfrak{i},\ p_1, \ldots, p_m \in Q, \\ 0 & \text{otherwise,} \end{cases}$$

- for any $m \in \{2, \ldots, |Q_I|\}$:

$$\mu_{I_{\text{join}}}^{m}(\{q_1, \ldots, q_m\}, q)$$

$$= \begin{cases} \mu_{\text{join}}^{m}(\{q_1, \ldots, q_m\}, q) & \text{if } q_1, \ldots, q_m, q \in Q, \\ 0 & \text{otherwise,} \end{cases}$$

- $\lambda_I(p) = \begin{cases} 1 & \text{if } p = \mathfrak{i}, \\ 0 & \text{if } p \in Q, \end{cases}$ and $\gamma_I(q) = \begin{cases} 0 & \text{if } q = \mathfrak{i}, \\ \gamma(q) & \text{if } q \in Q. \end{cases}$

Clearly, $\mathcal{A}_I$ is initial-state-normalized with $\mathfrak{i}$ as its unique initial state. Now we show $\mathcal{S}(\mathcal{A}_I) = \mathcal{S}(\mathcal{A})$. For a run $G = (V, E, \nu, \eta)$ of $\mathcal{A}$ we define $G' = (V, E, \nu', \eta)$ by $\nu'(\text{src}(G')) = \mathfrak{i}$ and $\nu'(v) = \nu(v)$ for all $v \in V \setminus \{\text{src}(G')\}$. If G' is a run of $\mathcal{A}_I$, we put $g(G) = G'$; otherwise $g(G)$ is undefined.

Any run G of $\mathcal{A}$ starts either with a sequential transition $p \xrightarrow{a} q$ or with a fork transition $p \rightarrow \{p_1, \ldots, p_m\}$. Note that $g(G)$ is undefined iff $\mu_{I_{\text{seq}}}(\mathfrak{i}, a, q) = 0$ or $\mu_{I_{\text{fork}}}^{m}(\mathfrak{i}, \{p_1, \ldots, p_m\}) = 0$, respectively.

Clearly, in case $g(G)$ is defined, $\mathbf{lab}(g(G)) = \mathbf{lab}(G)$ and every run of $\mathcal{A}_I$ whose source is labeled with $\mathfrak{i}$ is in the image of g. If G' is a run of $\mathcal{A}_I$ starting with initial state $\mathfrak{i}$ then we have

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

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

$$0 = \mu_{I_{\mathrm{seq}}}(\mathfrak{i}, a, q) = \bigoplus_{r \in Q} [\lambda(r) \circ \mu_{\mathrm{seq}}(r, a, q)] \ .$$

Hence, we have

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

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

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

Therefore, it is sufficient for calculating $(\mathcal{S}(\mathcal{A}), t)$ to sum up only over the runs G with $g(G) \in \mathcal{R}(\mathcal{A}_I)$. Hence, we get for any $t \in \mathrm{SP}$

$$\begin{aligned}
(\mathcal{S}(\mathcal{A}_I), t) &= \bigoplus_{q \in Q} \Big[\Big(\bigoplus_{G' : \mathfrak{i} \xrightarrow{t} q} \mathbf{wgt}(G') \Big) \circ \gamma_I(q) \Big] \\
&= \bigoplus_{q \in Q} \Big[\Big(\bigoplus_{G \in g^{-1}(G')} \lambda(\nu(\mathrm{src}(G))) \circ \mathbf{wgt}(G) \Big) \circ \gamma(q) \Big] \\
&= \bigoplus_{r, q \in Q} \lambda(r) \circ \mathbf{wgt}(r, t, q) \circ \gamma(q) \\
&= (\mathcal{S}(\mathcal{A}), t).
\end{aligned}$$

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

Since the construction in the proof given above does not introduce any fork or join transitions of an arity not already existent in the original wba $\mathcal{A}$ we have:

Corollary 4.5. *For any wbba $\mathcal{A}$ there is a normalized wbba $\mathcal{A}'$ with the same behavior as $\mathcal{A}$.*

4.3 Closure under Parallel Product and Iteration

Next, we construct from two wba $\mathcal{A}_1$ and $\mathcal{A}_2$ a wba $\mathcal{A}$ with the C-behavior $\mathcal{S}_C(\mathcal{A}) = \mathcal{S}_C(\mathcal{A}_1) \,\|\, \mathcal{S}_C(\mathcal{A}_2)$. At first sight, we would try to take $\mathcal{A}$ the disjoint union of $\mathcal{A}_1$ and $\mathcal{A}_2$, adding two states $\mathfrak{i}$ and $\mathfrak{f}$ as initial and final state, and, moreover, adding fork transitions $\mathfrak{i} \to \{p_1, p_2\}$ where p_i is initial in $\mathcal{A}_i$, and, similarly, adding join transitions $\{q_1, q_2\} \to \mathfrak{f}$ where q_i is final in $\mathcal{A}_i$ for $i = 1, 2$. But this construction fails in general. We cannot concentrate the old initial weights of $\mathcal{A}_1$ and $\mathcal{A}_2$ in the new fork transitions $\mathfrak{i} \to \{p_1, p_2\}$ or in the initial weight of $\mathfrak{i}$ because then the initial weights of $\mathcal{A}_1$ and $\mathcal{A}_2$ would not be multiplied in parallel anymore. But this is necessary for the desired behavior of $\mathcal{A}$. The construction can only be successful if all initial and final weights 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 4.4.

Proposition 4.6. *Let S_1 and S_2 be C-regular sp-series over Σ and $\mathbb{K}$. Then $S_1 \parallel S_2$ is C-regular.*

Proof. We give only a sketch of the proof. Let $\mathcal{A}_i$ be a normalized wba with $\mathcal{S}_C(\mathcal{A}_i) = S_i$ for $i = 1, 2$. Moreover, let $\mathfrak{i}_i$ and $\mathfrak{f}_i$ 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 $\mathfrak{i}$ and $\mathfrak{f}$ and, moreover, a fork $\mathfrak{i} \rightarrow_1 \{\mathfrak{i}_1, \mathfrak{i}_2\}$ and a join $\{\mathfrak{f}_1, \mathfrak{f}_2\} \rightarrow_1 \mathfrak{f}$. We put $\lambda(\mathfrak{i}) = 1$ and $\gamma(\mathfrak{f}) = 1$. All other initial and final weights are equal to 0. Every run G in $\mathcal{A}$ from $\mathfrak{i}$ to $\mathfrak{f}$ is of the form $G = G_1 \parallel_{\mathfrak{i},\mathfrak{f}} G_2$ where G_i is a run in $\mathcal{A}_i$ from $\mathfrak{i}_i$ to $\mathfrak{f}_i$ for $i = 1, 2$. By distributivity of $\diamond$ over $\oplus$, it is an easy exercise to show that $(\mathcal{S}_C(\mathcal{A}), t) = (S_1 \parallel S_2, t)$ for all $t \in \mathrm{SP}$. $\square$

Note. The above proof cannot be imitated for M-regular sp-series. Both in $\mathcal{A}_1$ and $\mathcal{A}_2$ there may be parallel non-atomic runs from $\mathfrak{i}_1$ to $\mathfrak{f}_1$ and from $\mathfrak{i}_2$ to $\mathfrak{f}_2$, respectively. Then the $\mathfrak{i}$-$\mathfrak{f}$-parallel product of these runs is a cascade branching run but no maximally branching run. Thus, these runs would not contribute to the M-behavior of $\mathcal{A}$, and, hence, the M-behavior of $\mathcal{A}$ is not $S_1 \parallel S_2$ anymore. We will come back to the closure of M-regular sp-series under parallel product in Chapter 7.

Again by analyzing the proof of Proposition 4.6 and considering Corollary 4.5 we get:

Corollary 4.7. *If S_1 and S_2 are C-recognized by some wbba $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, then $S_1 \parallel S_2$ is C-recognized by some wbba $\mathcal{A}$.*

To show the closure under parallel iteration we also have to use normalization of the wba involved.

Proposition 4.8. *If $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is C-regular, then $S^{\boxplus}$ is C-regular.*

Proof. Let $\mathcal{A} = (Q, \mu_{\mathrm{seq}}, \mu_{\mathrm{fork}}, \mu_{\mathrm{join}}, \lambda, \gamma)$ be a wba C-recognizing S. Due to Proposition 4.4 we assume $\mathcal{A}$ to be normalized. Let $\bar{\mathcal{A}}$ be a copy of $\mathcal{A}$. We define $\mathcal{A}'$ as follows:

- $Q' = Q \mathbin{\dot\cup} \bar{Q} \mathbin{\dot\cup} \{\hat{\mathfrak{i}}, \hat{\mathfrak{f}}\}$,

- $\mu'_{\text{seq}}(p,a,q) = \begin{cases} \mu_{\text{seq}}(p,a,q) & \text{if } p,q \in Q, \\ \bar{\mu}_{\text{seq}}(p,a,q) & \text{if } p,q \in \bar{Q}, \\ 0 & \text{otherwise,} \end{cases}$

- for all $m \in \{2, \dots, |Q'|\}$:

$$\begin{aligned} &\mu'^{m}_{\text{fork}}\big(p, \{p_1, \dots, p_m\}\big) \\ &= \begin{cases} \mu^m_{\text{fork}}\big(p, \{p_1, \dots, p_m\}\big) & \text{if } p, p_1, \dots, p_m \in Q, \\ \bar{\mu}^m_{\text{fork}}\big(p, \{p_1, \dots, p_m\}\big) & \text{if } p, p_1, \dots, p_m \in \bar{Q}, \\ 1 & \text{if } p = \hat{\mathfrak{i}}, m = 2 \text{ and} \\ & \{p_1, p_2\} = \{\hat{\mathfrak{i}}, \bar{\mathfrak{i}}\}, \\ 1 & \text{if } p = \hat{\mathfrak{i}}, m = 2 \text{ and} \\ & \{p_1, p_2\} = \{\mathfrak{i}, \bar{\mathfrak{i}}\}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

- for all $m \in \{2, \dots, |Q'|\}$:

$$\begin{aligned} &\mu'^{m}_{\text{join}}\big(\{p_1, \dots, p_m\}, p\big) \\ &= \begin{cases} \mu^m_{\text{join}}\big(\{p_1, \dots, p_m\}, p\big) & \text{if } p_1, \dots, p_m, p \in Q, \\ \bar{\mu}^m_{\text{join}}\big(\{p_1, \dots, p_m\}, p\big) & \text{if } p_1, \dots, p_m, p \in \bar{Q}, \\ 1 & \text{if } p = \hat{\mathfrak{f}}, m = 2 \text{ and} \\ & \{p_1, p_2\} = \{\hat{\mathfrak{f}}, \bar{\mathfrak{f}}\}, \\ 1 & \text{if } p = \hat{\mathfrak{f}}, m = 2 \text{ and} \\ & \{p_1, p_2\} = \{\mathfrak{f}, \bar{\mathfrak{f}}\}, \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

- $\lambda'(p) = \begin{cases} \lambda(p) & \text{if } p \in Q, \\ 1 & \text{if } p = \hat{\mathfrak{i}}, \\ 0 & \text{otherwise,} \end{cases}$ and $\gamma'(p) = \begin{cases} \gamma(p) & \text{if } p \in Q, \\ 1 & \text{if } p = \hat{\mathfrak{f}}, \\ 0 & \text{otherwise.} \end{cases}$

Claim 1. Let $G' : p' \to q'$ be a run of $\mathcal{A}'$. Then one of the following cases holds true:

- G' is a run in $\mathcal{A}$,
- G' is a run in $\bar{\mathcal{A}}$, or
- G' is a run from $\hat{\mathfrak{i}}$ to $\hat{\mathfrak{f}}$.

Indeed, for G' atomic Claim 1 holds by the definition of μ'_{seq}. Next, let $G' = G'_1 \cdot \ldots \cdot G'_m$ allow a sequential decomposition for $m \geq 2$. Assume $G'_1 : p'_0 \to p'_1$ with $p'_0 \in Q$. By induction G'_1 is a run in $\mathcal{A}$, and, especially, $p'_1 \in Q$ which is the starting state of G'_2. Repeating the argument along the sequence $G'_1, G'_2, \ldots, G'_m$ then G' is a run in $\mathcal{A}$. For $p'_0 \in \bar{Q}$ we get similarly that G' is a run in $\bar{\mathcal{A}}$. Now assume $p'_0 = \hat{\mathfrak{i}}$. By induction $p'_1 = \hat{\mathfrak{f}}$. Thus, G'_2 has to start in state $\hat{\mathfrak{f}}$. But there is neither a sequential nor a fork transition of $\mathcal{A}'$ starting in $\hat{\mathfrak{f}}$. Hence, $p'_0 = \hat{\mathfrak{i}}$ yields a contradiction. So Claim 1 is true for $G' = G'_1 \cdot \ldots \cdot G'_m$. Finally, we assume $G = \|_{p',q'}(G'_1, \ldots, G'_n)$ with $p', q' \in Q'$ and $n \geq 2$. Let $G'_i : p'_i \to q'_i$ for $i = 1, \ldots, n$. If $p' \in Q$ then $p'_1, \ldots, p'_n \in Q$. By induction $G'_1, \ldots, G'_n$ are runs in $\mathcal{A}$. Hence, $q' \in Q$ and G' is a run in $\mathcal{A}$. Reasoning by analogy $p' \in \bar{Q}$ implies $q' \in \bar{Q}$ and, hence, G' is a run in $\bar{\mathcal{A}}$. For $p = \hat{\mathfrak{i}}$, G' starts either with the fork transition $\hat{\mathfrak{i}} \to \{\hat{\mathfrak{i}}, \bar{\mathfrak{i}}\}$ or with $\hat{\mathfrak{i}} \to \{\mathfrak{i}, \hat{\mathfrak{i}}\}$. In the first case let $G_1 : \hat{\mathfrak{i}} \to q'_1$ and $G_2 : \bar{\mathfrak{i}} \to q'_2$. By induction $q'_1 = \hat{\mathfrak{f}}$ and G_2 is a run in $\bar{\mathcal{A}}$. Since G' is a run, we get $q'_2 = \bar{\mathfrak{f}}$ and $q = \hat{\mathfrak{f}}$. For the second case, $q = \hat{\mathfrak{f}}$ follows similarly. This proves Claim 1.

Now we consider runs G' in $\mathcal{A}'$ from an initial to a final state. By Claim 1 either $G' : \mathfrak{i} \to \mathfrak{f}$ being a run in $\mathcal{A}$ or $G' : \hat{\mathfrak{i}} \to \hat{\mathfrak{f}}$.

Claim 2. Let $G' : \hat{\mathfrak{i}} \to \hat{\mathfrak{f}}$ be a run in $\mathcal{A}'$. Then there is some $n \geq 2$ and runs $G_1, \ldots, G_{n-1}$ in $\bar{\mathcal{A}}$, and a run G_n in $\mathcal{A}$ such that for

$$\begin{aligned} H_1 &= \|_{\hat{\mathfrak{i}},\hat{\mathfrak{f}}}(G_{n-1}, G_n) \text{ and} \\ H_i &= \|_{\hat{\mathfrak{i}},\hat{\mathfrak{f}}}(G_{n-i}, H_{i-1}) \text{ for } i = 2, \ldots, n-1 \end{aligned}$$

we get $G' = H_{n-1}$.

We abbreviate the iterated product H_{n-1} of Claim 2 by $\Box(G_1, \ldots, G_n)$. To prove Claim 2 we distinguish what transition G' is starting with. For G' starting with $\hat{\mathfrak{i}} \to \{\mathfrak{i}, \bar{\mathfrak{i}}\}$ we get by Claim 1 and by definition of μ'_{join}

that $G' = \|_{\hat{\mathfrak{i}},\hat{\mathfrak{f}}}(G_1, G_2)$ with $G_1 : \mathfrak{i} \to \mathfrak{f}$ a run in $\mathcal{A}$ and $G_2 : \bar{\mathfrak{i}} \to \bar{\mathfrak{f}}$ a run in $\bar{\mathcal{A}}$. Hence, G' is of the desired form with $n = 2$. Now suppose G' starts with $\hat{\mathfrak{i}} \to \{\hat{\mathfrak{i}}, \bar{\mathfrak{i}}\}$. Again by Claim 1 and definition of μ'_{join} we get: $G' = \|_{\hat{\mathfrak{i}},\hat{\mathfrak{f}}}(G_1, H)$ with $G_1 : \bar{\mathfrak{i}} \to \bar{\mathfrak{f}}$ a run in $\bar{\mathcal{A}}$ and H a run from $\hat{\mathfrak{i}}$ to $\hat{\mathfrak{f}}$. By induction there is an integer $m \geq 2$ and runs $H_1, \ldots, H_{m-1}$ in $\bar{\mathcal{A}}$ and a run H_m in $\mathcal{A}$ such that $H = \Box(H_1, \ldots, H_m)$. This implies $G' = \Box(G_1, H_1, \ldots, H_m)$ which proves Claim 2.

Let $G' = \Box(G_1, \ldots, G_n)$ for some $n \geq 2$. Then we have:

$$\mathbf{lab}(G') = \mathbf{lab}(G_1) \parallel \ldots \parallel \mathbf{lab}(G_n) \, .$$

Since the weight of the fork and join transitions with $\hat{\mathfrak{i}}$ or $\hat{\mathfrak{f}}$ involved is 1, we also get:

$$\mathbf{wgt}(G') = \mathbf{wgt}(G_1) \Diamond \ldots \Diamond \mathbf{wgt}(G_n) \, . \tag{$\star$}$$

Now we have for every $t \in \mathrm{SP}$:

$$(\mathcal{S}_C(\mathcal{A}'), t)$$

$$= \bigoplus_{p',q' \in Q'} \bigoplus_{G' : p' \xrightarrow{t} q'} \lambda'(p') \circ \mathbf{wgt}(G') \circ \gamma'(q')$$

$$= \left[\bigoplus_{G' : \mathfrak{i} \xrightarrow{t} \mathfrak{f}} \mathbf{wgt}(G') \right] \oplus \left[\bigoplus_{G' : \hat{\mathfrak{i}} \xrightarrow{t} \hat{\mathfrak{f}}} \mathbf{wgt}(G') \right]$$

(by definition of λ' and γ')

$$= (\mathcal{S}_C(\mathcal{A}), t) \oplus \left[\bigoplus_{2 \leq n \leq \mathrm{wd}(t)} \bigoplus_{\substack{(t_1, \ldots, t_n): \\ t = t_1 \| \ldots \| t_n}} \bigoplus_{\substack{G' = \Box(G_1, \ldots, G_n) \\ \mathbf{lab}(G_i) = t_i}} \mathbf{wgt}(G') \right]$$

(by Claim 1, normalization of $\mathcal{A}$ and Claim 2)

$$=(\mathcal{S}_C(\mathcal{A}),t) \oplus \left[\bigoplus_{2\leq n\leq \mathrm{wd}(t)} \bigoplus_{\substack{(t_1,\ldots,t_n):\\ t=t_1\,\|\,\ldots\,\|\,t_n}} \bigoplus_{\substack{G'=\Box(G_1,\ldots,G_n)\\ \mathbf{lab}(G_i)=t_i}} \Big(\mathbf{wgt}(G_1) \diamond \ldots \diamond \mathbf{wgt}(G_n) \Big) \right]$$

(by Equation ($\star$))

$$=(\mathcal{S}_C(\mathcal{A}),t) \oplus \left[\bigoplus_{2\leq n\leq \mathrm{wd}(t)} \bigoplus_{\substack{(t_1,\ldots,t_n):\\ t=t_1\,\|\,\ldots\,\|\,t_n}} \Big((\mathcal{S}_C(\mathcal{A}),t_1) \diamond \ldots \diamond (\mathcal{S}_C(\mathcal{A}),t_n) \Big) \right]$$

(by distributivity of $\diamond$ over $\oplus$ and commutativity of $\oplus$)

$$= \bigoplus_{1\leq n\leq \mathrm{wd}(t)} \bigoplus_{\substack{(t_1,\ldots,t_n):\\ t=t_1\,\|\,\ldots\,\|\,t_n}} \Big[(S,t_1) \diamond \ldots \diamond (S,t_n) \Big]$$

(since $\mathcal{A}$ C-recognizes S)

$$=(S^{\boxplus},t).$$

Hence, $\mathcal{A}'$ C-recognizes $S^{\boxplus}$. $\Box$

Analyzing the proof given above we get:

Corollary 4.9. *If S is C-recognized by some wbba then $S^{\boxplus}$ is also C-recognized by some wbba.*

Note. As for the parallel product the proof of Proposition 4.8 cannot be adopted to M-regular sp-series because we introduced cascade branching runs to realize the parallel iteration.

Moreover, the M-regular sp-series are not closed under parallel iteration as the following example shows.

Example 4.10. We work in the setting of the Boolean bisemiring, *i.e.* in the case of sp-languages. Let $a \in \Sigma$. Obviously, $L = \{a\}$ is M-regular. Then

$$L^{\boxplus} = \{\underbrace{a \parallel \ldots \parallel a}_{n} \mid n \geq 1\}.$$

Assumed $L^{\boxplus}$ is M-recognizable by some wba $\mathcal{A}$ with finite-state set Q, there has to be an M-run G_n for every $n \geq 1$ from an initial state $\mathfrak{i}$ to a final state $\mathfrak{f}$ on

$$\underbrace{a \parallel \ldots \parallel a}_{n}.$$

Since G_n is an M-run it is of the form $G_n = \parallel_{\mathfrak{i},\mathfrak{f}}(H_1, \ldots, H_n)$ with $\mathbf{lab}(H_i) = a$ for $i = 1, \ldots, n$. Hence, the opening fork and the closing join transition of G_n have arity n. But by definition of a wba the arity of fork and join transitions is bounded by $|Q|$. Hence, $L^{\boxplus}$ is not M-regular.[1]

4.4 Closure under Sequential Product and Iteration

Now we turn to the closure of the class of regular sp-series under sequential product and sequential iteration. There are well-known counterparts of these closure properties in the theory of non-deterministic finite automata. Therefore, it is tempting to believe that constructions familiar from that 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 sequential product does not yield the correct result.

[1] Lodaya and Weil [LW00] allow in their definition of branching automata fork and join transitions of arbitrary arity. As already noted we have to work with sets instead of multisets when branching. Thus, the arity of our wba is automatically bounded by $|Q|$. But assumed a finite set of transitions, also the arity of fork and join transitions in the branching automata of Lodaya and Weil has to be uniformly bounded. Hence, a similar argument as ours would apply in Example 4.10.

We call a run *successful* if it is a run from an initial to a final state with weight distinct from 0.

Example 4.11. We work with the Boolean bisemiring $\mathbb{B}$, *i.e.* in the setting of sp-languages. Suppose that the wba $\mathcal{A}_1$ consists of the following transitions: a fork transition $\mathfrak{i}_1 \rightarrow_1 \{p_1, p_2\}$, and sequential transitions $p_1 \xrightarrow{a}_1 \mathfrak{f}_1$ and $p_2 \xrightarrow{a}_1 \mathfrak{f}_1$. Further, $\mathfrak{i}_1$ is the unique initial state and $\mathfrak{f}_1$ the unique final state. The wba $\mathcal{A}_2$ consists of the following transitions: a join transition $\{q_1, q_2\} \rightarrow_1 \mathfrak{f}_2$, and sequential transitions $\mathfrak{i}_2 \xrightarrow{a}_1 q_1$ and $\mathfrak{i}_2 \xrightarrow{a}_1 q_2$. The unique initial state is $\mathfrak{i}_2$ while $\mathfrak{f}_2$ is the only accepting state. Then the classical construction suggests to consider the wba consisting of all the transitions mentioned so far and, in addition, in particular sequential transitions $p_1 \xrightarrow{a}_1 \mathfrak{i}_2$ and $p_2 \xrightarrow{a}_1 \mathfrak{i}_2$. Figure 4.1 gives one successful run of the resulting wba, its label is $(aa) \parallel (aa)$. Since the language of both $\mathcal{A}_1$ and $\mathcal{A}_2$ is empty, the composition should not allow any successful run whatsoever.

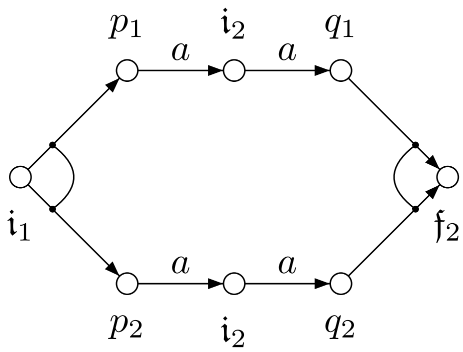


Figure 4.1: A problematic run in the classical product construction.

The problem in the example above is that the newly constructed automaton can switch in parallel sub-runs independently from $\mathcal{A}_1$ into $\mathcal{A}_2$. 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 run. In fork transitions, this signal is propagated along one branch only. In order not to duplicate runs, the signal is sent to the "smallest" of the states that arise from the fork transition.[2] Further, the newly constructed wba can only switch from $\mathcal{A}_1$ into $\mathcal{A}_2$ in the presence of this signal, and in any successful run, the signal has to be present at the final state.

Proposition 4.12. *Let $S_1, S_2 \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be two C-regular (M-regular) sp-series. Then $S_1 \cdot S_2$ is C-regular (M-regular), respectively.*

Proof. Let $\mathcal{A}_i = (Q_i, \mu_{i_{\mathrm{seq}}}, \mu_{i_{\mathrm{fork}}}, \mu_{i_{\mathrm{join}}}, \lambda_i, \gamma_i)$ be wba with $\mathcal{S}(\mathcal{A}_i) = S_i$ for $i = 1, 2$. We fix an arbitrary linear order $\leq$ on the state set Q_1 of $\mathcal{A}_1$. For $P \subseteq Q$ we denote by $\min_{\leq} P$ the least state of P with respect to $\leq$. The construction of a wba $\mathcal{A}$ recognizing $S_1 \cdot S_2$ is done in two steps. Firstly, we construct an automaton $\mathcal{A}'$ with $\mathcal{S}(\mathcal{A}') = \mathcal{S}(\mathcal{A}_1)$ as follows:

- $Q' = Q_1 \times \{0, 1\}$,
- $\mu'_{\mathrm{seq}}((p, x), a, (q, y)) = \begin{cases} \mu_{1_{\mathrm{seq}}}(p, a, q) & \text{if } x = y, \\ 0 & \text{otherwise,} \end{cases}$
- for all $m \in \{2, \dots, |Q'|\}$:
$$\mu'^{m}_{\mathrm{fork}}\Big((p, x), \big\{(p_1, x_1), \dots, (p_m, x_m)\big\}\Big)$$
$$= \begin{cases} \mu_{1}{}^{m}_{\mathrm{fork}}(p, \{p_1, \dots, p_m\}) & \text{if } p_1 = \min_{\leq}\{p_1, \dots, p_m\}, \\ & x_1 = x, \text{ and } x_i = 0 \\ & \text{for } i = 2, \dots, m, \\ 0 & \text{otherwise,} \end{cases}$$
- for all $m \in \{2, \dots, |Q'|\}$:
$$\mu'^{m}_{\mathrm{join}}\Big(\big\{(q_1, x_1), \dots, (q_m, x_m)\big\}, (q, x)\Big)$$
$$= \begin{cases} \mu_{1}{}^{m}_{\mathrm{join}}(\{q_1, \dots, q_m\}, q) & \text{if } x = x_1 \text{ and } x_i = 0 \\ & \text{for } i = 2, \dots, m, \\ 0 & \text{otherwise,} \end{cases}$$

[2]This is actually the reason why we have to assume that these 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.

- $\lambda'(p,x) = \begin{cases} \lambda_1(p) & \text{if } x = 1, \\ 0 & \text{otherwise,} \end{cases} \qquad \gamma'(q,x) = \begin{cases} \gamma_1(q) & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases}$

In the sequel, we refer to the second component of a state of $\mathcal{A}'$ as a signal. It is either 0 or 1.

Claim 1. If $G' : (p,x) \xrightarrow{t} (q,y)$ is a run of $\mathcal{A}'$ then $x = y$. Moreover, if (r,z) is a state of G' then $z \leq x$.

We prove this claim by structural induction. If $G' : (p,x) \xrightarrow{t} (q,y)$ is atomic then $x = y$ by the definition of μ'_{seq}. For $G' = G'_1 \cdot \ldots \cdot G'_m$ being sequential with $G'_i : (p_i, x_i) \xrightarrow{t_i} (p_{i+1}, x_{i+1})$ $(i = 1, \ldots, m)$ we have $x_i = x_{i+1}$ for all $i = 1, \ldots, m$, and every signal of a state of G'_i is smaller than or equal to x_i by induction hypothesis. Hence, $x = y$ and Claim 1 is true for G'. Now let $G' = \|_{(p,x),(q,y)}(G'_1, \ldots, G'_n)$ be a parallel run with $G'_i : (p_i, x_i) \xrightarrow{t_i} (q_i, y_i)$ for $i = 1, \ldots, n$. Let $p_1 = \min_{\leq}\{p_1, \ldots, p_n\}$. Then $x_1 = x$ and $x_i = 0$ for all $i = 2, \ldots, n$ by definition of μ'_{fork}. Hence, $y_1 = x_1$ and $y_i = 0$ for all $i = 2, \ldots, n$ by induction hypothesis. Moreover, all signals of states occurring in $G'_1, \ldots, G'_n$ are smaller than or equal to x. Now by definition of μ'_{join} we get $y = y_1 = x_1 = x$. This proves Claim 1.

We define the following sets of runs:

- $\mathcal{R}_{p,q}(\mathcal{A}_1)$ is the set of runs from p to q in $\mathcal{A}_1$,
- $\mathcal{R}^0_{p,q}(\mathcal{A}')$ and $\mathcal{R}^1_{p,q}(\mathcal{A}')$ are the sets of runs in $\mathcal{A}'$ from $(p,0)$ to $(q,0)$ and from $(p,1)$ to $(q,1)$, respectively.

We define $g_0 : \mathcal{R}^0_{p,q}(\mathcal{A}') \to \mathcal{R}_{p,q}(\mathcal{A}_1)$ by dropping the signals of the states of a run $G' \in \mathcal{R}^0_{p,q}(\mathcal{A}')$. Due to Claim 1 all states of such a run G' have signal 0. Thus, g_0 is a bijective mapping preserving labels and weights. Next, we define $g_1 : \mathcal{R}^1_{p,q}(\mathcal{A}') \to \mathcal{R}_{p,q}(\mathcal{A}_1)$ also by dropping the signals of the states. Note that runs from $\mathcal{R}^1_{p,q}(\mathcal{A}')$ can well contain states with signal 0. This is caused by possible branching. Thus, in order to show that g_1 is bijective and preserves labels and weights, one uses the corresponding result on g_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 4.4 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 wba $\mathcal{A}$ 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 $(\mathfrak{f}_1, 1)$ of $\mathcal{A}'$ by the unique initial state $\mathfrak{i}_2$ of $\mathcal{A}_2$, and call it $\mathfrak{s}$. We copy transitions and their weights from $\mathcal{A}'$ and $\mathcal{A}_2$ to $\mathcal{A}$ as far as the state $\mathfrak{s}$ is not involved. All transitions ending in $(\mathfrak{f}_1, 1)$ turn to transitions with same weight ending in $\mathfrak{s}$, and dually for the transitions starting in $\mathfrak{i}_2$. We put $\lambda(\mathfrak{s}) = \gamma(\mathfrak{s}) = 0$. The initial state of $\mathcal{A}$ is the unique initial state $(\mathfrak{i}_1, 1)$ of $\mathcal{A}'$, the final state of $\mathcal{A}$ is the state $\mathfrak{f}_2$ of $\mathcal{A}_2$ where initial and final weights carry over, respectively.

Claim 2. Let $p' \in Q'$ and $q \in Q_2$ with $p', q \neq \mathfrak{s}$. Then $G : p' \to q$ is a run of $\mathcal{A}$ from p' to q iff $G = G' \cdot G''$ with $G' : p' \to \mathfrak{s}$ a run in $\mathcal{A}'$ only and $G'' : \mathfrak{s} \to q$ a run in $\mathcal{A}_2$ only.

The "if-part" of Claim 2 is trivial. Vice versa, let $G : p' \to q$ be a run in $\mathcal{A}$ with $p' \in Q', q \in Q_2$, and $p', q \neq \mathfrak{s}$. G cannot be atomic because otherwise $p = \mathfrak{s}$ or $q = \mathfrak{s}$. If $G = G_1 \cdot \ldots \cdot G_m$ allows a sequential decomposition then there is an $i \in \{1, \ldots, m\}$ such that $G_1 \cdot \ldots \cdot G_i$ is a run in $\mathcal{A}'$ from p' to $(\mathfrak{f}_1, 1) = \mathfrak{s}$ and $G_{i+1} \cdot \ldots \cdot G_m$ is a run from $\mathfrak{s} = \mathfrak{i}_2$ to q in $\mathcal{A}_2$. Otherwise, there would have to be either a run from some $r_2 \in Q_2$ to some $r' \in Q'$ contradicting the definition of $\mathcal{A}$. Hence, there would be $\tilde{p} \in Q'$ and $\tilde{q} \in Q_2$ and a $j \in \{1, \ldots, m\}$ such that $G_j = \|_{\tilde{p},\tilde{q}}(H_1, \ldots, H_n)$ for some runs $H_1, \ldots, H_n$ 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 $\tilde{q}_i \in Q_2$ for $i = 1, \ldots, n$. By induction there would be a factorization $H_i = H_i' \cdot H_i''$ with $H_i' : \tilde{p}_i \to \mathfrak{s}$ in $\mathcal{A}'$ and $H_i'' : \mathfrak{s} \to \tilde{q}_i$ in $\mathcal{A}_2$ for $i = 1, \ldots, n$. Assume $p_1 = \min_{\leq}\{p_1, \ldots, p_n\}$. Then $x_i = 0$ for $i = 2, \ldots, n$. But this contradicts Claim 1 because $H_i' : (p_i, x_i) \to (\mathfrak{f}_1, 1)$ has to be a run for $i = 2, \ldots, n$.

Assumed the run G allows for some $n \geq 2$ a parallel decomposition $G = \|_{p',q}(G_1, \ldots, G_n)$, again by induction there would be a $j \in \{1, \ldots, n\}$ such that G_j starts in some $(p_j, 0) \in Q'$, ends in $q_j \in Q_2$, and crosses $\mathfrak{s}$, *i.e.* $G_j = G_j' \cdot G_j''$ with $G_j' : (p_j, 0) \to \mathfrak{s}$. This implies the existence of a run from $(p_j, 0)$ to $(f_1, 1)$ in $\mathcal{A}'$ contradicting Claim 1. Thus, G cannot allow a parallel decomposition. Hence, every run G from p' to q decomposes sequentially in the manner affirmed in Claim 2.

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

$$\begin{aligned}
(\mathcal{S}(\mathcal{A}),t) &= \bigoplus_{G:(\mathfrak{i}_1,1)\xrightarrow{t}\mathfrak{f}_2} \mathbf{wgt}(G) \\
&= \bigoplus_{\substack{t=t_1\cdot t_2 \\ G_1:(\mathfrak{i}_1,1)\xrightarrow{t_1}\mathfrak{s} \\ G_2:\mathfrak{s}\xrightarrow{t_2}\mathfrak{f}_2}} \mathbf{wgt}(G_1)\circ\mathbf{wgt}(G_2) \\
&= \bigoplus_{t=t_1\cdot t_2}\Bigg(\bigoplus_{G_1:(\mathfrak{i}_1,1)\xrightarrow{t_1}\mathfrak{s}} \mathbf{wgt}(G_1)\Bigg)\circ\Bigg(\bigoplus_{G_2:\mathfrak{s}\xrightarrow{t_2}\mathfrak{f}_2} \mathbf{wgt}(G_2)\Bigg) \\
&= \bigoplus_{t=t_1\cdot t_2} (\mathcal{S}(\mathcal{A}_1),t_1)\circ(\mathcal{S}(\mathcal{A}_2),t_2) \\
&=(S_1\cdot S_2,t).
\end{aligned}$$

This concludes the proof. $\square$

Starting in the preceding proof with wbba instead of wba we get:

Corollary 4.13. *Let S_1, S_2 be two regular series recognized by wbba $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Then $S_1\cdot S_2$ is also recognized by a wbba.*

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

Example 4.14. We work with the Boolean bisemiring $\mathbb{B}$, *i.e.* in the setting of sp-languages. Consider the wba from Figure 4.2 (left) where we omitted the weights; any transition depicted has weight 1 and no further transitions have non-zero weight. The support of the recognized sp-series is $\{a \parallel b, dae\}$. The classical construction for the sequential iteration tells us to add, among other transitions, one of the form $q_1 \xrightarrow{e} \mathfrak{i}$ since there is a sequential transition $q_1 \xrightarrow{e} \mathfrak{f}$ in the wba in consideration. But then we get the run depicted in Figure 4.2 (right) whose label is $(aeda) \parallel b$ which does not belong to the sequential iteration of the sp-language generated by the wba we started with.

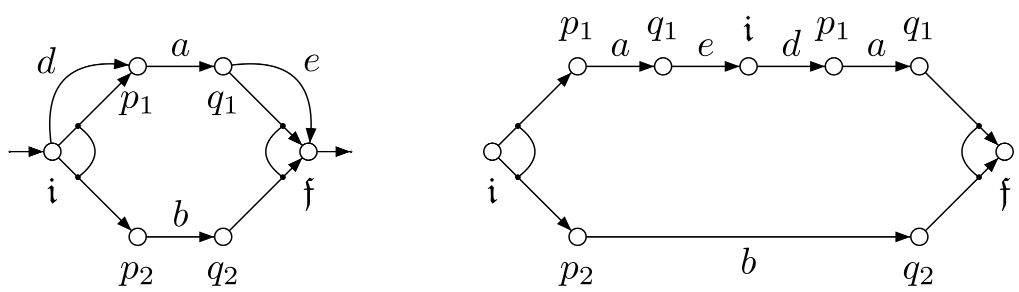


Figure 4.2: An unwanted run in the classical iteration construction.

Now it is not sufficient anymore to send just one signal as done in the last proof. There, we switched frome one wba into another one. Hence, it was sufficient to prevent $\mathcal{A}_1$ switching to $\mathcal{A}_2$ in at least one parallel sub-run. Now we have to ensure in any of the parallel sub-runs that the automaton does not switch from the parallel sub-run to the higher level the run has started at.

Lodaya and Weil's solution is, again, to use behaved automata[3]. 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 have to be present. As before, in any successful run, both signals are present in the first and the last state.

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

Proposition 4.15. *If $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is C-regular (M-regular), then the sequential iteration S^+ is C-regular (M-regular), respectively.*

[3] The notion of "behaved automata" is determined by the restrictions forced by the construction for sequential iteration. As we have indicated these are stronger than those for the sequential product. The notion of "behavedness" originally defined in [LW00] was too weak. In [LW01] Lodaya and Weil used the stronger notion of "behavedness" working for sequential product and iteration.

Proof. Let $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ be a wba recognizing S. We assume an arbitrary but fixed linear order $\leq$ on Q. For $P \subseteq Q$ we denote by $\min_{\leq} P$ the least element and by $\max_{\leq} P$ the greatest element of P. Again we construct a wba recognizing S^+ in two steps. Firstly, we build an automaton $\mathcal{A}'$ with the same behavior as $\mathcal{A}$ similar to the construction in the last proof, but this time with two signals for the state:

- $Q' = Q \times \{0,1\}^2$,
- $\mu'_{\text{seq}}\Big((p,x,x'), a, (q,y,y')\Big) = \begin{cases} \mu_{\text{seq}}(p,a,q) & \text{if } x = y, x' = y', \\ 0 & \text{otherwise,} \end{cases}$
- for all $m \in \{2, \dots, |Q'|\}$:

$$\mu'^{m}_{\text{fork}}\Big((p,x,x'), \big\{(p_1,x_1,x'_1), \dots, (p_m,x_m,x'_m)\big\}\Big)$$

$$= \begin{cases} \mu^m_{\text{fork}}(p, \{p_1, \dots, p_m\}) & \text{if } p_1 = \min_{\leq}\{p_1, \dots, p_m\}, \\ & p_m = \max_{\leq}\{p_1, \dots, p_m\}, \\ & x_1 = x, x'_1 = 0, x_m = 0, \\ & x'_m = x', \text{ and } x_i = x'_i = 0 \\ & \text{for all } i = 2, \dots, m-1, \\ 0 & \text{otherwise,} \end{cases}$$

- for all $m \in \{2, \dots, |Q'|\}$:

$$\mu'^{m}_{\text{join}}\Big(\big\{(q_1,x_1,x'_1), \dots, (q_m,x_m,x'_m)\big\}, (q,x,x')\Big)$$

$$= \begin{cases} \mu^m_{\text{join}}(\{q_1, \dots, q_m\}, q) & \text{if } x_1 = x, x'_1 = 0, x_m = 0, \\ & x'_m = x', \text{ and } x_i = x'_i = 0 \\ & \text{for all } i = 2, \dots, m-1, \\ 0 & \text{otherwise,} \end{cases}$$

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

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

We refer to the second and third component of a state as the signals of this state. A run in $\mathcal{A}'$ is between states with same signals only:

Claim 1. If $G' : (p,x,x') \to (q,y,y')$ is a run of $\mathcal{A}'$ then $x = y$ and $x' = y'$. Moreover, if (r,z,z') is a state occurring in G' then $z \le x$ and $z' \le x'$.

Indeed, Claim 1 is obvious for atomic runs. Let $G' = G'_1 \cdot \ldots \cdot G'_m$ be the sequential decomposition of G' with $m \ge 2$. By induction Claim 1 holds true for $G'_1 : (p,x,x') \to (p_1,x_1,x'_1)$. Hence, $x_1 = x$, $x'_1 = x'$, and all signals of states of G'_1 are smaller or equal (x,x'). Repeating the argument along the sequence $G'_1,\ldots,G'_m$ we get Claim 1 for G'. Now let $G' = \|_{p',q'}(G'_1,\ldots,G'_n)$ with $p' = (p,x,x')$, $q' = (q,y,y')$, and $n \ge 2$. Further let $G'_i : (p_i,x_i,x'_i) \to (q_i,y_i,y'_i)$ for $i = 1,\ldots,n$. We suppose $p_1 = \min_{\le}\{p_1,\ldots,p_n\}$ and $p_n = \max_{\le}\{p_1,\ldots,p_n\}$. By definition of μ'_{fork} we have $x = x_1$, $x' = x'_n$, $x_i = 0$ for $i = 2,\ldots,n$, and $x'_j = 0$ for $j = 1,\ldots,n-1$. Hence, $x_i \le x$ and $x'_i \le x'$ for $i = 1,\ldots,n$. By induction, $y_1 = x_1 = x$, $y'_n = x'_n = x'$, $y_i = 0$ for $i = 2,\ldots,n$, and $y'_j = 0$ for $j = 1,\ldots,n-1$. By definition of μ'_{join} this implies $y = y_1 = x$ and $y' = y'_n = x'$. This proves Claim 1.

By Claim 1 one shows $\mathcal{S}(\mathcal{A}') = \mathcal{S}(\mathcal{A})$ along the same lines as in the proof of Proposition 4.12.

Due to Proposition 4.4 $\mathcal{A}$ can be assumed to be normalized. Then $\mathcal{A}'$ is normalized too. Let $\mathfrak{i}$ and $\mathfrak{f}$ denote the unique initial and final state of $\mathcal{A}$, respectively. Then $\mathfrak{i}' = (\mathfrak{i},1,1)$ and $\mathfrak{f}' = (\mathfrak{f},1,1)$ are the unique initial and final state of $\mathcal{A}'$, respectively. Now we construct from $\mathcal{A}'$ a wba $\mathcal{A}^+$ 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 $p' \xrightarrow{a}_k \mathfrak{f}'$ of $\mathcal{A}'$ a transition $p' \xrightarrow{a}_k \mathfrak{i}'$ in $\mathcal{A}^+$, and for every join transition $\{q'_1,\ldots,q'_m\} \to_l \mathfrak{f}'$ of $\mathcal{A}'$ a further join $\{q'_1,\ldots,q'_m\} \to_l \mathfrak{i}'$ in $\mathcal{A}^+$. The initial and final weights remain the same. Therefore, $\mathcal{A}^+$ has still the unique initial state $\mathfrak{i}'$ and the unique final state $\mathfrak{f}'$.

Now, let $\mathcal{R}_{\mathfrak{i}',\mathfrak{i}'}(\mathcal{A}^+)$ be the set of all runs in $\mathcal{A}^+$ from $\mathfrak{i}'$ to $\mathfrak{i}'$ such that $\mathfrak{i}'$ does not appear in between. By $\mathcal{R}_{\mathfrak{i}',\mathfrak{f}'}(\mathcal{A}')$ we denote the set of all runs in $\mathcal{A}'$ going from $\mathfrak{i}'$ to $\mathfrak{f}'$. The mapping $g : \mathcal{R}_{\mathfrak{i}',\mathfrak{i}'}(\mathcal{A}^+) \to \mathcal{R}_{\mathfrak{i}',\mathfrak{f}'}(\mathcal{A}')$ is defined as follows. It maps $G \in \mathcal{R}_{\mathfrak{i}',\mathfrak{i}'}(\mathcal{A}^+)$ to $G' \in \mathcal{R}_{\mathfrak{i}',\mathfrak{f}'}(\mathcal{A}')$ by labeling the sink of G with $\mathfrak{f}'$ instead of $\mathfrak{i}'$. We show that g is well defined. Indeed, since $\mathfrak{i}'$ does appear in G as the label of the source and the sink only all but the last transition of G are also transitions in $\mathcal{A}'$. The last transition is either of the form $p' \xrightarrow{a}_k \mathfrak{i}'$ or is a join $\{q'_1, \ldots, q'_m\} \rightarrow_l \mathfrak{i}'$ for some states $p', q'_1, \ldots, q'_m$ and $k, l \in \mathbb{K}$. But then $p' \xrightarrow{a}_k \mathfrak{f}'$ respectively $\{q'_1, \ldots, q'_m\} \rightarrow_l \mathfrak{f}'$ are transitions of $\mathcal{A}'$ by definition of $\mathcal{A}^+$. Hence, $g(G) = G'$ is well defined.

Now, by definition of $\mathcal{A}^+$ and g it follows immediately that g is bijective and preserves labels and weights.

Claim 2. $G : \mathfrak{i}' \rightarrow \mathfrak{f}'$ is a run in $\mathcal{A}^+$ iff there is some $m \geq 1$ and $G_j \in \mathcal{R}_{\mathfrak{i}',\mathfrak{i}'}(\mathcal{A}^+)$ for $j = 1, \ldots, m-1$ and $G_m \in \mathcal{R}_{\mathfrak{i}',\mathfrak{f}'}(\mathcal{A}')$ such that $G = G_1 \cdot \ldots \cdot G_m$.

Let $G : \mathfrak{i}' \rightarrow \mathfrak{f}'$ be a run in $\mathcal{A}^+$. Either state $\mathfrak{i}'$ appears only once and then G is also a run in $\mathcal{A}'$. Then G is of the desired form of Claim 2 with $m = 1$. Otherwise, $\mathfrak{i}'$ appears more than once in G. Let $G = G_1 \cdot \ldots \cdot G_m$ ($m \geq 1$) be the sequential decomposition of G. Assumed one G_i is not atomic, *i.e.* $G_i = \|_{p',q'}(H_1, \ldots, H_n)$ for some $n \geq 2$ starting with fork transition $p' \rightarrow \{p'_1, \ldots, p'_n\}$. By definition of ${\mu^+}_{\text{fork}}$ none of the p'_j carries signal $(1,1)$ for $j = 1, \ldots, n$. It is obvious that Claim 1 holds true also for $\mathcal{A}^+$. Thus, we get that none of the states in $H_1, \ldots, H_n$ carries signal $(1,1)$. Especially, $\mathfrak{i}'$ cannot occur in $H_1, \ldots, H_n$. Hence, $\mathfrak{i}'$ appears at the upper level of G only. But then, obviously, G allows the decomposition given by Claim 2. Vice versa, every run of this form is clearly a run from $\mathfrak{i}'$ to $\mathfrak{f}'$ in $\mathcal{A}^+$.

Now we get for any sp-poset $t \in \mathrm{SP}$:

$$
\begin{aligned}
(\mathcal{S}(\mathcal{A}^+), t) &= \bigoplus_{G:\mathfrak{i}' \xrightarrow{t} \mathfrak{f}'} \mathbf{wgt}(G) \\
&= \bigoplus_{\substack{G = G_1 \cdot \ldots \cdot G_m: \\ \mathfrak{i}' \xrightarrow{t_1} \mathfrak{i}' \xrightarrow{t_2} \ldots \xrightarrow{t_m} \mathfrak{f}'}} \mathbf{wgt}(G_1) \circ \ldots \circ \mathbf{wgt}(G_m) \\
&= \bigoplus_{m \geq 1} \bigoplus_{t = t_1 \cdot \ldots \cdot t_m} \bigoplus_{\substack{G_1 : \mathfrak{i}' \xrightarrow{t_1} \mathfrak{i}' \\ \ldots \\ G_m : \mathfrak{i}' \xrightarrow{t_m} \mathfrak{f}'}} \mathbf{wgt}(G_1) \circ \ldots \circ \mathbf{wgt}(G_m)
\end{aligned}
$$

$$
\begin{aligned}
&= \bigoplus_{m\geq 1} \bigoplus_{t=t_1\cdot\ldots\cdot t_m} \bigoplus_{\substack{G_1:\mathfrak{i}' \xrightarrow{t_1} \mathfrak{i}' \\ \cdots \\ G_m:\mathfrak{i}' \xrightarrow{t_m} \mathfrak{f}'}} \mathbf{wgt}(g(G_1)) \circ \ldots \\
&\qquad\qquad\qquad\qquad \ldots \circ \mathbf{wgt}(g(G_{m-1})) \circ \mathbf{wgt}(G_m) \\
&= \bigoplus_{m\geq 1} \bigoplus_{t=t_1\cdot\ldots\cdot t_m} (\mathcal{S}(\mathcal{A}'), t_1) \circ \ldots \circ (\mathcal{S}(\mathcal{A}'), t_m) \\
&= (S^+, t).
\end{aligned}
$$

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

Again the above construction for sequential iteration turns a wbba into a wbba.

Corollary 4.16. *Let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be recognized by a wbba. Then S^+ is recognized by a wbba.*

4.5 From Rationality to Regularity

Now we can state the main theorem of this chapter.

Theorem 4.17. *Let $\mathbb{K}$ be an arbitrary bisemiring. Every rational sp-series $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is C-regular. Moreover, any rational sp-series S can be C-recognized by a normalized wbba.*

Proof. All monomials are obviously C-regular because their support is a letter only. By Propositions 4.1, 4.2, 4.6, 4.8, 4.12, and 4.15 the C-regular sp-series are closed under sum, scalar products, parallel product and iteration, and sequential product and iteration. Hence, every rational sp-series is C-regular.

Clearly, all monomials are C-recognized by a wbba. By Corollaries 4.3, 4.7, 4.9, 4.13, and 4.16 sp-series C-recognized by a wbba are closed under all rational operations. Thus, every rational sp-series is C-recognized by a wbba. This wbba can be normalized due to Corollary 4.5. $\square$

Corollary 4.18. *Every sequential-rational sp-series* $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ *is C-regular.*

Remark 4.19. All constructions done in Propositions 4.1, 4.2, 4.4, 4.6, 4.8, 4.12, and 4.15 are effective. Given a rational expression for some rational sp-series S we are able to construct a wba (and even a wbba) $\mathcal{A}$ that recognizes S. Naturally, effectiveness is restricted to the assumption that the bisemiring $\mathbb{K}$ is given in an effective way. Especially, we have to calculate in $\mathbb{K}$ when constructing normalized wba and wba realizing closure under scalar products.

5 Bounded Width and Bounded Depth

In Chapter 6 we will show that regular sp-series of bounded width are sequential-rational. In order to construct a sequential-rational expression from a given wba we will need a hierarchy of the fork and join transitions of the automaton. This hierarchy can be enforced if the sp-series recognized by the wba is of bounded width. In this chapter we show how the bounded width property of a regular sp-series S can be reflected in the structure of a wba recognizing S. For this we introduce the notion of "bounded depth".

Let $\mathcal{A}$ be a wba over an alphabet Σ and a bisemiring $\mathbb{K}$. We define a *depth function* $\mathrm{dp} : \mathcal{R}(\mathcal{A}) \to \mathbb{N}$ as follows:

- Every atomic run G is of depth 0.
- If $G = G_1 \cdot \ldots \cdot G_m$ is the sequential decomposition of G, then $\mathrm{dp}(G) = \max\{\mathrm{dp}(G_i) \mid i = 1, \ldots, m\}$.
- If $G = \|_{p,q}(G_1, \ldots, G_n)$ for some $p, q \in Q$, then $\mathrm{dp}(G) = 1 + \max\{\mathrm{dp}(G_i) \mid i = 1, \ldots, n\}$.

Hence, the depth of a run measures the nesting of branchings within a run. A wba $\mathcal{A}$ is of *bounded C-depth* (*bounded M-depth*) if there is a $d \in \mathbb{N}$ with $\mathrm{dp}(G) \leq d$ for any C-run (M-run) G of $\mathcal{A}$, respectively.

Since every M-run is a C-run, any wba $\mathcal{A}$ of bounded C-depth is also of bounded M-depth. The following conclusions are valid both for cascade branching and maximally branching mode. Therefore, we speak again of "bounded depth", "behavior" and so on, and agree that the running mode used is fixed.

Proposition 5.1. *Let $\mathcal{A}$ be a wba of bounded depth. Then $\mathcal{S}(\mathcal{A})$ is of bounded width.*

Proof. Let $d \in \mathbb{N}$ with $\mathrm{dp}(G) \leq d$ for all $G \in \mathcal{R}(\mathcal{A})$. Moreover, let B denote the highest of all arities of fork and join transitions of $\mathcal{A}$. We get for $G = \|_{p,q}(G_1, \ldots, G_n)$ with $p, q \in Q$:

$$\mathrm{wd}(\mathbf{lab}(G)) \leq B \cdot \max\{\mathrm{wd}(\mathbf{lab}(G_i)) \mid i = 1, \ldots, n\}$$

where $\mathrm{dp}(G) > \mathrm{dp}(G_i)$ for $i = 1, \ldots, n$. If $G = G_1 \cdot \ldots \cdot G_m$ then $\mathrm{wd}(\mathbf{lab}(G)) = \max\{\mathrm{wd}(\mathbf{lab}(G_i)) \mid i = 1, \ldots, n\}$. Hence, for any $G \in \mathcal{R}(\mathcal{A})$:

$$\mathrm{wd}(\mathbf{lab}(G)) \leq B^d\ .$$

Thus, the support of $\mathcal{S}(\mathcal{A})$ is of bounded width with a bound of B^d. $\square$

Now, the converse implication remains to be shown: any regular sp-series of bounded width can be accepted by a wba of bounded depth. The corresponding statement for sp-languages was shown by Kuske [Kus03] by counting and thereby limiting the depth of a run. That proof can be extended to wba over bisemirings that do not allow an additive decomposition of 0. We include this construction even if it does not yield the desired result for arbitrary bisemirings. But we will make use of it later on in the proof of Lemma 7.3.

Construction 5.2. Let $\mathcal{A} = (Q, \mu_{\mathrm{seq}}, \mu_{\mathrm{fork}}, \mu_{\mathrm{join}}, \lambda, \gamma)$ be a wba over Σ and $\mathbb{K}$, and let $d \in \mathbb{N}$. Then $\mathcal{A}_{|d} = (Q', {\mu'}_{\mathrm{seq}}, {\mu'}_{\mathrm{fork}}, {\mu'}_{\mathrm{join}}, \lambda', \gamma')$ is defined as follows:

- $Q' = Q \times \{0, 1, \ldots, d\}$,
- ${\mu'}_{\mathrm{seq}}((p,x), a, (q,y)) = \begin{cases} \mu_{\mathrm{seq}}(p,a,q) & \text{if } x = y, \\ 0 & \text{otherwise,} \end{cases}$
- for all $m \in \{2, \ldots, |Q'|\}$:

$${\mu'}^m_{\mathrm{fork}}\left((p,x), \left\{(p_1,x_1), \ldots, (p_m,x_m)\right\}\right)$$

$$= \begin{cases} \mu^m_{\mathrm{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } x_1 = \cdots = x_m = x+1, \\ 0 & \text{otherwise,} \end{cases}$$

- for all $m \in \{2, \ldots, |Q'|\}$:

$$\mu'^{m}_{\text{join}}\left(\left\{(q_1, y_1), \ldots, (q_m, y_m)\right\}, (q, y)\right)$$

$$= \begin{cases} \mu^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) & \text{if } y_1 = \cdots = y_m = y + 1, \\ 0 & \text{otherwise,} \end{cases}$$

- $\lambda'(p, x) = \begin{cases} \lambda(p) & \text{if } x = 0, \\ 0 & \text{otherwise,} \end{cases}$ and $\gamma'(q, y) = \begin{cases} \gamma(q) & \text{if } y = 0, \\ 0 & \text{otherwise.} \end{cases}$

$\mathcal{A}_{|d}$ is called the *d-depth counter wba* of $\mathcal{A}$.

The states of $\mathcal{A}_{|d}$ have two components. $\mathcal{A}_{|d}$ simulates $\mathcal{A}$ in the first component and counts the nesting of fork and join transitions in the second one.

Proposition 5.3. *For any wba $\mathcal{A}$ and any $d \in \mathbb{N}$ the wba $\mathcal{A}_{|d}$ is of bounded depth.*

Proof. As any fork transition of $\mathcal{A}_{|d}$ increments the depth counter in the second component of the states and since the counter is bounded by d, any run of $\mathcal{A}_{|d}$ has depth at most d. Hence, $\mathcal{A}_{|d}$ is of bounded depth. □

A bisemiring $\mathbb{K}$ is called *zero-sum-free* if its addition is zero-sum-free, *i.e.* $k \oplus k' = 0$ implies $k = k' = 0$ for all $k, k' \in \mathbb{K}$. In particular, all bisemirings with an idempotent addition are zero-sum-free.

Lemma 5.4. *Let $\mathcal{A}$ be a wba over Σ and a zero-sum-free bisemiring $\mathbb{K}$. Moreover, let $\mathcal{S}(\mathcal{A})$ have bounded width. For*

$$d = \sup\{\mathrm{wd}(t) \mid t \in \operatorname{supp} \mathcal{S}(\mathcal{A})\} - 1$$

we get $\mathcal{S}(\mathcal{A}_{|d}) = \mathcal{S}(\mathcal{A})$.

Proof. Let $\mathcal{A}$ and $\mathcal{A}_{|d}$ be as in Construction 5.2. Let $p \in Q$ be initial, $q \in Q$ final, and let $t \in \mathrm{SP}$. Furthermore, let $\mathcal{R}_{(p,t,q)}(\mathcal{A}_{|d})$ be the set of all runs of $\mathcal{A}_{|d}$ going from state $(p, 0)$ to state $(q, 0)$ with label $t \in \mathrm{SP}$ and weight distinct from zero. Similarly, let $\mathcal{R}_{(p,t,q)}(\mathcal{A})$ denote the set of all runs of $\mathcal{A}$ from p to q with label t and weight distinct from zero. Then we define for all $t \in \mathrm{SP}$ a mapping $g_{(p,t,q)} : \mathcal{R}_{(p,t,q)}(\mathcal{A}_{|d}) \to \mathcal{R}_{(p,t,q)}(\mathcal{A})$ by deleting the second component of each state. It is clear from the definition that $g_{(p,t,q)}$ preserves the weight of a run. Moreover, $g_{(p,t,q)}$ is injective because the second component of a state in a run of $\mathcal{R}_{(p,t,q)}(\mathcal{A}_{|d})$ is uniquely determined by the first component and the branching structure of the run. Now let $G \in \mathcal{R}_{(p,t,q)}(\mathcal{A})$. By definition, $\mathbf{wgt}(G) \neq 0$. Since $\mathbb{K}$ is zero-sum-free, p is initial, and q is final, we have $t \in \operatorname{supp} \mathcal{S}(\mathcal{A})$. By induction on the structure of G and t, respectively, we get $\mathrm{dp}(G) < \mathrm{wd}(t)$. Since $\mathrm{wd}(t) \leq d+1$ we get $\mathrm{dp}(G) \leq d$. Hence, there is a run $G' \in \mathcal{R}_{(p,t,q)}(\mathcal{A}_{|d})$ with $g_{(p,t,q)}(G') = G$. Therefore, for all initial states $p \in Q$, all final states $q \in Q$, and all $t \in \mathrm{SP}$ the function $g_{(p,t,q)}$ is bijective and preserves weights . Note that (p, x) is initial in $\mathcal{A}_{|d}$ iff $x = 0$ and p is initial in $\mathcal{A}$. Similarly, (q, y) is final in $\mathcal{A}_{|d}$ iff $y = 0$ and q is final in $\mathcal{A}$. Furthermore, $\lambda'(p, 0) = \lambda(p)$ and $\gamma'(q, 0) = \gamma(q)$. Thus, we get for all $t \in \mathrm{SP}$:

$$
\begin{aligned}
(\mathcal{S}(\mathcal{A}_{|d}), t) &= \bigoplus_{p,q \in Q} \bigoplus_{G' : (p,0) \xrightarrow{t} (q,0)} \lambda'(p, 0) \circ \mathbf{wgt}(G') \circ \gamma'(q, 0) \\
&= \bigoplus_{\substack{p,q \in Q \\ p \text{ initial} \\ q \text{ final}}} \bigoplus_{g_{(p,t,q)}(G') : p \xrightarrow{t} q} \lambda(p) \circ \mathbf{wgt}(g(G')) \circ \gamma(q) \\
&= \bigoplus_{p,q \in Q} \bigoplus_{G : p \xrightarrow{t} q} \lambda(p) \circ \mathbf{wgt}(G) \circ \gamma(q) \\
&= (\mathcal{S}(\mathcal{A}), t) \; .
\end{aligned}
$$

□

Remark 5.5. Let $\mathcal{A}$ be a wba of bounded depth with depth bound d. Then $\mathcal{S}(\mathcal{A}_{|d}) = \mathcal{S}(\mathcal{A})$.

Remark 5.6. If $\mathcal{A}$ is a wbba and $d \in \mathbb{N}$, then $\mathcal{A}_{|d}$ is a wbba.

Lemma 5.4 shows that any regular width-bounded sp-series over a zero-sum-free bisemiring can be recognized by a wba of bounded depth. The following example makes clear that the depth counter construction does not yield the same result for arbitrary bisemirings.

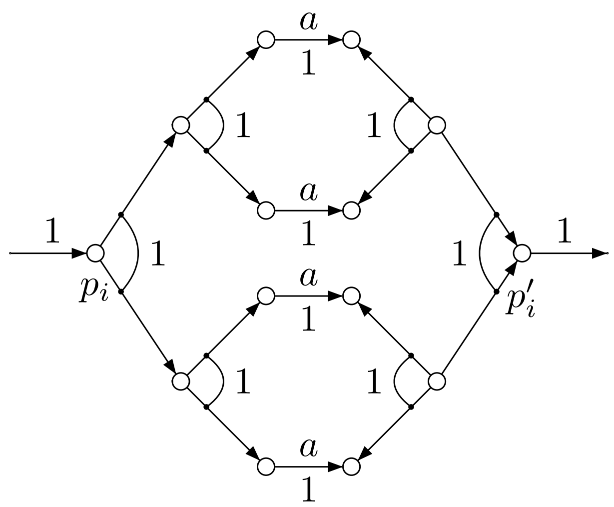


Figure 5.1: A run on $a \parallel a \parallel a \parallel a$ of depth 2 and weight 1.

Example 5.7. We consider the bisemiring $(\mathbb{Z}, +, \cdot, \cdot, 0, 1)$. Then there exists a wba $\mathcal{A}$ with $\mathrm{wd}\big(\mathrm{supp}\,\mathcal{S}(\mathcal{A})\big) = 3$ such that the only runs labeled by $t = a \parallel a \parallel a \parallel a$ are those depicted in Figures 5.1 and 5.2. The first of them has weight 1, the second -1, hence their weights sum up to 0. Further, the depth of the first run is 2 while the depth of the second one is 3. Applying the depth counter construction with $d = 2$ as in Lemma 5.4 would disallow the second run only. Hence, it results in a wba $\mathcal{A}_{|d}$ with $(\mathcal{S}(\mathcal{A}_{|d}), t) = 1 \neq (\mathcal{S}(\mathcal{A}), t)$. Since this problem does not arise if we chose $d = 3$, it is tempting to use Construction 5.2 with some $d \geq \mathrm{wd}(\mathrm{supp}(\mathcal{S}(\mathcal{A})))$.

We only sketch the idea why this cannot work either. Let $\mathcal{A}$ be a wba that has (among others) states p_i, p'_i, q_i, q'_i for $0 \leq i \leq 2$ (the remaining states correspond to the unlabeled nodes in Figures 5.1 and 5.2). Apart from the non-zero transitions in the figures, $\mathcal{A}$ can fork from p_i into $\{p_{(i+1) \bmod 3}, p_{(i+2) \bmod 3}\}$ and from q_i into $\{q_{(i+1) \bmod 3}, q_{(i+2) \bmod 3}\}$ at weight 1. Dually, it can join the states $p'_{(i+1) \bmod 3}$ and $p'_{(i+2) \bmod 3}$ into p'_i and the states $q'_{(i+1) \bmod 3}$ and $q'_{(i+2) \bmod 3}$ into q'_i at weight 1. We set

$$\lambda(r) = \begin{cases} 1 & \text{if } r = p_i, \\ -1 & \text{if } r = q_i, \\ 0 & \text{otherwise,} \end{cases}$$

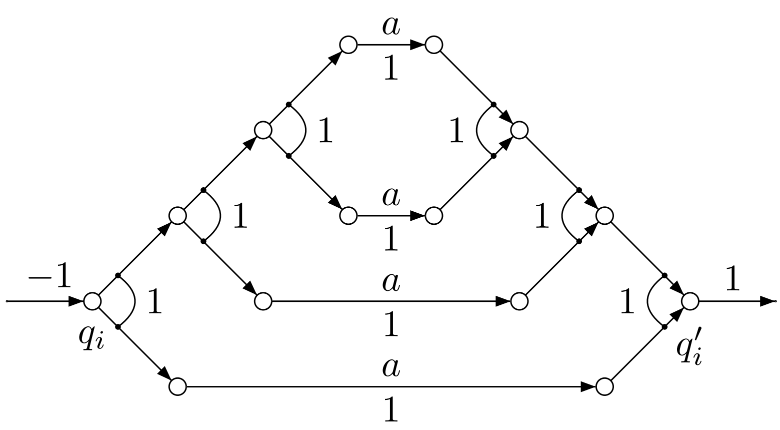


Figure 5.2: A run on $a \,\|\, a \,\|\, a \,\|\, a$ of depth 3 and weight -1.

$$\gamma(r) = \begin{cases} 1 & \text{if } r = p_i' \text{ or } r = q_i', \\ 0 & \text{otherwise.} \end{cases}$$

To compute the weight of an sp- poset t in $\mathcal{S}(\mathcal{A})$, it suffices to consider runs that start in p_i or q_i. A run starting in p_i will firstly fork into several copies of the states p_0, p_1, and p_2. Then it will perform the run from Figure 5.1 before it joins the states p_0', p_1', and p_2' in a cascade into state p_i'. The weight of this run will be 1. Such a run is only possible if t is an anti-chain of $4k$ a-labeled nodes (for some $k \in \mathbb{N}$) which we denote by a_k. Then the run has depth at most $k+1$. Similarly, a run starting in q_i will first fork into several copies of the states q_0, q_1, and q_2. Then it will perform the run from Figure 5.2 before it joins the states q_0', q_1', and q_2' in a cascade into state q_i'. The weight of this run will be -1. Again, it is only possible if $t = a_k$. But now, the run can have depth $k+2$. Since there is a bijection between the runs starting in p_i and those starting in q_i, we get $(\mathcal{S}(\mathcal{A}), t) = 0$ for all $t \in \mathrm{SP}$. Especially, $\mathcal{S}(\mathcal{A})$ is of bounded width.

Now perform the depth counter construction with $d = k+1$. Then the run on a_k starting in q_0 with depth $k+2$ is missing, but its "partner" starting in p_0 with depth $k+1$ is still present. This implies $(\mathcal{S}(\mathcal{A}_{|d}), a_k) > 0$. Hence, there is no $d \in \mathbb{N}$ such that the d-depth counter wba of $\mathcal{A}$ has the same behavior as $\mathcal{A}$.

For a wba $\mathcal{A}$ with width-bounded behavior, Example 5.7 has shown that it does not suffice to restrict the depth of runs in $\mathcal{A}$ uniformly to get a wba of bounded depth with the same behavior as $\mathcal{A}$. To overcome this problem, we will keep track of the actual width of a poset (and not just the depth of a run). This is achieved by a stack where the widths encountered up to the last fork transition are stored. More precisely: let G be a run and x a node in G. We describe the content of the stack that the new automaton assumes at the node x. Firstly, the stack at x contains all elements already on the stack at the source $\mathrm{src}(G)$. If x is at same depth level as $\mathrm{src}(G)$, then only the topmost element of the stack may be changed. Now, as many elements as the difference between the depth level of x and that of $\mathrm{src}(G)$ are put on the stack additionally. Those fork transitions between $\mathrm{src}(G)$ and x that are unmatched before x (*i.e.* not closed by a join transition) form a sequence. Two consecutive such forks either follow each other in a cascade immediately, or they limit a sub-run of G consisting of all the nodes in between them. For each of those forks an element is pushed onto the stack which equals the width of the enclosed sub-run , or equals 1 if the consecutive forks form a cascade. The topmost element of the stack determines the width of the sub-run between the last unmatched fork and x. If there is no unmatched fork, *i.e.* x is at the same depth level as $\mathrm{src}(G)$, then the topmost element is the maximum of the topmost element at $\mathrm{src}(G)$ and the width of the sub-run between $\mathrm{src}(G)$ and x. In order to limit the successful runs 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 from now on. If $G = G_1 \cdot \ldots \cdot G_m$ then G_i is a *direct sub-run* of G for each $i = 1, \ldots, m$. Similarly, for $G = \|_{p,q}(G_1, \ldots, G_n)$ the runs $G_1, \ldots, G_n$ are *direct sub-runs* of G. We write $H \sqsubset G$ if H is a direct sub-run of G. The reflexive and transitive closure of $\sqsubset$ is denoted by $\preceq$. If $H \preceq G$ we say H is a *sub-run* of G. If $H \preceq G$ and $H \neq G$ we call H a *proper sub-run* of G. By $S_{|n}$ we denote the *restriction* of S to the sp-posets of width less than or equal to n, *i.e.*

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

Lemma 5.8. *Let $\mathbb{K}$ be an arbitrary bisemiring, $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ and $n \in \mathbb{N}$ with $n \geq 1$. If S is regular then $S_{|n}$ can be recognized by a depth-bounded wba.*

Proof. Let $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ be a wba 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 non-empty words over $[n]$. From $\mathcal{A}$ we construct a new automaton $\mathcal{A}'$ with state set $Q' = \{(p, w) \in Q \times [n]^+ \mid |w| \leq n\}$ 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 run and the values stored within this stack keep track of the width of the label of the run.

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

$$\mu'_{\text{seq}}((p, u), a, (q, v)) = \begin{cases} \mu_{\text{seq}}(p, a, q) & \text{if } u = v, \\ 0 & \text{otherwise.} \end{cases}$$

- A fork transition increases the depth of a run, hence it pushes a new value onto the stack. Since there are no parallel actions after the fork yet, this value is 1. Hence, for all $m \in \{2, \ldots, |Q|\}$:

$$\mu'^{m}_{\text{fork}}\left((p, u), \left\{(p_1, u_1), \ldots, (p_m, u_m)\right\}\right)$$
$$= \begin{cases} \mu^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } u_1 = \cdots = u_m = u1, \text{[1]} \\ 0 & \text{otherwise.} \end{cases}$$

  If $m > |Q|$ then μ'^{m}_{fork} equals constantly zero.

- Since a join transition results in a node of smaller depth, it decreases the size of the stack. The width of the sub-run[2] since the matching fork transition f is the sum of the widths of its parallel sub-runs, *i.e.* of the top stack elements at the nodes joined. The width of the sub-run since the previous unmatched fork transition f' is the maximum of this sum and the width of the sub-run 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.

[1] Here $u1$ denotes the word from $[n]^\star$ concatenated of u and the letter $1 \in [n]$.
[2] By the "width of a sub-run" we mean the width of the label of the sub-run.

Hence, for any $m \in \{2, \ldots, |Q|\}$:

$$\mu'^{m}_{\text{join}}\left(\left\{(q_1, v_1), \ldots, (q_m, v_m)\right\}, (q, v)\right) = \mu^{m}_{\text{join}}(\{q_1, \ldots, q_m\}, q)$$

if there are $w \in [n]^\star$ and $x, y_1, \ldots, y_m, z \in [n]$ such that $v_i = wxy_i$ for $i = 1, \ldots, m$, and $v = wz$ with $z = \max\{x, y_1 + \cdots + y_m\}$.

Otherwise, $\mu'^{m}_{\text{join}}\left(\left\{(q_1, v_1), \ldots, (q_m, v_m)\right\}, (q, v)\right) = 0$, also for $m > |Q|$.

- At the source of a successful run, 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 run 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 run 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 show firstly that a run 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 run in which case it gets replaced.

Claim 1. Let $G : (p, u) \xrightarrow{t} (q, v)$ be a run of $\mathcal{A}'$ with $\text{wd}(t) \leq n$. Then $u = wx$, $v = wy$ with $w \in [n]^\star$, $x, y \in [n]$, and $y = \max\{x, \text{wd}(t)\}$.

Let $G = (p, u) \xrightarrow{t} (q, v)$ be atomic. Then $t \in \Sigma$ and $\text{wd}(t) = 1$. Hence, Claim 1 follows immediately from the definition of μ'_{seq}. If $G = G_1 \cdot \ldots \cdot G_m$ is the sequential decomposition of G with $\mathbf{lab}(G_i) = t_i$, $\mathbf{lab}(G) = t$ and $\text{wd}(t) \leq n$ we have $\text{wd}(t) = \max\{\text{wd}(t_i) \mid i = 1, \ldots, m\}$. By structural induction we get Claim 1 for the run G. Now let $G =$

$\|_{(p,u),(q,v)}(G_1, \dots, G_m)$ with $G_i = (p_i, u_i) \xrightarrow{t_i} (q_i, v_i)$ for $i = 1, \dots, m$. For $t = \mathbf{lab}(G)$ we have $\mathrm{wd}(t) \le n$ by assumption and $t = t_1 \| \dots \| t_m$. Hence, $\mathrm{wd}(t_i) < n$. By the definition of fork transitions in $\mathcal{A}'$ we have $u_1 = \dots = u_m = u1$. By induction there are $x_i \in [n]$ such that $v_i = ux_i$ and $x_i = \mathrm{wd}(t_i)$ for $i = 1, \dots, m$. 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 $y = \max\{x, x_1 + \dots + x_m\} = \max\{x, \mathrm{wd}(t_1) + \dots + \mathrm{wd}(t_m)\} = \max\{x, \mathrm{wd}(t)\}$. This is Claim 1.

Now consider $t \in \mathrm{SP}$ with $\mathrm{wd}(t) > n$ and suppose H is a run in $\mathcal{A}'$ with label t. By decomposition of H there is a sub-run G of H with $\mathrm{wd}(\mathbf{lab}(G)) > n$ and $G = \|_{p',q'}(G_1, \dots, G_m)$ such that $\mathrm{wd}(\mathbf{lab}(G_i)) \le n$ for $i = 1, \dots, m$. Let $p' = (p, u)$ and $q' = (q, v)$. For $G_1, \dots, G_m$ we can apply Claim 1. If $u = wx$ with $w \in [n]^\star$ and $x \in [n]$ then $v = wy$ with $y = \max\{x, \mathrm{wd}(t)\} > n$ by definition of μ'_{fork} and μ'_{join}. Since $y \le n$ by definition, a run G with $\mathrm{wd}(\mathbf{lab}(G)) > n$ cannot exist. Hence, for all $t \in \mathrm{SP}$ with $\mathrm{wd}(t) > n$ we have $t \notin \operatorname{supp} \mathcal{S}(\mathcal{A}')$.

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

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

For any run G' of $\mathcal{A}'$ we define $g(G')$ by just forgetting the second component (*i.e.* the word) of the states of run G'. By definition of $\mathcal{A}'$ the mapping g is well defined and preserves labels and weights of a run. Now let $g_{r,s} : \mathcal{R}_{r',s'(\forall\alpha)}(\mathcal{A}') \to \mathcal{R}_{r,s(\le n)}(\mathcal{A})$ be the restriction of g to $\mathcal{R}_{r',s'(\forall\alpha)}(\mathcal{A}')$. Then $g_{r,s}$ is injective because the second component of the states of a run $G' \in \mathcal{R}_{r',s'(\forall\alpha)}(\mathcal{A}')$ is determined by $g_{r,s}(G')$ and r'. Our next aim is to show that any run of $\mathcal{A}$ can be simulated by a run of $\mathcal{A}'$ provided the width of the label is at most n, *i.e.* we want to show surjectivity of $g_{r,s}$. We prove this by induction on the depth of the run. The following claim forms the inductive argument:

Claim 2. Let $G : p \xrightarrow{t} q$ be a run in $\mathcal{A}$ with $\mathrm{wd}(t) \le n$ and $\mathrm{dp}(G) = d$.

Furthermore, let $u \in [n]^+$ with $|u| + d \leq n$. Then there are $v \in [n]^+$ and a run $G' = (p, u) \xrightarrow{t} (q, v)$ in $\mathcal{A}'$ such that $g(G') = G$.

For any atomic run G the depth of G is 0 and Claim 2 is obvious. Next, let $G = G_1 \cdot \ldots \cdot G_m$ allow a sequential decomposition. Note that $\mathrm{dp}(G) = \max\{\mathrm{dp}(G_i) \mid i = 1, \ldots, m\}$. By structural induction, Claim 2 is true for G_1. Considering Claim 1 and applying induction to $G_2, \ldots, G_m$, Claim 2 holds true for G too. Now, let $G = \parallel_{p,q}(G_1, \ldots, G_m)$ have a parallel decomposition. Let $G_i : p_i \to q_i$ for $i = 1, \ldots, m$. We construct G' as follows: It starts in (p, u) with a fork simulating the first fork in G. This is possible because $|u| + d \leq n$. This fork branches into states whose second component is $u1$. Note that $\mathrm{dp}(G_i) \leq d - 1$ for $i = 1, \ldots, m$. Applying the induction hypothesis for $G_1, \ldots, G_m$ we get runs $G'_1, \ldots, G'_m$ with $g(G'_i) = G_i$. By Claim 1, for $i = 1, \ldots, m$ the run G'_i ends in a state whose second component is ux_i where $x_i = \max\{1, \mathrm{wd}(\mathbf{lab}(G'_i))\} = \mathrm{wd}(\mathbf{lab}(G'_i))$. Since $x_1 + \cdots + x_m = \mathrm{wd}(\mathbf{lab}(G'_1)) + \cdots + \mathrm{wd}(\mathbf{lab}(G'_m)) = \mathrm{wd}(\mathbf{lab}(G)) \leq n$ there is the required join transition simulating the last join in G. Then $g(G') = G$ is obvious by construction. This proves Claim 2.

Now we return to the mapping $g_{r,s}$. Any run G from $\mathcal{R}_{r,s(\leq n)}(\mathcal{A})$ satisfies the prerequisites of Claim 2. By Claims 2 and 1, there is $G' \in \mathcal{R}_{r',s'(\forall\alpha)}(\mathcal{A}')$ with $g_{r,s}(G') = G$. Hence, $g_{r,s}$ is surjective. Thus, there is a bijective mapping $g_{r,s} : \mathcal{R}_{r',s'(\forall\alpha)}(\mathcal{A}') \to \mathcal{R}_{r,s(\leq n)}(\mathcal{A})$ preserving labels and weights 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 $\mathcal{S}(\mathcal{A}') = S_{|n}$. □

As a consequence of the last result we get:

Corollary 5.9. *Let S be a regular sp-series. Then S is of bounded width iff it can be recognized by a wba of bounded depth.*

Proof. Let S be of bounded width, and let $n = \max(\mathrm{wd}(\mathrm{supp}\, S))$. Hence, $S = S_{|n}$. By Lemma 5.8, S can be recognized by a wba of bounded depth. On the other hand, if S is recognized by a wba of bounded depth d, then S is of bounded width by Proposition 5.1. □

Since the construction in the proof of Lemma 5.8 does not change the maximal arity of fork and join transitions of the wba in consideration, we have:

Corollary 5.10. *Let S be recognized by a wbba. Then S is of bounded width iff S is recognized by a wbba of bounded depth.*

Corollary 5.11. *Every sequential-rational sp-series S is C-recognized by a wba, and even a wbba, of bounded depth.*

Proof. By Corollary 4.18, every sequential-rational sp-series S is C-regular. Due to Proposition 3.13, S has bounded width. Therefore, by Corollary 5.9, S is C-recognized by a wba of bounded depth. Applying Theorem 4.17 and Corollary 5.10, S is even C-recognized by a wbba of bounded depth. □

Remark 5.12. Construction 5.2 of the d-depth counter wba $\mathcal{A}_{|d}$ of a wba $\mathcal{A}$ and the construction in the proof of Lemma 5.8 for the restriction $S_{|n}$ are effective.

Note. Let $\mathcal{A}$ be a wba recognizing S. Suppose we know that S is of bounded width. The construction from the proof of Lemma 5.8 can only be used to get a wba of bounded depth recognizing S if we know a uniform upper bound for the width of the support of S.

Next, we present a concept equivalent to the property of bounded depth. Let $\mathcal{A}$ be a wba and $G = \|_{p,q}(G_1, \dots, G_n)$ a C-run (M-run) in $\mathcal{A}$. Let f denote the starting fork transition of G and j the finishing join transition of G. Then we say that (f, j) is a *matching pair* in the C-mode (M-mode) and that G is *limited* by (f, j). For two matching pairs (f, j) and (f', j') we put $(f, j) \prec (f', j')$ if there exists a parallel C-run (M-run) G limited by (f', j') that contains a proper sub-run limited by (f, j). If the relation $\prec$ on the set of all matching pairs in the C-mode (M-mode) of $\mathcal{A}$ is irreflexive, then $\mathcal{A}$ is called *C-fork acyclic* (*M-fork acyclic*)[3]. The relation $\prec$ is called the *nesting relation*. Note that every C-fork acyclic wba is also M-fork acyclic. But there are M-fork acyclic automata which are not C-fork acyclic, cf. the wba presented in Figure 7.1. A run where a matching pair is nested within itself is called a *cyclic run*. The following lemma is true for both running modi.

[3] The notion of fork acyclicity was introduced by Lodaya and Weil [LW00]. There, they forbid the nesting of matching pairs within itself for successful runs only. But in [LW01] they also defined this notion considering all runs.

Lemma 5.13. *Let $\mathcal{A}$ be a wba. $\mathcal{A}$ is fork acyclic iff $\mathcal{A}$ is of bounded depth.*

Proof. Let $\mathcal{A}$ be fork acyclic. Assume $\mathcal{A}$ is not of bounded depth. Hence, for every $d \in \mathbb{N}$ there is a run of depth greater than d. Let B be the number of matching pairs of $\mathcal{A}$. Note that B is finite because there are only finitely many fork and join transitions. Choose $d > B$, and let G be a run of depth at least d. Then G contains a sequence $(G_i)_{i=1,\ldots,d}$ of d mutually different sub-runs such that

- G_{i+1} is a proper sub-run of G_i for $i = 1, \ldots, d-1$, and
- G_i is a parallel run limited by a matching pair (f_i, j_i) for $i = 1, \ldots, d$.

Since $d > B$ there are $i_1 \neq i_2 \in \{1, \ldots, d\}$ with $(f_{i_1}, j_{i_1}) = (f_{i_2}, j_{i_2})$. This contradicts $\prec$ being irreflexive. Hence, $\mathcal{A}$ is of bounded depth.

Vice versa, assume $\prec$ is not irreflexive, *i.e.* there is a matching pair (f, j) and a cyclic run G limited by (f, j) where G has a proper sub-run H also limited by (f, j). Hence, $\mathrm{dp}(G) > \mathrm{dp}(H)$. We construct an infinite sequence of cyclic runs in $\mathcal{A}$ which all contain H as a proper sub-run as follows:

- $G_0 = G$,
- if G_i is given, then we get G_{i+1} by substituting[4] the sub-run H of G_i by the run G.

Since H and G are limited by the same matching pair, each G_i is a run of $\mathcal{A}$ for $i \in \mathbb{N}$.[5] Moreover, $\mathrm{dp}(G_{i+1}) > \mathrm{dp}(G_i)$ for almost all i because of $\mathrm{dp}(G) > \mathrm{dp}(H)$. Hence, $\mathcal{A}$ is not of bounded depth. □

We conclude this chapter by discussing decidability of bounded depth and bounded width. For this, we adapt results by Lodaya and Weil [LW01]

[4] We omit the formal details of this substitution. The sub-graph H of G_i is replaced by the graph G and all arrows going to the source of H now go to the source of G, and dually for the sink of H.

[5] Note that G_{i+1} is an M-run if G_i is an M-run.

for series-Σ-languages[6] and the respective branching automata. We restrict the bisemirings considered to the class of positive bisemirings. A bisemiring $\mathbb{K}$ is *positive* if $k * l = 0$ implies $k = 0$ or $l = 0$ for $* \in \{\oplus, \circ, \diamondsuit\}$ and all $k, l \in \mathbb{K}$. That means $\mathbb{K}$ is zero-sum-free and zero-divisor-free for both products. Before discussing decidability of bounded width we state a result on the decidability of empty support needed in the sequel.

Lemma 5.14. *Let $\mathbb{K}$ be a positive bisemiring and $\mathcal{A}$ a wba over $\mathbb{K}$. It is decidable whether $\mathcal{S}_C(\mathcal{A})$ and $\mathcal{S}_M(\mathcal{A})$ have empty support or not.*

The idea of the proof is due to Lodaya and Weil [LW01, Prop. 2.4].

Proof. Let $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$. We define three different sets of pairs of states from Q:

- C is the set of all pairs (p, q) with $p, q \in Q$ such that there is a C-run from p to q on some sp-poset t,
- M is the set of all pairs (p, q) such that there is an M-run from p to q on some $t \in \mathrm{SP}$, and
- $\overline{M}$ is the set of all pairs (p, q) such that there is a sequential M-run from p to q on some $t \in \mathrm{SP}$.

Since $\mathbb{K}$ is positive, it suffices to decide whether C and M, respectively, contain a pair $(\mathfrak{i}, \mathfrak{f})$ with $\mathfrak{i}$ an initial and $\mathfrak{f}$ a final state. If so, then the support is not empty, otherwise it is.

We show how to compute C and M. For any $R \subseteq Q \times Q$ let R^+ denote the transitive closure of R, let $R^2 = R \circ R$, $R^3 = R^2 \circ R$ where $\circ$ is the usual relation product. Let T be the set of all pairs (p, q) such that there is an $a \in \Sigma$ with $\mu_{\text{seq}}(p, a, q) \neq 0$. We put $C_0 = M_0 = \overline{M}_0 = T^+$. Now, let C_k, M_k, and $\overline{M}_k$ for some $k \in \mathbb{N}$ be constructed. Let R_k^C and R_k^M, respectively, be the set of all pairs (p, q) such that there is a fork transition $p \to \{p_1, \ldots, p_m\}$, a join transition $\{q_1, \ldots, q_m\} \to q$, a permutation

[6] A series-Σ-algebra is a Σ-algebra equipped with an extra binary, associative operation $\circ$, called the sequential product. A series-Σ-language is a subset of the free series-Σ-algebra $S_\Sigma(A)$ over an alphabet A.

$\alpha \in S_m$[7] such that $(p_i, q_{\alpha(i)}) \in C_k$ and $(p_i, q_{\alpha(i)}) \in \overline{M}_k$, respectively, for all $i = 1, \ldots, m$. Then we put $C_{k+1} = (C_k \cup R_k^C)^+$, $M_{k+1} = (M_k \cup R_k^M)^+$, and $\overline{M}_{k+1} = (T \cup M_{k+1}^2 \cup M_{k+1}^3)^+$. By induction on k, one shows easily that

- C_k is the set of all pairs (p, q) such that there is a C-run from p to q of depth at most k,
- M_k is the set of all pairs (p, q) such that there is an M-run from p to q of depth at most k, and
- $\overline{M}_k$ is the set of all pairs (p, q) such that there is a sequential M-run from p to q of depth at most k.

By definition $C_{k+1} \supseteq C_k$ and $M_{k+1} \supseteq M_k$. Moreover, if $C_{k+1} = C_k$ then $C_n = C_k$ for all $n \geq k$. Similarly, $M_{k+1} = M_k$ implies $\overline{M}_{k+1} = \overline{M}_k$, and hence $M_n = M_k$ for all $n \geq k$. Thus, C and M are computable, and the problem of empty support is decidable. □

Fork acyclicity and bounded depth are properties of the set of runs of a given wba $\mathcal{A}$ only. It is not necessary to evaluate the weights of the runs, and, thus, to consider the behavior of $\mathcal{A}$ to decide these properties. Therefore, no restrictions are imposed on the bisemiring in the following lemma.

Lemma 5.15. *Let $\mathcal{A}$ be a wba over an arbitrary bisemiring $\mathbb{K}$. It is decidable whether $\mathcal{A}$ is C-fork and M-fork acyclic or not.*

The proof enhances an idea of Lodaya and Weil [LW01, Prop. 6.12].

Proof. Let C, M, and $\overline{M}$ be like in the proof of Lemma 5.14. We have shown in this proof that C, M, and $\overline{M}$ are computable. We construct a set $B_C \subseteq C$ as follows. Let $(p, q) \in C$. If there is a fork transition $p \to \{p_1, \ldots, p_m\}$, a join transition $\{q_1, \ldots, q_m\} \to q$, an $\alpha \in S_m$ such that $(p_i, q_{\alpha(i)}) \in C$ for $i = 1, \ldots, m$, then put $(p, q) \in B_C$. Hence, $(p, q) \in B_C$ iff there is a parallel non-atomic C-run between p and q. Similarly, we

[7] Here, S_m denotes the symmetric group on m elements.

construct by using M and $\overline{M}$ a set $B_M \subseteq M$ with $(p,q) \in B_M$ iff there is a parallel non-atomic M-run between p and q.

Now, we build a directed graph Γ_C with vertex set C and two kinds of directed edges as follows. For all $(\tilde{p},\tilde{q}) \in B_C$, fork transitions $\tilde{p} \to \{p_1,\dots,p_m\}$, join transitions $\{q_1,\dots,q_m\} \to \tilde{q}$, $\alpha \in S_m$ with $(p_i, q_{\alpha(i)}) \in C$ for $i = 1,\dots,m$ we add a red arrow from $(\tilde{p},\tilde{q})$ to $(p_i, q_{\alpha(i)})$ for every $i = 1,\dots,m$. Furthermore, for every $(p,q) \in C$ and every $(\tilde{p},\tilde{q}) \in B_C$ we put a blue arrow from (p,q) to $(\tilde{p},\tilde{q})$ if

- $p = \tilde{p}$ and $(\tilde{q}, q) \in C$, or
- $q = \tilde{q}$ and $(p, \tilde{p}) \in C$, or
- $(p,\tilde{p}) \in C$ and $(\tilde{q}, q) \in C$.

Claim 1. $\mathcal{A}$ is C-fork acyclic iff Γ_C has no cycle with a red arrow.

Indeed, if $\mathcal{A}$ is not C-fork acyclic it follows immediately from the construction of Γ_C that a cycle with a red arrow exists. Vice versa, suppose there is a cycle in Γ_C with a red arrow from (p,q) to (p',q'). By definition of red arrows, $(p,q) \in B_C$ and there is a run $G = \|_{p,q}(G_1,\dots,G_m)$ from p to q such that G_i is a run from p' to q' for some i. Since there is a path from (p',q') to (p,q) in Γ_C, there is a run H from p' to q' with a sub-run $\widetilde{H}$ from p to q. Hence, substitution of $\widetilde{H}$ by G in H yields a run H' from p' to q'. Now substituting H' for G_i in G implies the existence of a run $G' = \|_{p,q}(G_1,\dots,G_{i-1},H',G_{i+1},\dots,G_m)$ where a matching pair is nested within itself. Hence, $\mathcal{A}$ is not C-fork acyclic.

Since it is decidable whether Γ_C has a cycle with a red edge or not, C-fork acyclicity of $\mathcal{A}$ is decidable.

Decidability of M-fork acyclicity is shown in a similar way, constructing a graph Γ_M with vertex set M. $\square$

Opposed to the notion of fork acyclicity or bounded depth, we have to calculate weights in order to decide whether a regular sp-series is of bounded width. As we have seen in Example 5.7 a wba of unbounded depth may recognize a series of bounded width. Therefore, we turn to positive bisemirings now.

Theorem 5.16. *Let $\mathbb{K}$ be a positive bisemiring and S a C-regular or M-regular sp-series. It is decidable whether S is of bounded width.*

Proof. Let $\mathcal{A}$ be a wba recognizing S. Let C and M be as in the last two proofs. By Lemma 5.15 we can decide whether $\mathcal{A}$ is fork acyclic or not. But moreover, we can even compute the set $Z_C \subseteq C$, and $Z_M \subseteq M$ respectively, with $(p, q) \in Z_C$ iff there is a C-run from p to q in which a matching pair is nested within itself, and similarly for Z_M. We compute Z_C as follows. Firstly, we compute all the elements of all cycles of Γ_C that have at least one red arrow. Obviously, every such element belongs to Z_C. Next, we compute all pairs $(p, q) \in C$ such that there is a path from (p, q) to an element of a cycle with a red arrow. All these pairs also belong to Z_C because there is a run from p to q that contains a cyclic run as a sub-run. Obviously, no other pair of states belongs to Z_C. The set Z_M is calculated similarly.

Finally, we look for all pairs $(\mathfrak{i}, \mathfrak{f}) \in Z_C$, and $(\mathfrak{i}, \mathfrak{f}) \in Z_M$ respectively, where $\mathfrak{i}$ is initial and $\mathfrak{f}$ is final, and denote this set by W. Since $\mathbb{K}$ is positive, $S = \mathcal{S}(\mathcal{A})$ is of bounded width iff $W = \emptyset$. Indeed, $W = \emptyset$ implies that there is no successful cyclic run. Hence, the depth of all successful runs is uniformly bounded, and, therefore, also the width of S. On the other hand, suppose $W \neq \emptyset$. Then there are successful runs of arbitrary depth. Positiveness of $\mathbb{K}$ implies the existence of sp-posets of arbitrary width in the support of S. Hence, S is not of bounded width. □

Remark 5.17. It is an open question if emptiness and bounded width of the support of regular sp-series are decidable for non-positive underlying bisemirings. For classical weighted automata over words, Eilenberg [Eil74, Thm. 8.1] showed the decidability of the equality problem (and, hence, of the problem of empty support) for regular series in case the underlying semiring is a sub-semiring of a field F. This includes $\mathbb{Z}$ which is not positive. It is not clear if one may carry over that result to branching automata.

6 The Fundamental Theorem

In this chapter we prove the converse of Corollary 4.18 for sp-series of bounded width. Moreover, we will show the coincidence of the class of sequential-rational sp-series with the class of C-regular sp-series recognized by a wba of bounded depth.

Theorem 6.1. *Let $\mathcal{A}$ be a wba of bounded depth over an arbitrary bisemiring $\mathbb{K}$. Then both $\mathcal{S}_C(\mathcal{A})$ and $\mathcal{S}_M(\mathcal{A})$ are sequential-rational sp-series.*

Proof. For this proof, f denotes a fork and j a join transition. By M we denote the set of all matching pairs. Let $G = G_1 \cdot \ldots \cdot G_m$ be a run of $\mathcal{A}$ in its sequential decomposition ($m \geq 1$). Then we say that a matching pair (f, j) is used at the upper level of G if there is a G_i ($i \in \{1, \ldots, m\}$) limited by (f, j).

Firstly, we prove Theorem 6.1 for the C-running mode. By Lemma 5.13, $\mathcal{A}$ is fork acyclic, *i.e.* the nesting relation $\prec$ on M is irreflexive. Substituting sub-runs it is easy to see that $\prec$ is transitive. Let $\preceq$ be the reflexive closure of $\prec$. Then $\preceq$ is anti-symmetric. Indeed, for $(f, j) \preceq (f', j')$ and $(f', j') \preceq (f, j)$ either $(f, j) = (f', j')$ or $(f, j) \prec (f', j')$ and $(f', j') \prec (f, j)$. The latter case implies $(f, j) \prec (f, j)$ in contradiction to $\prec$ being irreflexive. Hence, $(f, j) = (f', j')$. Thus, $\preceq$ is a partial order and can be extended to a linear one.

We fix an arbitrary linear extension $\sqsubseteq$ of the partial order $\preceq$ on the set of matching pairs M and consider the linearly ordered set $(M, \sqsubseteq)$. Let $J \subseteq M$ and p, q be states of $\mathcal{A}$. We denote by $S^J_{p,q}$ the series with

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

where $t \in \mathrm{SP}$ and the runs G, over which the sum extends, are such that only matching 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 $|J|$.

Firstly, let $|J| = 0$, *i.e.* no forks and joins are used. Then $S^{\varnothing}_{p,q}$ is a regular sp-series where the parallel product is not used, *i.e.* it can be identified with a regular word series with values from the semiring $(K, \oplus, \circ, 0, 1)$ as pointed out in Remark 3.11. By a result of Schützenberger [Sch61b] we know that $S^{\varnothing}_{p,q}$ 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 $f : r \to_k \{r_1, \ldots, r_n\}$ and $j : \{s_1, \ldots, s_n\} \to_l s$. We define the sp-series $S(f,j)$ for every $t \in \mathrm{SP}$ by

$$\Big(S(f,j), t\Big) = \bigoplus_{\substack{G : r \xrightarrow{t} s \\ G \text{ limited by } (f,j)}} \mathbf{wgt}(G) \tag{6.1}$$

where the sum ranges over all parallel runs $G = \|_{r,s}(H_1, \ldots, H_n)$ limited by (f,j) and having label t. But then $H_1, \ldots, H_n$ contain only matching 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 $(M, \sqsubseteq)$. Let S_n denote the permutation group on $\{1, \ldots, n\}$. Thus, we have:

$$S(f,j) = k \cdot \left(\sum_{\pi \in S_n} \left[S^{J'}_{r_1, s_{\pi(1)}} \parallel \ldots \parallel S^{J'}_{r_n, s_{\pi(n)}} \right] \right) \cdot l \tag{6.2}$$

and the sp-series $S^{J'}_{r_i, s_{\pi(i)}}$ with $i \in \{1, \ldots, n\}$ and $\pi \in S_n$ 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 runs from p to q with matching pairs from J only use the maximal element (f,j) of J at the upper level only. Thus, $S^{J}_{p,q}$ can be built from $S(f,j)$ and $S^{J'}_{p',q'}$ for some $p', q' \in Q$ using sequential-rational operations. Unfortunately, we have to distinguish eight cases. Here, we deal in detail with two cases only.

We get for $p \neq r$, $s \neq q$ and $r = s$:

$$S^{J}_{p,q} = S^{J'}_{p,q} + S^{J'}_{p,r} \cdot S(f,j)^{+} \cdot S^{J'}_{s,q} + S^{J'}_{p,r} \cdot S(f,j)^{+} \cdot \left(S^{J'}_{s,r} \cdot S(f,j)^{+} \right)^{+} \cdot S^{J'}_{s,q} \tag{6.3}$$

where the first sp-series $S^{J'}_{p,q}$ of this sum covers all runs that do not use (f,j) at the upper level. The second one covers all runs $G_1 \cdot G_2 \cdot G_3$ such that G_2 is a sequence of consecutive "bubbles" from $S(f,j)$, but neither in G_1 nor in G_3 (f,j) appears as a matching pair. Note that $S(f,j)$ can be iterated because of $r = s$. The third series covers all runs where we have more than one such sequence of consecutive "bubbles" from $S(f,j)$.

Similarly, we get for $p \neq r$, $q \neq s$ and $r \neq s$:

$$S^J_{p,q} = S^{J'}_{p,q} + S^{J'}_{p,r} \cdot S(f,j) \cdot S^{J'}_{s,q} + S^{J'}_{p,r} \cdot S(f,j) \cdot \Big(S^{J'}_{s,r} \cdot S(f,j) \Big)^{+} \cdot S^{J'}_{s,q} \tag{6.4}$$

for which the argumentation is almost the same as that for Equation (6.3) despite the fact that this time $S(f,j)$ is not iterated because $r \neq s$.[1]

The sp-series $S(f,j)$ is sequential-rational as seen before. All other sp-series appearing on the right hand side of Equations (6.3) and (6.4) (and of those equations we omitted) 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. M is an initial segment of $(M, \sqsubseteq)$. Thus, $S^M_{p,q}$ is sequential-rational for all $p, q \in Q$. Hence,

$$\mathcal{S}_C(\mathcal{A}) = \sum_{p,q \in Q} \lambda(p) \cdot S^M_{p,q} \cdot \gamma(q)$$

is sequential-rational.

The proof for $\mathcal{S}_M(\mathcal{A})$ is more involved because Equation (6.2) is not true anymore. In Equation (6.2) the sp-series $S^{J'}_{r_i, s_{\pi(i)}}$ have to be replaced by sp-series that sum up over sequential runs between r_i and $s_{\pi(i)}$ only. To cover this situation we introduce the sp-series $T^J_{p,q}$ with

$$(T^J_{p,q}, t) = \bigoplus_{\substack{G : p \xrightarrow{t} q \\ G \text{ sequential}}} \mathbf{wgt}(G)$$

where $t \in \mathrm{SP}$, and where the sum extends over all sequential M-runs G from p to q with label t that use matching pairs from J only. In particular, the support of $T^J_{p,q}$ contains sequential sp-posets only. Note that $S^{\varnothing}_{p,q} = T^{\varnothing}_{p,q}$. The sp-series $S(f,j)$ is defined as in Equation (6.1). But this time we sum

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

up over M-runs of course. With the same notation as given above we get this time

$$S(f,j) = k\cdot\left(\sum_{\pi\in S_n}\left[T^{J'}_{r_1,s_{\pi(1)}} \parallel \ldots \parallel T^{J'}_{r_n,s_{\pi(n)}}\right]\right)\cdot l \tag{6.5}$$

where the sp-series $T^{J'}_{r_i,s_{\pi(i)}}$ are sequential-rational by induction hypothesis. Now we proceed as for C-running behavior showing that $S^J_{p,q}$ is sequential-rational for all $p,q \in Q$. But, moreover, we have to show that $T^J_{p,q}$ is sequential-rational too. Again, we have to distinguish some cases. Here, we state only the case dual to Equation (6.3). For $p \neq r$, $q \neq s$, and $r = s$ we have

$$T^J_{p,q} = T^{J'}_{p,q} + S^{J'}_{p,r}\cdot S(f,j)^+\cdot S^{J'}_{s,q} + S^{J'}_{p,r}\cdot S(f,j)^+\cdot\left(S^{J'}_{s,r}\cdot S(f,j)^+\right)^+\cdot S^{J'}_{s,q} \tag{6.6}$$

which is almost equivalent to Equation (6.3). Only the first addend differs because no parallel product is allowed. In the same way as for $S^J_{p,q}$ we get that $T^J_{p,q}$ is sequential-rational which serves as the induction hypothesis for Equation (6.5). Now we conclude as shown above that $\mathcal{S}_M(\mathcal{A})$ is sequential-rational. $\square$

The special case $\mathbb{K} = \mathbb{B}$ was shown by Lodaya and Weil [LW00]. Their proof uses a nested induction which we simplified to just one induction along the linear order of matching pairs.

Note. The characterization of the behavior of general wba by rational operations, as defined here, is not successful even in the case of the Boolean bisemiring. The obvious idea to allow in addition the parallel iteration does not give the desired result. Lodaya and Weil [LW00, Section 5] give an example of a regular sp-language not being rational, in fact the least language L containing letter c and where $x \in L$ implies $a \parallel (bx) \in L$. The problem arises from the fact that for a rational sp-series S there is an $n \in \mathbb{N}$ 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 $\parallel$. But, as it can be seen by means of the sp-language L given above, there are regular sp-series whose support allows an unbounded alternation of $\cdot$ and $\parallel$.

In [LW01], Lodaya and Weil define another notion of rationality for sp-languages. This notion of rationality uses ξ-substitution and a restricted

ξ-exponentiation following Thatcher's and Wright's definition of rationality for tree languages [TW68]. With this stronger concept of rationality they are able to show the equivalence of regular and rational sp-languages. However, it is not clear how to generalize these notions for sp-series over bisemirings. This is due to the problem of defining a useful ξ-substitution. One may succeed if the underlying bisemiring is actually a semiring. This is subject of further research and is discussed more precisely in the conclusion of this thesis.

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

Theorem 6.2. *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 C-recognized by a wba of bounded depth.*
3. *S is C-regular and has bounded width.*

Proof. Due to Corollary 4.18 and Proposition 3.13 (1) implies (3). By Corollary 5.9, (3) and (2) are equivalent. (2) implies (1) by Theorem 6.1. □

Moreover, every sequential-rational sp-series is C-recognized by a wba of certain kind.

Theorem 6.3. *Let $\mathbb{K}$ be a bisemiring. Every sequential-rational sp-series $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is C-recognized by a normalized wbba of bounded depth.*

Proof. By Corollary 5.11, S is C-recognized by a wbba of bounded depth. It is obvious that the normalization construction in the proof of Proposition 4.4 does not affect the bounded depth property of $\mathcal{A}$. Applying Corollary 4.5 we get the claim. □

By putting $\mathbb{K} = \mathbb{B}$ we get as a consequence of Theorem 6.2 the result of Lodaya and Weil [LW00, Thm. 4.12].

Corollary 6.4. *Let $L \subseteq \mathrm{SP}$ 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 fork acyclic branching automaton.*
3. *L is regular and has bounded width.*

7 The Different Concepts of Regularity

In this chapter we compare the two concepts of regularity defined for wba, *i.e.* the cascade branching and the maximally branching running mode.

In Example 4.10 we have considered the sp-language $L = \{a\}$ and have shown that

$$L^{\boxplus} = \{\underbrace{a \parallel \ldots \parallel a}_{n} \mid n \geq 1\}$$

is not M-regular. But $L^{\boxplus}$ is C-regular. For instance, it is C-recognized by the automaton $\mathcal{A}$ depicted in Figure 7.1. Thus, in the case of unbounded width there are C-regular sp-series that are not M-regular.

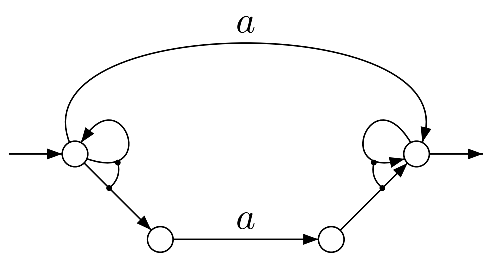


Figure 7.1: A branching automaton $\mathcal{A}$ C-recognizing $\{a\}^{\boxplus}$.

Firstly, we show that every M-regular sp-series of bounded width is C-regular.

Proposition 7.1. *Let $\mathcal{A}$ be a wba of bounded depth over the bisemiring $\mathbb{K}$. Then there is another wba $\mathcal{A}'$ with $\mathcal{S}_M(\mathcal{A}) = \mathcal{S}_M(\mathcal{A}') = \mathcal{S}_C(\mathcal{A}')$.*

The main idea of the following proof is to "destroy" the runs of $\mathcal{A}$ that are C-runs only. Note that in the M-running mode a direct sub-run of a parallel run has to be an atomic or a sequential run. Therefore, either no fork and join transitions occur in this direct sub-run or there are at least three states at the upper level of this sub-run (cf. Figure 3.4). We will control these conditions in the new automaton $\mathcal{A}'$ by keeping track of the occurrence of join transitions and by counting the states of the same depth up to three.

Proof. Let d be the maximal depth of an M-run in the automaton $\mathcal{A} = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$. We put $[3] = \{1, 2, 3\}$, $[3]^\star$ denotes the set of all finite words over $[3]$, and $[3]^+$ the set of all non-empty finite words over $[3]$. We define the new automaton $\mathcal{A}'$ as follows:

- $Q' = \{(p, u, i) \in Q \times [3]^+ \times \{0, 1\} \mid |u| \leq d + 1\}$,
- ${\mu'}_{\text{seq}}\big((p, u, i), a, (q, v, j)\big) = \mu_{\text{seq}}(p, a, q)$ if $j = 0$ and there are $w \in [3]^\star$ and $x, y \in [3]$ such that $u = wx, v = wy, y = \min\{x+1, 3\}$, otherwise ${\mu'}_{\text{seq}}\big((p, u, i), a, (q, v, j)\big) = 0$,
- for $m \in \{2, \dots, |Q|\}$ we put
$$\begin{aligned}&{\mu'}^m_{\text{fork}}\big((p, u, i), \{(p_1, u_1, i_1), \dots, (p_m, u_m, i_m)\}\big)\\ =&\mu^m_{\text{fork}}(p, \{p_1, \dots, p_m\})\end{aligned}$$
if $u_1 = \dots = u_m = u1$ and $i_1 = \dots = i_m = 0$, otherwise
$${\mu'}^m_{\text{fork}}\big((p, u, i), \{(p_1, u_1, i_1), \dots, (p_m, u_m, i_m)\}\big) = 0,$$
- for $m \in \{2, \dots, |Q|\}$ we put
$$\begin{aligned}&{\mu'}^m_{\text{join}}\big(\{(q_1, v_1, j_1), \dots, (q_m, v_m, j_m)\}, (q, v, j)\}\big)\\ =&\mu^m_{\text{join}}(\{q_1, \dots, q_m\}, q)\end{aligned}$$
if $j = 1$, and there are $x_1, \dots, x_m, z \in [3]$, $w \in [3]^\star$ such that $v_i = wzx_i$ for $i = 1, \dots, m$, and $v = wz'$ with $z' = \min\{3, z+1\}$, and, moreover, $j_i = 1$ implies $x_i = 3$ for $i = 1, \dots, m$, otherwise
$${\mu'}^m_{\text{join}}\big(\{(q_1, v_1, j_1), \dots, (q_m, v_m, j_m)\}, (q, v, j)\}\big) = 0,$$
- $\lambda'\big((p, u, i)\big) = \begin{cases}\lambda(p) & \text{if } u = 1 \text{ and } i = 0,\\ 0 & \text{otherwise,}\end{cases}$

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

The wba $\mathcal{A}'$ works as follows: In the first component it simulates $\mathcal{A}$. In the second component $\mathcal{A}'$ counts the states of each depth level up to 3 by a stack. Since $\mathcal{A}$ is of bounded depth, the size of this stack can be bounded. Moreover, we keep control of the occurrence of join transitions by the third component. It is 1 for ending states of join transitions, and 0 otherwise. Thus, by a comparison of the second and third component we can control if the automaton is allowed to join again.

We will show that $\mathcal{S}_C(\mathcal{A}') = \mathcal{S}_M(\mathcal{A})$. For this, we need some information about the features of C-runs of $\mathcal{A}'$.

Claim 1. Let $G : p' \xrightarrow{t} q'$ be a C-run of $\mathcal{A}'$ with $p' = (p,u,i)$ and $q' = (q,v,j)$. Then there are $w \in [3]^\star$, $x, y \in [3]$ with $u = wx, v = wy, x \leq y$.

Indeed, for every atomic run the claim holds true by definition of μ'_{seq}. Let $G = G_1 \cdot \ldots \cdot G_m$ $(m \geq 2)$ admit a sequential decomposition. By induction hypothesis, the claim is true for G_1. We proceed by induction along the factors of the sequential product and get Claim 1 for G. Finally, let $G = \|_{p',q'}(G_1,\ldots,G_n)$ with $n \geq 2$ and $u = wx$ for $w \in [3]^\star, x \in [3]$. Let $p'_k = (p_k, u_k, i_k)$ be the ending states of the opening fork transition of G, and $q'_k = (q_k, v_k, j_k)$ the starting states of the terminating join transition of G $(k = 1,\ldots,n)$. By definition of μ'_{fork}, $u_1 = \cdots = u_n = wx1$. Thus, $v_k = wxy_k$ with $y_k \in [3]$ $(k = 1,\ldots,n)$ by induction hypothesis. Considering the definition of μ'_{join}, we conclude $v = wy$ with $y = \min\{3, x+1\} \geq x$. This shows Claim 1.

Claim 2. Let $G : p' \xrightarrow{t} q'$ be a C-run of $\mathcal{A}'$ with $p' = (p,u,i)$, $q' = (q,v,j)$, and $u = w1$ for some $w \in [3]^\star$.

i. If G is parallel and not atomic, then $j = 1$ and $v = w2$.

ii. If G is atomic or sequential then $j = 0$ or $v = w3$.

For part i, we get $j = 1$ immediately from the definition of μ'_{join}. Because $G = \|_{p',q'}(G_1,\ldots,G_n)$ with $n \geq 2$, there is a fork $p' \to \{p'_1,\ldots,p'_n\}$ and a join $\{q'_1,\ldots,q'_n\} \to q$ with $p'_k = (p_k,u_k,i_k)$ and $q'_k = (q_k,v_k,j_k)$ for $k = 1,\ldots,n$. Thus, $i_k = 0$ and $u_k = u1 = w11$ for $k = 1,\ldots,n$ by definition of μ'_{fork}. By Claim 1, we conclude $v_k = w1x_k$ with $x_k \in [3]$ for $k = 1, \ldots, n$. Hence, $v = w2$ by the definition of μ'_{join}.

To show part ii, firstly, let G be atomic. Then $j = 0$ by definition of μ'_{seq}. Now, let $G = G_1 \cdot \ldots \cdot G_m$ $(m \geq 2)$ be the sequential decomposition of G. $G_1 : p' \xrightarrow{t_1} p'_1$ is either atomic or parallel. By the definition of μ'_{seq} and part i, respectively, we get in both cases $u_1 = w2$. Arguing similarly as in the proof of part i for G_2 we get $u_2 = w3$. Proceeding along the sub-runs G_i $(i = 3, \ldots, m)$ we get finally $v = w3$.

Now, we can show that for $\mathcal{A}'$ the C- and M-running mode do not differ from one another.

Claim 3. If $G : (p, u, i) \xrightarrow{t} (q, v, j)$ is a C-run of $\mathcal{A}'$ then G is also an M-run of $\mathcal{A}'$. Hence, $\mathcal{S}_C(\mathcal{A}') = \mathcal{S}_M(\mathcal{A}')$.

If there would be a C-run H of $\mathcal{A}'$ which is not an M-run then there is a sub-run G of H with the same feature such that all proper sub-runs of G are M-runs. This run G has to be a parallel run, *i.e.* $G = \|_{p',q'}(G_1, \ldots, G_n)$ with $n \geq 2$ and $p' = (p, u, i), q' = (q, v, j)$. Moreover, there is an opening fork transition $p' \to \{p'_1, \ldots, p'_n\}$ and a closing join transition $\{q'_1, \ldots, q'_n\} \to q$ of G. Because G is not an M-run, one of the direct sub-runs, say $G_1 : p'_1 \xrightarrow{t_1} q'_1$, has to be a parallel non-atomic run. Let $p'_k = (p_k, u_k, i_k)$ and $q'_k = (q_k, v_k, j_k)$ for $k = 1, \ldots, n$. Then we have $u_1 = u1$ and $i_1 = 0$. By Claim 2, part i, we get $v_1 = u2$ and $j_1 = 1$. Thus, by definition of μ'_{join}, the join transition $\{q'_1, \ldots, q'_m\} \to q'$ cannot exist. This contradicts the containment of $\{q'_1, \ldots, q'_m\} \to q'$ in the C-run G. Hence, G is also an M-run. Now, we get $\mathcal{S}_C(\mathcal{A}') = \mathcal{S}_M(\mathcal{A}')$ immediately and Claim 3 is proven.

Now, $\mathcal{S}_M(\mathcal{A}') = \mathcal{S}_M(\mathcal{A})$ remains to be shown. For this, let $\mathcal{R}_M(\mathcal{A})$ and $\mathcal{R}_M(\mathcal{A}')$ be the sets of all M-runs of $\mathcal{A}$ and $\mathcal{A}'$, respectively. Moreover, let $\mathcal{R}_M^{1,0}(\mathcal{A}')$ be the set of all M-runs of $\mathcal{A}'$ starting in a state $(p, 1, 0)$ with $p \in Q$. We define a mapping $g : \mathcal{R}_M(\mathcal{A}') \to \mathcal{R}_M(\mathcal{A})$ by replacing every state $(p, u, i) \in Q'$ of a run $G' \in \mathcal{R}_M(\mathcal{A}')$ by the state $p \in Q$. This mapping is well defined because of the definition of the three transition functions of $\mathcal{A}'$. Moreover, g preserves the label and the weight of G, as it can be seen easily. Next, we show the following.

Claim 4. Let $G : p \xrightarrow{t} q$ be an M-run of $\mathcal{A}$, $u \in [3]^+$ with $|u| + \mathrm{dp}(G) \leq d+1$, and $i \in \{0,1\}$. Then there are $v \in [3]^+$, $j \in \{0,1\}$, and an M-run $G' : (p,u,i) \xrightarrow{t} (q,v,j)$ in $\mathcal{A}'$ with $g(G') = G$.

Indeed, for any atomic run G the claim is obvious by the definition of μ'_{seq}. Next, let $G = G_1 \cdot \ldots \cdot G_m$ $(m \geq 2)$ be a sequential run. Since $\mathrm{dp}(G_1) \leq \mathrm{dp}(G)$ there is an M-run G'_1 in $\mathcal{A}'$ with $g(G'_1) = G_1$ by induction hypothesis. Suppose $G'_1 : (p_1,u_1,i_1) \to (p_2,u_2,i_2)$. By Claim 1, $|u_2| = |u_1| = |u|$. Hence, $|u_2| + \mathrm{dp}(G_2) \leq d+1$. Hence, there is an M-run G'_2 with $g(G'_2) = G_2$. Repeating this argument along the sequence $G_1, \ldots, G_m$, we get M-runs $G'_1, \ldots, G'_m$ in $\mathcal{A}'$ with $g(G'_k) = G_k$ for $k = 1, \ldots, m$ such that $G' = G'_1 \cdot \ldots \cdot G'_m$ is an M-run of $\mathcal{A}'$. Hence, $g(G') = G$. Finally, let $G = \|_{p,q}(G_1, \ldots, G_n)$ $(n \geq 2)$ be a parallel run. We construct a run G' as follows. The opening fork $p \to \{p_1, \ldots, p_n\}$ is replaced by $(p,u,i) \to \{(p_1, u1, 0), \ldots, (p_n, u1, 0)\}$. This is possible because of $|u| + \mathrm{dp}(G) \leq \mathrm{d}{+}1$. Furthermore, $\mathrm{dp}(G_k) \leq \mathrm{dp}(G) - 1$ for $k = 1, \ldots, n$. Thus, there are $G'_1, \ldots, G'_n \in \mathcal{R}_M(\mathcal{A}')$ with $g(G'_k) = G_k$ for $k = 1, \ldots, n$ by induction hypothesis. Then $G'_k : p'_k \xrightarrow{t_k} q'_k$ with $q'_k = (q_k, v_k, j_k)$ for $k = 1, \ldots, n$. By Claim 1 there are $y_k \in [3]$ such that $v_k = uy_k$ for $k = 1, \ldots, n$. Since G is an M-run of $\mathcal{A}$, neither $G_1, \ldots, G_n$ nor $G'_1, \ldots, G'_n$ are parallel non-atomic runs. Hence, by Claim 2, part ii, $j_k = 0$ or $y_k = 3$ for $k = 1, \ldots, n$. Therefore, there are $v \in [3]^+$ and $j \in \{0,1\}$ such that the join transition $\{(q_1, v_1, j_1), \ldots, (q_n, v_n, j_n)\} \to (q, v, j)$ exists in $\mathcal{A}'$. Hence, we put $G' = \|_{p',q'}(G'_1, \ldots, G'_n)$ and have $g(G') = G$. This proves Claim 4.

Claim 5. $g : \mathcal{R}^{1,0}_M(\mathcal{A}') \longrightarrow \mathcal{R}_M(\mathcal{A})$ is a bijective mapping preserving labels and weights.

We have nothing more to show but g to be bijective. It is clear by the definition of the transition functions of $\mathcal{A}'$ that every run $G' \in \mathcal{R}^{1,0}_M(\mathcal{A}')$ is uniquely determined by $g(G')$ and the state G' is starting with. Hence, $g : \mathcal{R}^{1,0}_M(\mathcal{A}') \longrightarrow \mathcal{R}_M(\mathcal{A})$ is injective. From Claim 4 we can deduce surjectivity immediately.

Claim 5 and the definition of λ' and γ' imply $\mathcal{S}_M(\mathcal{A}') = \mathcal{S}_M(\mathcal{A})$. Thus, we get $\mathcal{S}_C(\mathcal{A}') = \mathcal{S}_M(\mathcal{A})$ by Claim 3. □

Note. The proof of Proposition 7.1 does not work for unbounded depth. In this case, the size of the stack, being the second component of states of $\mathcal{A}$, would grow up to infinity. Hence, we would leave the frame of finite-state automata.

Now, we turn to the question under which conditions C-regular sp-series of bounded width are also M-regular. The next example will show that restrictions on the underlying bisemiring have to be imposed.

Example 7.2. We consider a slightly changed version of the bisemiring of non-proper sp-languages from Example 2.12 with another parallel multiplication. Let $\mathbb{K} = (\mathfrak{P}(\mathrm{SP}^1), \cup, \cdot, \diamond, \emptyset, \{\varepsilon\})$ where $\cdot$ denotes the sequential product of sp-posets lifted to languages. We put for $L, L' \subseteq \mathrm{SP}^1$:

$$L \diamond L' = \big(L \setminus \{\varepsilon\}\big) \parallel \big(L' \setminus \{\varepsilon\}\big)$$

where $\parallel$ denotes the usual parallel product lifted to languages. Obviously, $\diamond$ is associative and commutative. Also distributivity of $\diamond$ over $\cup$ is easy to verify. Note that $\{\varepsilon\} \diamond L = \emptyset$ for all $L \subseteq \mathrm{SP}^1$.

Consider the following sp-series S:

$$(S, t) = \begin{cases} \big\{e\big((d(a \parallel b)d) \parallel c\big)e\big\} & \text{if } t = a \parallel b \parallel c, \\ \emptyset & \text{otherwise.} \end{cases}$$

Obviously, S is of bounded width and is C-regular as the wba of Figure 7.2 shows.

Suppose S is M-recognizable by a wba $\mathcal{A} = (Q, \mu_{\mathrm{seq}}, \mu_{\mathrm{fork}}, \mu_{\mathrm{join}}, \lambda, \gamma)$. Since $(S, a \parallel b \parallel c)$ is a singleton, there has to be an M-run $G : p \xrightarrow{a \parallel b \parallel c} q$ with $\lambda(p) \cdot \mathbf{wgt}(G) \cdot \gamma(q) = \big\{e\big((d(a \parallel b)d) \parallel c\big)e\big\}$. Moreover, G has to start with a fork transition of arity 3 and to end with a join transition of arity 3 because G is an M-run. Thus, there are $k_f, k_j, k_a, k_b, k_c \neq \emptyset$ from $\mathbb{K}$ with $\mathbf{wgt}(G) = k_f \cdot (k_a \diamond k_b \diamond k_c) \cdot k_j$. Keeping in mind initial and final weights, there are $k, l, k_a, k_b, k_c \neq \emptyset$ from $\mathbb{K}$ with

$$\big\{e\big((d(a \parallel b)d) \parallel c\big)e\big\} = k \cdot (k_a \diamond k_b \diamond \cdot k_c) \cdot l.$$

Note that $k_j \neq \{\varepsilon\}$ for $j = a, b, c$ because otherwise $k_a \diamond k_b \diamond k_c = \emptyset$, and, hence, $k \cdot (k_a \diamond k_b \diamond k_c) \cdot l = \emptyset$. Therefore, $k_a \diamond k_b \diamond k_c$ is a parallel sp-poset with at least three parallel factors. The sp-poset $e\big((d(a \parallel b)d) \parallel c\big)e$

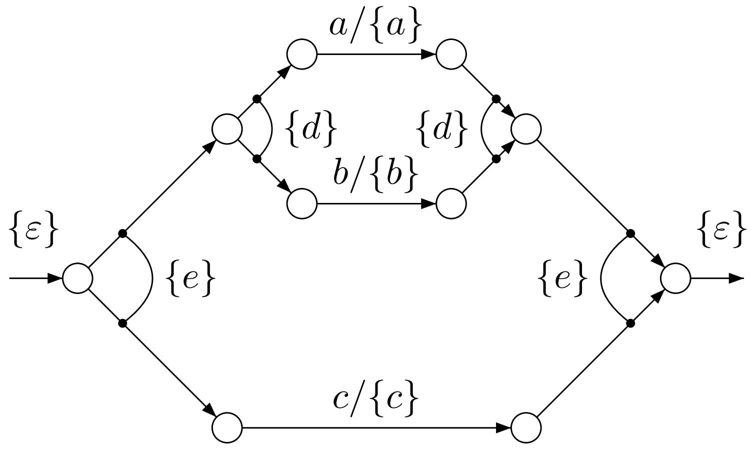


Figure 7.2: A wba C-recognizing S from Example 7.2.

decomposes sequentially in three factors. Since $k_a \diamond k_b \diamond k_c$ yields a parallel sp-poset, we get $k = l = \{e\}$ and $k_a \diamond k_b \diamond k_c = \{(d(a \parallel b)d) \parallel c\}$. But $(d(a \parallel b)d) \parallel c$ allows a parallel decomposition of length 2 only which contradicts $k_a \diamond k_b \diamond k_c$ having a parallel decomposition of length at least 3. Hence, S cannot be M-regular.

The bisemiring used in this example is idempotent. Moreover, we could have chosen the sequential multiplication commutative. This would not have changed our argumentation. Thus, there is an idempotent and commutative bisemiring $\mathbb{K}$ and a C-regular sp-series S over $\mathbb{K}$ of bounded width which is not M-regular.

One problem in Example 7.2 is that the two multiplications of the bisemiring are absolutely independent of each other. Consider Figure 7.2 once again. There, the weight of $a \parallel b \parallel c$ is realized by a parallel multiplication of two runs, the run on $a \parallel b$ and that on c. But the weight of the run on $a \parallel b$ is a sequential product because the weights of the opening fork and join transition have to be considered. So the weights of the "inner" fork and join transitions are nested within a parallel product of the weights of the "outer" run. To construct an M-run on $a \parallel b \parallel c$ with the same weight, these weights have to be summarized in weights for one fork and one join transition, and have to be sequentially multiplied with the parallel product of the weights of the sub-runs. Hence, we may succeed if there is some connection between the sequential and the parallel multiplication. Therefore, we consider in the sequel distributive bisemirings and bisemirings where sequential and parallel multiplication coincide.

Recall that a bisemiring is *distributive* if the sequential multiplication distributes over the parallel one. Firstly, we show that for distributive bisemirings every regular sp-series of bounded width can be recognized by a wba where fork and join transitions have weight 1 only. We call a wba 0-1-*branching* if the fork and join transition functions take values in $\{0, 1\}$ only. Here, 0 is the zero of $\mathbb{K}$ and 1 the sequential unit.

Lemma 7.3. *Let $\mathcal{A}$ be a wba of bounded depth over a distributive bisemiring $\mathbb{K}$. There is a* 0-1-*branching wba $\mathcal{A}'$ of bounded depth with $\mathcal{S}_C(\mathcal{A}') = \mathcal{S}_C(\mathcal{A})$ and $\mathcal{S}_M(\mathcal{A}') = \mathcal{S}_M(\mathcal{A})$.*

Proof. Again, in this proof we speak of runs, behavior etc. instead of C-runs, C-behaviors (M-runs, M-behaviors) because the proof is the same for C- and M-running mode.

Let d be the maximal depth of a run in $\mathcal{A}$. Then we can apply the depth counter construction of Construction 5.2 and get the automaton $\mathcal{A}_{|d}$. From now on, we refer to the second component of the states of $\mathcal{A}_{|d}$ as the *level* of this state, denoted by $\mathrm{lv}(p)$ for a state p. By Construction 5.2, the level function has the following properties:

- if p is initial or final, then $\mathrm{lv}(p) = 0$,
- if $p \to \{p_1, \dots, p_m\}$ is a fork transition, then $\mathrm{lv}(p) + 1 = \mathrm{lv}(p_1) = \dots = \mathrm{lv}(p_m)$, and, dually, for any join transition $\{q_1, \dots, q_m\} \to q$ we have $\mathrm{lv}(q) + 1 = \mathrm{lv}(q_1) = \dots = \mathrm{lv}(q_m)$,
- if $p \xrightarrow{a} q$ is a sequential transition, then $\mathrm{lv}(p) = \mathrm{lv}(q)$.

If a wba can be provided with a level function fulfilling these three properties, we speak of a *level wba*. It is clear that for a run $G : p \xrightarrow{t} q$ in a level wba we have $\mathrm{lv}(p) = \mathrm{lv}(q)$. Since $\mathcal{S}(\mathcal{A}_{|d}) = \mathcal{S}(\mathcal{A})$ by Remark 5.5, $\mathcal{A}$ can be assumed to be a level wba with $\mathrm{lv}(p) \in \{0, \ldots, d\}$ for all $p \in Q$. Moreover, we speak of a *fork transition* $p \to \{p_1, \ldots, p_m\}$ *of level* i if $\mathrm{lv}(p) = i - 1$. Dually, $\{q_1, \ldots, q_m\} \to q$ is a *join transition of level* i if $\mathrm{lv}(q) = i - 1$.

The proof will be as follows:

- Firstly, we construct another level wba $\mathcal{A}_1$ with $\mathcal{S}(\mathcal{A}_1) = \mathcal{S}(\mathcal{A})$ such that all fork and join transitions of level 1 have weight 1.[1]

- We continue the construction along the levels of the wba, *i.e.* build automata $\mathcal{A}_i$ with the same behavior as $\mathcal{A}$ such that all fork and join transitions up to level i $(i = 1, \ldots, d)$ have weight 1 only.

- The wba $\mathcal{A}_d$ is the 0-1-branching wba we searched for.

We start with the construction of $\mathcal{A}_1$. From now on, f denotes a fork transition of $\mathcal{A}$, and j a join transition of $\mathcal{A}$. Let F_1 be the set of all fork transitions of level 1 in $\mathcal{A}$, and J_1 the set of all join transitions of level 1 in $\mathcal{A}$. We define the state set Q_1 as follows: Firstly, let $Q \subseteq Q_1$. Furthermore, if $f : p \to \{p_1, \ldots, p_m\}$ is a fork transition of level 1, then we define new states $p_1^f, \ldots, p_m^f \in Q_1$. Dually, for any join transition $j : \{q_1, \ldots, q_m\} \to q$ of level 1 we put $q_1^j, \ldots, q_m^j \in Q_1$. Since there are a finite number of fork and join transitions only, the set Q_1 is finite. Additionally, we define the new level function lv_1 by

$$\mathrm{lv}_1(p') = \begin{cases} \mathrm{lv}(p') & \text{if } p' \in Q, \\ \mathrm{lv}(p) & \text{if } p' = p^f \text{ for } p \in Q, f \in F_1, \\ \mathrm{lv}(p) & \text{if } p' = p^j \text{ for } p \in Q, j \in J_1. \end{cases}$$

Now we "move" the weights of the fork and join transitions of level 1 in $\mathcal{A}$ to the next higher level. For this, we define the transition functions of $\mathcal{A}_1$

[1] Remember that we do not speak of a "transition" if the weight is 0.

as follows:

$$\mu 1_{\mathrm{seq}}(p',a,q')$$

$$= \begin{cases} \mu_{\mathrm{seq}}(p',a,q') & \text{if } p',q' \in Q, \\ \mu_{\mathrm{fork}}(f) \circ \mu_{\mathrm{seq}}(p,a,q') & \text{if } p' = p^f \text{ and } q' \in Q \\ & (p \in Q, f \in F_1), \\ \mu_{\mathrm{seq}}(p',a,q) \circ \mu_{\mathrm{join}}(j) & \text{if } p' \in Q \text{ and } q' = q^j \\ & (q \in Q, j \in J_1), \\ \mu_{\mathrm{fork}}(f) \circ \mu_{\mathrm{seq}}(p,a,q) \circ \mu_{\mathrm{join}}(j) & \text{if } p' = p^f \text{ and } q' = q^j \\ & (p,q \in Q, f \in F_1, j \in J_1). \end{cases}$$

For $f : p \to \{p_1,\ldots,p_m\}$ in $\mathcal{A}$ with $f \in F_1$ we put

$$\mu 1^m_{\mathrm{fork}}(p,\{p^f_1,\ldots,p^f_m\}) = 1.$$

Dually, for $j : \{q_1,\ldots,q_m\} \to q$ in $\mathcal{A}$ with $j \in J_1$ we put

$$\mu 1^m_{\mathrm{join}}(\{q^j_1,\ldots,q^j_m\},q) = 1.$$

Apart from that, there are no other fork and join transitions of level 1 in $\mathcal{A}_1$. All fork and join transitions of $\mathcal{A}$ of higher level than 1 remain fork and join transitions of $\mathcal{A}_1$ with the same weights. Moreover, we define the following new fork transitions: If $p^f \in Q_1$ for some $p \in Q, f \in F_1$ and if $p \to_k \{p_1,\ldots,p_m\}$ is a fork transition of level 2 in $\mathcal{A}$, then we put

$$\mu 1^m_{\mathrm{fork}}(p^f,\{p_1,\ldots,p_m\}) = \mu_{\mathrm{fork}}(f) \circ k\ .$$

Dually, for $q^j \in Q_1$ with $q \in Q, j \in J_1$ and $\{q_1,\ldots,q_m\} \to_l q$ being a join transition of level 2 in $\mathcal{A}$, we put

$$\mu 1^m_{\mathrm{join}}(\{q_1,\ldots,q_m\},q^j) = l \circ \mu_{\mathrm{join}}(j)\ .$$

Finally, the initial and final weight functions remain the same:

$$\lambda_1(p) = \begin{cases} \lambda(p) & \text{if } p \in Q, \\ 0 & \text{otherwise,} \end{cases} \quad \gamma_1(q) = \begin{cases} \gamma(q) & \text{if } q \in Q, \\ 0 & \text{otherwise.} \end{cases}$$

By definition, $\mathcal{A}_1$ together with lv_1 is a level wba. Moreover, all fork and join transitions of level 1 have weight 1 in $\mathcal{A}_1$. Obviously, $\mathcal{A}_1$ is of bounded depth with depth bound d.

Now, we show $\mathcal{S}(\mathcal{A}_1) = \mathcal{S}(\mathcal{A})$. Let $g : \mathcal{R}(\mathcal{A}_1) \to \mathcal{R}(\mathcal{A})$ be as follows: g maps a run G of $\mathcal{A}_1$ to a run $g(G)$ of $\mathcal{A}$ by replacing all state labels p^f and q^j $(p, q \in Q, f \in F_1, j \in J_1)$ in G by p and q, respectively. By definition of $\mathcal{A}_1$, g is well defined. Firstly, we show that every run in $\mathcal{A}$ with weight distinct from 0 has a pre-image under g.

Claim 1. Let $G : p \xrightarrow{t} q$ be a run in $\mathcal{A}$ with $\mathbf{wgt}(G) \neq 0$.

i. There is a run $G' : p \xrightarrow{t} q$ in $\mathcal{A}_1$ with $g(G') = G$.
ii. Suppose there are $f \in F_1, j \in J_1$ such that $p^f, q^j \in Q_1$. Then we have:
    - If $\mu_{\text{fork}}(f) \circ \mathbf{wgt}(G) \neq 0$, then there is a run $G' : p^f \xrightarrow{t} q$ with $g(G') = G$.
    - If $\mathbf{wgt}(G) \circ \mu_{\text{join}}(j) \neq 0$, then there is a run $G' : p \xrightarrow{t} q^j$ with $g(G') = G$.
    - If $\mu_{\text{fork}}(f) \circ \mathbf{wgt}(G) \circ \mu_{\text{join}}(j) \neq 0$, then there is a run $G' : p^f \xrightarrow{t} q^j$ with $g(G') = G$.

We prove Claim 1 by induction on the structure of runs of $\mathcal{A}$. Let the run $G : p \xrightarrow{a} q$ be atomic. Then part (i) is clear by definition of $\mu 1_{\text{seq}}$. Assume $p^f \in Q_1$ for some $f \in F_1$ and $\mu_{\text{fork}}(f) \circ \mathbf{wgt}(G) \neq 0$. Hence, $\mu_{\text{fork}}(f) \circ \mu_{\text{seq}}(p, a, q) = \mu 1_{\text{seq}}(p^f, a, q) \neq 0$. Thus, $G' : p^f \xrightarrow{a} q$ is a run of $\mathcal{A}_1$ with $g(G') = G$. The remainder of part (ii) follows similarly. Now let $G = G_1 \cdot \ldots \cdot G_m$ $(m \geq 2)$ be sequentially decomposed with $G_i : p_i \xrightarrow{t_i} p_{i+1}$ for $i = 1, \ldots, m$ with $p_1 = p$ and $p_{m+1} = q$. Since $\mathbf{wgt}(G) \neq 0$, we get $\mathbf{wgt}(G_i) \neq 0$ for $i = 1, \ldots, m$. By induction hypothesis there are runs $G'_i : p_i \xrightarrow{t_i} p_{i+1}$ with $g(G'_i) = G_i$ for $i = 1, \ldots, m$. Hence, $G' = G'_1 \cdot \ldots \cdot G'_m$ is a run of $\mathcal{A}_1$ with $g(G') = G$. For part (ii) we assume $p^f \in Q_1$ for some $f \in F_1$ and $\mu_{\text{fork}}(f) \circ \mathbf{wgt}(G) \neq 0$. Hence, $\mu_{\text{fork}}(f) \circ \mathbf{wgt}(G_1) \neq 0$. By induction hypothesis there is a run $G'_1 : p^f \xrightarrow{t_1} p_2$ with $g(G'_1) = G_1$. Concluding similarly as for part (i) we get $G' = G'_1 \cdot \ldots \cdot G'_m$ with $g(G') = G$. Again, the other statements of part (ii) follow similarly.

Finally, let $G = \|_{p,q}(G_1, \ldots, G_n)$ have a parallel decomposition with $n \geq 2$. Let G be limited by the fork transtion $f : p \to \{p_1, \ldots, p_n\}$ and the join transition $j : \{q_1, \ldots, q_n\} \to q$, and let $G_i : p_i \xrightarrow{t_i} q_i$ for $i = 1, \ldots, n$. Firstly, we suppose $\mathrm{lv}(p) = \mathrm{lv}(q) = 0$. Hence, $f \in F_1$ and $j \in J_1$. Then we have to show part (i) only, because $p^{f'}$ and $q^{j'}$ cannot exist for any $f' \in F_1, j' \in J_1$. By definition of $\mathcal{A}_1$, there are $p \to_1 \{p_1^f, \ldots, p_n^f\}$ and $\{q_1^j, \ldots, q_n^j\} \to_1 q$. Since

$$\mu_{\mathrm{fork}}(f) \circ \left[\mathbf{wgt}(G_1) \diamond \ldots \diamond \mathbf{wgt}(G_n)\right] \circ \mu_{\mathrm{join}}(j) = \mathbf{wgt}(G) \neq 0$$

we get $\mu_{\mathrm{fork}}(f) \circ \mathbf{wgt}(G_i) \circ \mu_{\mathrm{join}}(j) \neq 0$ for $i = 1, \ldots, n$ by distributivity of $\mathbb{K}$. By induction hypothesis there are $G_i' : p_i^f \xrightarrow{t_i} q_i^j$ with $g(G_i') = G_i$ for $i = 1, \ldots, n$. We put $G' = \|_{p,q}(G_1', \ldots, G_n')$. Obviously, $g(G') = G$. This shows Claim 1 in the first case. Now, we assume $\mathrm{lv}(p)$ and $\mathrm{lv}(q)$ are greater than 0. Thus, f and j are also transitions in $\mathcal{A}_1$. Moreover, $\mathbf{wgt}(G) \neq 0$ implies $\mathbf{wgt}(G_i) \neq 0$ for $i = 1, \ldots, n$. Hence, there are runs $G_i' : p_i \xrightarrow{t_i} q_i$ in $\mathcal{A}_1$ with $g(G_i') = G_i$. Thus, for $G' = \|_{p,q}(G_1', \ldots, G_n')$ we have $g(G') = G$ which shows part (i). Now let $\bar{f} \in F_1$ such that $p^{\bar{f}} \in Q_1$ and $\mu_{\mathrm{fork}}(\bar{f}) \circ \mathbf{wgt}(G) \neq 0$. Hence, $\mu_{\mathrm{fork}}(\bar{f}) \circ \mu_{\mathrm{fork}}(f) \neq 0$. Thus, $p^{\bar{f}} \to \{p_1, \ldots, p_n\}$ is a fork transition in $\mathcal{A}_1$. Let $G_1', \ldots, G_n'$ be as given above. Hence, $G' = \|_{p^{\bar{f}},q}(G_1', \ldots, G_n')$ is a run in $\mathcal{A}_1$ with $g(G') = G$. The remainder of part (ii) follows similarly. Altogether this proves Claim 1.

Let $\mathcal{R}_0(\mathcal{A}_1)$ be the set of all runs of $\mathcal{A}_1$ starting at a state of level 0. Let g_0 be the restriction of g to $\mathcal{R}_0(\mathcal{A}_1)$.

Claim 2. Let $G : p \xrightarrow{t} q$ be a run in $\mathcal{A}$ with $\mathrm{lv}(p) = \mathrm{lv}(q) = 0$ and $\mathbf{wgt}(G) \neq 0$. Then there is a run $G' : p \xrightarrow{t} q$ in $\mathcal{A}_1$ with $g_0(G') = G$.

This follows from Claim 1 immediately because g preserves the level of each state label.

Clearly, both g and g_0 preserve the label of a run.

Claim 3. The mapping $g_0 : \mathcal{R}_0(\mathcal{A}_1) \to \mathcal{R}(\mathcal{A})$ is injective.

Let us assume there are runs $G_1' \neq G_2' \in \mathcal{R}_0(\mathcal{A}_1)$ with $g_0(G_1') = g_0(G_2') = G$. We write $G_i' = (V_i, E_i, \nu_i, \eta_i)$. By definition of g and g_0, G_1' and G_2' can differ in the state labeling only. More precisely, there is a vertex v of G_1' and G_2' such that without loss of generality:

1. $\nu_1(v) = p \in Q$ and $\nu_2(v) = p^f$ for some $f \in F_1$, or
2. $\nu_1(v) = p \in Q$ and $\nu_2(v) = p^j$ for some $j \in J_1$, or
3. $\nu_1(v) = p^f$ and $\nu_2(v) = p^j$ for some $f \in F_1, j \in J_1$.

In the first case, the fork transition f of level 1 is used in $G = g_0(G'_2)$. Hence, there would be a fork transition of level 1 in G'_1 whose one branch terminates in $p \in Q$. But this is impossible by the definition of $\mu 1_{\mathrm{fork}}$. The second case yields a contradiction in a similar way. Now, consider the third case. The state p^f can be reached by some fork transition only. Moreover, a run from $\mathcal{R}_0(\mathcal{A}_1)$ cannot start in p^f because $\mathrm{lv}_1(p^f) \neq 0$. Dually, p^j can be leaved by a join transtion only, and no run from $\mathcal{R}_0(\mathcal{A}_1)$ can terminate in p^j. Hence, in G'_1 and G'_2 there is a node in which a fork transition terminates and a join transition starts. But then G'_1 and G'_2 cannot be runs. Thus, g_0 is injective.

Claim 4. Let $G' : p' \xrightarrow{t} q'$ be a run of $\mathcal{A}_1$.

- If $p', q' \in Q$, then $\mathbf{wgt}(G') = \mathbf{wgt}(g(G'))$.
- If $p' = p^f$ for some $p \in Q, f \in F_1$ and $q' \in Q$, then $\mathbf{wgt}(G') = \mu_{\mathrm{fork}}(f) \circ \mathbf{wgt}(g(G'))$.
- If $p' \in Q$ and $q' = q^j$ for some $q \in Q, j \in J_1$, then $\mathbf{wgt}(G') = \mathbf{wgt}(g(G')) \circ \mu_{\mathrm{join}}(j)$.
- If $p' = p^f$ and $q' = q^j$ for some $p, q \in Q, f \in F_1, j \in J_1$, then $\mathbf{wgt}(G') = \mu_{\mathrm{fork}}(f) \circ \mathbf{wgt}(g(G')) \circ \mu_{\mathrm{join}}(j)$.

For atomic runs Claim 4 is obvious by the definition of $\mu 1_{\mathrm{seq}}$. Let $G' = G'_1 \cdot \ldots \cdot G'_m$ $(m \geq 2)$ admit a sequential decomposition. Note that $G'_i : p'_i \to p'_{i+1}$ is such that $p'_i \in Q$ for $i = 2, \ldots, m$ because the states of Q are the only ones in $\mathcal{A}_1$ that allow in-going sequential and join transitions as well as out-going sequential and fork transitions. By the induction hypothesis and a simple case distinction for G'_1 and G'_m, Claim 4 follows for G'. Now, let $G' = \|_{p',q'}(G'_1, \ldots, G'_n)$ for $n \geq 2$. If $\mathrm{lv}_1(p') = \mathrm{lv}_1(q') \geq 2$, then $p', q' \in Q$. Then G' starts with a fork transition $f : p' \to \{p_1, \ldots, p_n\}$ and ends with a join transition $j : \{q_1, \ldots, q_n\} \to q'$ where $p_i, q_i \in Q$ for $i = 1, \ldots, n$. Hence, we get

$$\begin{aligned}
\mathbf{wgt}(G') &= \mu_{1\,\text{fork}}(f) \circ \left[\mathbf{wgt}(G'_1)\diamond \ldots \diamond \mathbf{wgt}(G'_n)\right] \circ \mu_{1\,\text{join}}(j) \\
&= \mu_{\text{fork}}(f) \circ \left[\mathbf{wgt}(g(G'_1))\diamond \ldots \diamond \mathbf{wgt}(g(G'_n))\right] \circ \mu_{\text{join}}(j) \\
&= \mathbf{wgt}(g(G')).
\end{aligned}$$

Now, suppose $\mathrm{lv}_1(p') = \mathrm{lv}_1(q') = 1$. For $p', q' \in Q$ we proceed the same way as in the last case. Next, let $p' = p^{\bar{f}}$ for some $p \in Q, \bar{f} \in F_1$ and let $q' \in Q$. Then G' starts with a fork transition $f : p^{\bar{f}} \to \{p_1, \ldots, p_n\}$ and ends with a join transition $j : \{q_1, \ldots, q_n\} \to q'$ with $p_i, q_i \in Q$ for $i = 1, \ldots, n$. Hence, we have

$$\begin{aligned}
\mathbf{wgt}(G') &= \mu_{1\,\text{fork}}(f) \circ \left[\mathbf{wgt}(G'_1)\diamond \ldots \diamond \mathbf{wgt}(G'_n)\right] \circ \mu_{1\,\text{join}}(j) \\
&= \mu_{\text{fork}}(\bar{f}) \circ \mu_{\text{fork}}(p, \{p_1, \ldots, p_n\}) \circ \\
&\qquad\qquad \left[\mathbf{wgt}(g(G'_1))\diamond \ldots \diamond \mathbf{wgt}(g(G'_n))\right] \circ \mu_{\text{join}}(j) \\
&= \mu_{\text{fork}}(\bar{f}) \circ \mathbf{wgt}(g(G')).
\end{aligned}$$

The other cases follow similarly. Finally, we assume $\mathrm{lv}_1(p') = \mathrm{lv}_1(q') = 0$. Hence, $p', q' \in Q$ and G' is limited by a fork transition $p' \to_1 \{p_1^f, \ldots, p_n^f\}$ for some $f \in F_1$ and a join transition $\{q_1^j, \ldots, q_n^j\} \to_1 q'$ for some $j \in J_1$. Now, we get by induction hypothesis and distributivity of $\mathbb{K}$:

$$\begin{aligned}
\mathbf{wgt}(G') &= 1 \circ \left[\mathbf{wgt}(G'_1)\diamond \ldots \diamond \mathbf{wgt}(G'_n)\right] \circ 1 \\
&= \Big[\Big(\mu_{\text{fork}}(f) \circ \mathbf{wgt}(g(G'_1)) \circ \mu_{\text{join}}(j)\Big)\diamond \\
&\qquad\qquad \ldots \diamond \Big(\mu_{\text{fork}}(f) \circ \mathbf{wgt}(g(G'_n)) \circ \mu_{\text{join}}(j)\Big)\Big] \\
&= \mu_{\text{fork}}(f) \circ \left[\mathbf{wgt}(g(G'_1))\diamond \ldots \diamond \mathbf{wgt}(g(G'_n))\right] \circ \mu_{\text{join}}(j) \\
&= \mathbf{wgt}(g(G')).
\end{aligned}$$

This proves Claim 4.

Claim 4 implies that g_0 preserves the weight of a run. Let $\mathcal{R}_0^{\neq 0}(\mathcal{A})$ and $\mathcal{R}_0^{\neq 0}(\mathcal{A}_1)$ be the sets of runs of $\mathcal{A}$ and $\mathcal{A}_1$, respectively, starting at a state of level 0 and having weight distinct from zero. By Claims 2, 3, and 4 we get that $g_0 : \mathcal{R}_0^{\neq 0}(\mathcal{A}_1) \to \mathcal{R}_0^{\neq 0}(\mathcal{A})$ is a bijective mapping preserving labels and weights. Considering initial and final weights, we get immediately $\mathcal{S}(\mathcal{A}_1) = \mathcal{S}(\mathcal{A})$.

Now, suppose we have constructed a wba $\mathcal{A}_i$ with $i < d$ and the following properties:

- $\mathcal{A}_i$ is a level automaton of bounded depth with depth bound d,
- all fork and join transitions up to level i have weight 1, and
- $\mathcal{S}(\mathcal{A}_i) = \mathcal{S}(\mathcal{A})$.

We will indicate the construction of $\mathcal{A}_{i+1}$ only. It will be clear from the construction that $\mathcal{A}_{i+1}$ is a level wba of bounded depth, and that all fork and join transitions up to level $i+1$ have weight 1. The proof of $\mathcal{S}(\mathcal{A}_{i+1}) = \mathcal{S}(\mathcal{A}_i)$ follows the same lines as this for $\mathcal{S}(\mathcal{A}_1) = \mathcal{S}(\mathcal{A})$ with some obvious adaptions. $\mathcal{A}_{i+1}$ is constructed as follows: Q_{i+1} contains all states of Q_i. If $f : p \to \{p_1, \ldots, p_m\}$ is a fork transition of level $i+1$ in $\mathcal{A}_i$, then $p_1^f, \ldots, p_m^f \in Q_{i+1}$. Dually, for any join transition $j : \{q_1, \ldots, q_m\} \to q$ of level $i+1$ in $\mathcal{A}_i$, we put $q_1^j, \ldots, q_m^j \in Q_{i+1}$. Let F_i and J_i denote the sets of all fork and join transitions of level i in $\mathcal{A}_i$, respectively. The new level function is given by:

$$\mathrm{lv}_{i+1}(p') = \begin{cases} \mathrm{lv}_i(p') & \text{if } p' \in Q_i, \\ \mathrm{lv}_i(p) & \text{if } p' = p^f \text{ for some } p \in Q_i, f \in F_i, \\ \mathrm{lv}_i(p) & \text{if } p' = p^j \text{ for some } p \in Q_i, j \in J_i. \end{cases}$$

The sequential, fork and join transitions are defined in an analogue manner as for $\mathcal{A}_1$. But now, we transfer the weights of the fork and join transitions of level $i+1$ to the next higher level. Again, the initial and final weights remain unchanged, and all states from $Q_{i+1} \setminus Q_i$ are neither initial nor final.

Now suppose $\mathcal{A}_d$ has been constructed. Since there are no fork and join transitions of higher level than d, $\mathcal{A}_d$ has the following properties:

- $\mathcal{A}_d$ is 0-1-branching,
- $\mathcal{A}_d$ is a level wba of bounded depth, and
- $\mathcal{S}(\mathcal{A}_d) = \mathcal{S}(\mathcal{A})$.

This shows Lemma 7.3. □

Proposition 7.4. *Let S be a C-regular (M-regular) sp-series of bounded width over a distributive bisemiring $\mathbb{K}$. Then S can be C-recognized (M-recognized) by a normalized* 0-1*-branching wba of bounded depth.*

Proof. By Corollary 5.9, S can be recognized by a wba of bounded depth. Moreover, this wba can be normalized due to Proposition 4.4. Note that the normalization construction does not affect the bounded depth property. By Lemma 7.3, this normalized wba can be turned into a 0-1-branching wba of bounded depth. A short analysis of the proof of Lemma 7.3 shows that the construction given there preserves normalization. □

From now on, we call a bisemiring $\mathbb{K} = (K, \oplus, \circ, \circ, 0, 1)$ where sequential and parallel multiplication coincide a *doubled semiring*. Note that any doubled semiring is a commutative bisemiring.

Proposition 7.5. *Let $\mathbb{K}$ be a distributive bisemiring or a doubled semiring. Then every sequential-rational sp-series over $\mathbb{K}$ is M-regular.*

Proof. Obviously, all monomials are M-regular. By Propositions 4.1, 4.2, 4.12, and 4.15 M-regular sp-series are closed under sum, scalar products, sequential product, and sequential iteration. It remains to show the closure under parallel product.

Firstly, let $\mathbb{K}$ be a distributive bisemiring. Let S_1 and S_2 be two M-regular sp-series of bounded width over $\mathbb{K}$. By Proposition 7.4, both S_1 and S_2 can be M-recognized by normalized 0-1-branching wba $\mathcal{A}_1$ and $\mathcal{A}_2$ of bounded depth. We denote the unique initial state of $\mathcal{A}_i$ by $\mathfrak{i}_i$ and the unique final one by $\mathfrak{f}_i$ for $i = 1, 2$. We define the wba $\mathcal{A}$ as follows:

- $Q = Q_1 \mathbin{\dot\cup} Q_2 \mathbin{\dot\cup} \{\mathfrak{i}, \mathfrak{f}\}$,
- $\mu_{\text{seq}}(p, a, q) = \begin{cases} \mu_{1_{\text{seq}}}(p, a, q) & \text{if } p, q \in Q_1, \\ \mu_{2_{\text{seq}}}(p, a, q) & \text{if } p, q \in Q_2, \\ 0 & \text{otherwise,} \end{cases}$

- for $m \in \{2, \ldots, |Q|\}$ we put:[2]

$$\mu^m_{\text{fork}}(p, \{p_1, \ldots, p_m\})$$

$$= \begin{cases} {\mu_1}^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } p, p_1, \ldots, p_m \in Q_1, \\ {\mu_2}^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } p, p_1, \ldots, p_m \in Q_2, \\ 1 & \text{if } p = \mathfrak{i} \text{ and } \exists j \in \{2, \ldots, m-2\} : \\ & {\mu_1}^j_{\text{fork}}(\mathfrak{i}_1, \{p_1, \ldots, p_j\}) = 1, \\ & {\mu_2}^{m-j}_{\text{fork}}(\mathfrak{i}_2, \{p_{j+1}, \ldots, p_m\}) = 1, \\ 1 & \text{if } p = \mathfrak{i}, p_1 = \mathfrak{i}_1 \text{ and} \\ & {\mu_2}^{m-1}_{\text{fork}}(\mathfrak{i}_2, \{p_2, \ldots, p_m\}) = 1, \\ 1 & \text{if } p = \mathfrak{i}, p_m = \mathfrak{i}_2 \text{ and} \\ & {\mu_1}^{m-1}_{\text{fork}}(\mathfrak{i}_1, \{p_1, \ldots, p_{m-1}\}) = 1, \\ 1 & \text{if } p = \mathfrak{i}, \{p_1, \ldots, p_m\} = \{\mathfrak{i}_1, \mathfrak{i}_2\}, \\ 0 & \text{otherwise,} \end{cases}$$

- for $m \in \{2, \ldots, |Q|\}$ we put:

$$\mu^m_{\text{join}}(\{q_1, \ldots, q_m\}, q)$$

$$= \begin{cases} {\mu_1}^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) & \text{if } q, q_1, \ldots, q_m \in Q_1, \\ {\mu_2}^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) & \text{if } q, q_1, \ldots, q_m \in Q_2, \\ 1 & \text{if } q = \mathfrak{f} \text{ and } \exists j \in \{2, \ldots, m-2\} : \\ & {\mu_1}^j_{\text{join}}(\{q_1, \ldots, q_j\}, \mathfrak{f}_1) = 1, \\ & {\mu_2}^{m-j}_{\text{join}}(\{q_{j+1}, \ldots, q_m\}, \mathfrak{f}_2) = 1, \\ 1 & \text{if } q = \mathfrak{f}, q_1 = \mathfrak{f}_1 \text{ and} \\ & {\mu_2}^{m-1}_{\text{join}}(\{q_2, \ldots, q_m\}, \mathfrak{f}_2) = 1, \\ 1 & \text{if } q = \mathfrak{f}, q_m = \mathfrak{f}_2 \text{ and} \\ & {\mu_1}^{m-1}_{\text{join}}(\{q_1, \ldots, q_{m-1}\}, \mathfrak{f}_1) = 1, \\ 1 & \text{if } q = \mathfrak{f}, \{q_1, \ldots, q_m\} = \{\mathfrak{f}_1, \mathfrak{f}_2\}, \\ 0 & \text{otherwise,} \end{cases}$$

[2] In this proof, we write a subset P of $Q_1 \dot\cup Q_2$ in such a way that all states from Q_1 in P proceed the states from Q_2 in P.

- $\lambda(p) = \begin{cases} 1 & \text{if } p = \mathfrak{i}, \\ 0 & \text{otherwise,} \end{cases} \qquad \gamma(q) = \begin{cases} 1 & \text{if } q = \mathfrak{f}, \\ 0 & \text{otherwise.} \end{cases}$

We will show that $\mathcal{S}_M(\mathcal{A}) = \mathcal{S}_M(\mathcal{A}_1) \parallel \mathcal{S}_M(\mathcal{A}_2)$. For this, we define a mapping from pairs of successful runs in $\mathcal{A}_1$ and $\mathcal{A}_2$ to successful runs of $\mathcal{A}$. Let $\mathcal{R}_M^{sc}(\mathcal{A})$ be the set of all M-runs in $\mathcal{A}$ from $\mathfrak{i}$ to $\mathfrak{f}$, and, similarly, $\mathcal{R}_M^{sc}(\mathcal{A}_i)$ the set of all runs in $\mathcal{A}_i$ from $\mathfrak{i}_i$ to $\mathfrak{f}_i$ for $i = 1, 2$. Now, we define the mapping $g : \mathcal{R}_M^{sc}(\mathcal{A}_1) \times \mathcal{R}_M^{sc}(\mathcal{A}_2) \to \mathcal{R}_M^{sc}(\mathcal{A})$. Let $G_1 \in \mathcal{R}_M^{sc}(\mathcal{A}_1), G_2 \in \mathcal{R}_M^{sc}(\mathcal{A}_2)$, and let $G_1 = \parallel_{\mathfrak{i}_1,\mathfrak{f}_1}(G_{11}, \ldots, G_{1m})$ and $G_2 = \parallel_{\mathfrak{i}_2,\mathfrak{f}_2}(G_{21}, \ldots, G_{2n})$ with $m, n \geq 1$ be the parallel decomposition of G_1 and G_2, respectively. Note that for $m = 1$ the run G_1 is sequential, and likewise for $n = 1$ and G_2. Furthermore, since G_1 and G_2 are M-runs, all the runs G_{1i} for $i = 1, \ldots, m$ and G_{2j} for $j = 1, \ldots, n$ are sequential runs. We define $g(G_1, G_2)$ by

$$g(G_1, G_2) = \parallel_{\mathfrak{i},\mathfrak{f}}(G_{11}, \ldots, G_{1m}, G_{21}, \ldots, G_{2n}).$$

By definition of $\mathcal{A}$, $g(G_1, G_2)$ is an M-run of $\mathcal{A}$. Moreover, we get immediately

$$\mathbf{lab}[g(G_1, G_2)] = \mathbf{lab}(G_1) \parallel \mathbf{lab}(G_2).$$

Claim 1. For $G_1 \in \mathcal{R}_M^{sc}(\mathcal{A}_1), G_2 \in \mathcal{R}_M^{sc}(\mathcal{A}_2)$ we have

$$\mathbf{wgt}[g(G_1, G_2)] = \mathbf{wgt}(G_1) \diamond \mathbf{wgt}(G_2).$$

Indeed, for G_1 and G_2 being sequential the claim is obvious. Now, assume G_1 is sequential but G_2 is not. Hence, $G_2 = \parallel_{\mathfrak{i}_2,\mathfrak{f}_2}(G_{21}, \ldots, G_{2n})$ with $n \geq 2$ and $G_{2i} : p_{2i} \to q_{2i}$ for $i = 1, \ldots, n$. Then we get

$$\begin{aligned} \mathbf{wgt}(g(G_1, G_2)) &= \mu_{\text{fork}}^{n+1}(\mathfrak{i}, \{\mathfrak{i}_1, p_{21}, \ldots, p_{2n}\}) \circ \Big[\mathbf{wgt}(G_1) \diamond \mathbf{wgt}(G_{21}) \diamond \\ &\qquad \ldots \diamond \mathbf{wgt}(G_{2n})\Big] \circ \mu_{\text{join}}^{n+1}(\{\mathfrak{f}_1, q_{21}, \ldots, q_{2n}\}, \mathfrak{f}) \\ &= \mathbf{wgt}(G_1) \diamond \Big[\mathbf{wgt}(G_{21}) \diamond \ldots \diamond \mathbf{wgt}(G_{2n})\Big] \\ &= \mathbf{wgt}(G_1) \diamond \mathbf{wgt}(G_2) \end{aligned}$$

because $\mathcal{A}_2$ is 0-1-branching. For the other cases we conclude similarly. This proves Claim 1.

Obviously, g is injective. Now, let $G \in \mathcal{R}_M^{sc}(\mathcal{A})$. By definition of $\mathcal{A}$, G cannot be atomic. Suppose $G = G_1 \cdot \ldots \cdot G_m$. Since the only out-going transitions of $\mathfrak{i}$ are fork transitions, G_1 starts with a fork transition $\mathfrak{i} \to \{p_1, \dots, p_m\}$. Then there are $i \neq j \in \{1, \dots, m\}$ with $p_i \in Q_1$ and $p_j \in Q_2$. It is an easy exercise to show that each run of $\mathcal{A}$ starting in a state of Q_1 also ends in a state of Q_1, and, dually, for Q_2. Thus, G_1 terminates with a join transition joining states both from Q_1 and Q_2. By definition of μ_{join}, G_1 ends in state $\mathfrak{f}$. Since $\mathfrak{f}$ admits no out-going transitions, we get $G = G_1$, *i.e.* G is of the form $G = \|_{\mathfrak{i},\mathfrak{f}}(H_1, \dots, H_n)$ with $H_1, \dots, H_j \in \mathcal{R}^{sc}_M(\mathcal{A}_1)$ and $H_{j+1}, \dots, H_n \in \mathcal{R}^{sc}_M(\mathcal{A}_2)$ for some $j \in \{1, \dots, n\}$. We put $G_1 = \|_{\mathfrak{i}_1,\mathfrak{f}_1}(H_1, \dots, H_j)$ and $G_2 = \|_{\mathfrak{i}_2,\mathfrak{f}_2}(H_{j+1}, \dots, H_n)$. These are well defined M-runs of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. Moreover, we get $g(G_1, G_2) = G$. This shows surjectivity of g, and, hence, bijectivity. Now we get for any $t \in \mathrm{SP}$:

$$\begin{aligned}
&\Big(\mathcal{S}_M(\mathcal{A}_1) \,\|\, \mathcal{S}_M(\mathcal{A}_2), t\Big) \\
&= \bigoplus_{\substack{(t_1,t_2): \\ t=t_1 \,\|\, t_2}} \Big[(\mathcal{S}_M(\mathcal{A}_1), t_1) \diamond (\mathcal{S}_M(\mathcal{A}_2), t_2)\Big] \\
&= \bigoplus_{\substack{(t_1,t_2): \\ t=t_1 \,\|\, t_2}} \Bigg[\Big(\bigoplus_{\substack{G_1 \in \mathcal{R}^{sc}_M(\mathcal{A}_1) \\ \mathbf{lab}(G_1)=t_1}} \mathbf{wgt}(G_1)\Big) \diamond \Big(\bigoplus_{\substack{G_2 \in \mathcal{R}^{sc}_M(\mathcal{A}_2) \\ \mathbf{lab}(G_2)=t_2}} \mathbf{wgt}(G_2)\Big)\Bigg] \\
&= \bigoplus_{\substack{(t_1,t_2): \\ t=t_1 \,\|\, t_2}} \Bigg[\bigoplus_{\substack{G_1 : \mathfrak{i}_1 \xrightarrow{t_1} \mathfrak{f}_1 \\ G_2 : \mathfrak{i}_2 \xrightarrow{t_2} \mathfrak{f}_2}} \Big(\mathbf{wgt}(G_1) \diamond \mathbf{wgt}(G_2)\Big)\Bigg] \\
&= \bigoplus_{\substack{G_1 : \mathfrak{i}_1 \xrightarrow{t_1} \mathfrak{f}_1 \\ G_2 : \mathfrak{i}_2 \xrightarrow{t_2} \mathfrak{f}_2 \\ t=t_1 \,\|\, t_2}} \Big(\mathbf{wgt}(G_1) \diamond \mathbf{wgt}(G_2)\Big) \\
&= \bigoplus_{\substack{G_1 : \mathfrak{i}_1 \xrightarrow{t_1} \mathfrak{f}_1 \\ G_2 : \mathfrak{i}_2 \xrightarrow{t_2} \mathfrak{f}_2 \\ t=t_1 \,\|\, t_2}} \mathbf{wgt}(g(G_1, G_2))
\end{aligned}$$

$$= \bigoplus_{G:\mathfrak{i} \xrightarrow{t} \mathfrak{f}} \mathbf{wgt}(G)$$

$$= (\mathcal{S}_M(\mathcal{A}), t).$$

This shows the closure of M-regular sp-series of bounded width over a distributive bisemiring under parallel product.

Now, let $\mathbb{K}$ be a doubled semiring, and let S_1, S_2 be two M-regular sp-series of bounded width over $\mathbb{K}$. Then both S_1 and S_2 can be M-recognized by normalized wba $\mathcal{A}_1$ and $\mathcal{A}_2$ of bounded depth. Again, let $\mathfrak{i}_i$ and $\mathfrak{f}_i$ denote the unique initial and final state of $\mathcal{A}_i$ for $i = 1, 2$. The wba $\mathcal{A}$ M-recognizing $S_1 \parallel S_2$ is constructed almost in the same way as for distributive bisemirings. Only the weights of the fork and join transitions differ. With $Q = Q_1 \,\dot\cup\, Q_2 \,\dot\cup\, \{\mathfrak{i}, \mathfrak{f}\}$ we put:

- for $m \in \{2, \ldots, |Q|\}$

$$\mu^m_{\text{fork}}(p, \{p_1, \ldots, p_m\})$$

$$= \begin{cases} \mu_1{}^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } p, p_1, \ldots, p_m \in Q_1, \\ \mu_2{}^m_{\text{fork}}(p, \{p_1, \ldots, p_m\}) & \text{if } p, p_1, \ldots, p_m \in Q_2, \\ \mu_1{}^j_{\text{fork}}(\mathfrak{i}_1, \{p_1, \ldots, p_j\}) \circ & \\ \quad \mu_2{}^{m-j}_{\text{fork}}(\mathfrak{i}_2, \{p_{j+1}, \ldots, p_m\}) & \text{if } p = \mathfrak{i} \text{ and } \exists j \in \{2, \ldots, m-2\} : \\ & p_1, \ldots, p_j \in Q_1, \\ & p_{j+1}, \ldots, p_m \in Q_2, \\ \mu_1{}^{m-1}_{\text{fork}}(\mathfrak{i}_1, \{p_1, \ldots, p_{m-1}\}) & \text{if } p = \mathfrak{i}, p_1, \ldots, p_{m-1} \in Q_1, \\ & p_m = \mathfrak{i}_2, \\ \mu_2{}^{m-1}_{\text{fork}}(\mathfrak{i}_2, \{p_2, \ldots, p_m\}) & \text{if } p = \mathfrak{i}, p_1 = \mathfrak{i}_1, \\ & p_2, \ldots, p_m \in Q_2, \\ 1 & \text{if } p = \mathfrak{i}, \{p_1, \ldots, p_m\} = \{\mathfrak{i}_1, \mathfrak{i}_2\}, \\ 0 & \text{otherwise,} \end{cases}$$

- for $m \in \{2, \ldots, |Q|\}$

$$
\begin{aligned}
&\mu^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) \\
&= \begin{cases}
\mu 1^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) & \text{if } q, q_1, \ldots, q_m \in Q_1, \\
\mu 2^m_{\text{join}}(\{q_1, \ldots, q_m\}, q) & \text{if } q, q_1, \ldots, q_m \in Q_2, \\
\mu 1^j_{\text{join}}(\{q_1, \ldots, q_j\}, \mathfrak{f}_1) \circ & \\
\quad \mu 2^{m-j}_{\text{join}}(\{q_{j+1}, \ldots, q_m\}, \mathfrak{f}_2) & \text{if } q = \mathfrak{f} \text{ and } \exists j \in \{2, \ldots, m-2\} : \\
& q_1, \ldots, q_j \in Q_1, \\
& q_{j+1}, \ldots, q_m \in Q_2, \\
\mu 1^{m-1}_{\text{join}}(\{q_1, \ldots, q_{m-1}\}, \mathfrak{f}_1) & \text{if } q = \mathfrak{f}, q_1, \ldots, q_{m-1} \in Q_1, \\
& q_m = \mathfrak{f}_2, \\
\mu 2^{m-1}_{\text{join}}(\{q_2, \ldots, q_m\}, \mathfrak{f}_2) & \text{if } q = \mathfrak{f}, q_1 = \mathfrak{f}_1, \\
& q_2, \ldots, q_m \in Q_2, \\
1 & \text{if } q = \mathfrak{f}, \{q_1, \ldots, q_m\} = \{\mathfrak{f}_1, \mathfrak{f}_2\}, \\
0 & \text{otherwise.}
\end{cases}
\end{aligned}
$$

The proof of $\mathcal{S}_M(\mathcal{A}) = \mathcal{S}_M(\mathcal{A}_1) \,\|\, \mathcal{S}_M(\mathcal{A}_2)$ follows the same lines as that for distributive bisemirings. But now, the proof of the equivalent to Claim 1 does not use 0-1-branching automata but the coincidence of sequential and parallel multiplication. We omit the details.

Hence, M-regular sp-series of bounded width over doubled semirings are closed under parallel product. $\square$

Now we can state the main result about C- and M-regularity.

Theorem 7.6. *Let $\mathbb{K}$ be an arbitrary bisemiring and $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ an sp-series of bounded width.*

1. *If S is M-regular, then S is also C-regular.*
2. *If $\mathbb{K}$ is distributive or a doubled semiring and S is C-regular, then S is M-regular.*

Proof. Since S is M-regular and of bounded width, by Corollary 5.9, it can be M-recognized by a wba of bounded depth. Then S is C-regular by

Proposition 7.1. Vice versa, every C-regular sp-series S of bounded width is sequential-rational by Theorem 6.2. If $\mathbb{K}$ is distributive or a doubled semiring, this implies M-regularity of S by Proposition 7.5. □

For distributive bisemirings and doubled semirings we get another result in the spirit of Kleene and Schützenberger.

Theorem 7.7. *Let $\mathbb{K}$ be distributive or a doubled semiring. The following are equivalent for $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$:*

1. *S is sequential-rational.*
2. *S is M-regular and of bounded width.*
3. *S is M-recognized by a wba of bounded depth.*

Proof. Apply Theorems 6.2 and 7.6, and Corollary 5.9. □

8 The Hadamard Product for SP-Series

The generalization of the intersection of languages to series is the Hadamard product. It is no Cauchy product where both supports and weights are multiplied but the usual pointwise product of two functions. Within the scope of sp-series it is possible to define a sequential and parallel Hadamard product because the bisemiring admits two multiplications. Here, we are interested in the closure of regular sp-series under Hadamard product. In a wba the sequential multiplication "dominates" the parallel one. More precisely, any run of a wba is first of all a sequence of sequential, fork and join transitions, and several sets of sub-runs where the sub-runs of every set are composed in parallel. Moreover, we will see that nice results in this chapter can be achieved for doubled semirings only. Thus, it does not matter which of both products we consider. Therefore, we study the sequential Hadamard product only.

Let S and T be two sp-series over Σ and $\mathbb{K}$. The *sequential Hadamard product* $S \odot T$ of S and T is defined by:

$$S \odot T = \sum_{t \in \mathrm{SP}} \Big((S,t) \circ (T,t) \Big)\, t.$$

Hence, $\mathrm{supp}(S \odot T) \subseteq \mathrm{supp}(S) \cap \mathrm{supp}(T)$. From now on, we will speak just of the Hadamard product instead of the sequential Hadamard product.

For word series over a semiring there are two main ways to prove the closure of regular series under Hadamard product. Either the product automaton simulating two automata simultaneously is constructed [SS78], or one exploits algebraic characterizations using finitely generated semimodules [BR88]. We have to follow the first way which has the advantage of being constructive. Such a construction should be based on the usual product automaton from formal language theory. There the state set is taken as the direct product of the two state sets of the automata involved. However, this construction cannot be generalized straightforwardly for the C-running mode as the following example shows.

Example 8.1. We work with the Boolean bisemiring $\mathbb{B}$, *i.e.* within the setting of sp-languages. The language $L = \{a \parallel b \parallel c\}$ is C-recognized both by $\mathcal{A}_1$ and $\mathcal{A}_2$ as shown in Figure 8.1. In a product automaton $\mathcal{A}$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ there would be the following sequential transitions: $(p_2, q_4) \xrightarrow{a} (p_8, q_6)$, $(p_4, q_5) \xrightarrow{b} (p_6, q_7)$, and $(p_5, q_3) \xrightarrow{c} (p_7, q_9)$. Moreover, (p_1, q_1) would be the unique initial state. How should we define the new fork transitions starting in (p_1, q_1)? From a permissive point of view, we could allow two fork transitions $(p_1, q_1) \to \{(p_2, q_2), (p_3, q_3)\}$ and $(p_1, q_1) \to \{(p_2, q_3), (p_3, q_2)\}$ in $\mathcal{A}$. Similarly, there may be fork transitions $(p_3, q_2) \to \{(p_4, q_4), (p_5, q_5)\}$ and $(p_3, q_2) \to \{(p_4, q_5), (p_5, q_4)\}$. But even with these four fork transitions we cannot fork into the states (p_2, q_4) and (p_5, q_3) where two of the three sequential transitions start. Therefore, the language C-recognized by $\mathcal{A}$ would be empty in contradiction to $L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = \{a \parallel b \parallel c\}$.

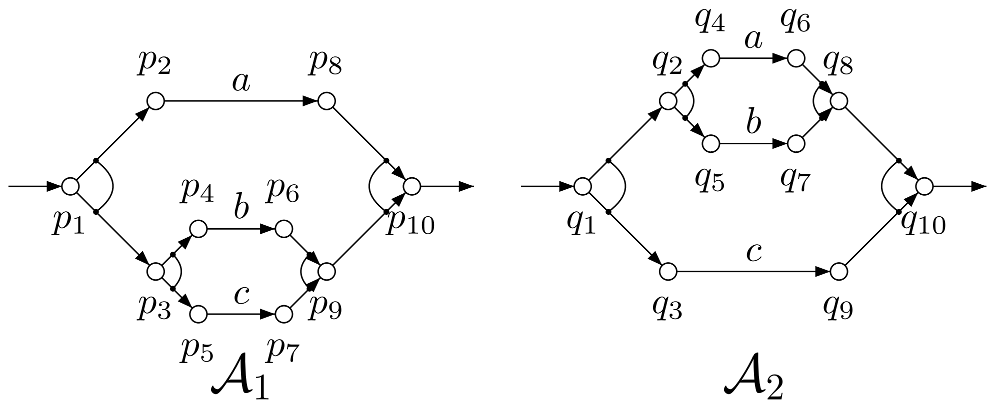


Figure 8.1: Two wba C-recognizing $\{a \parallel b \parallel c\}$.

The main problem in Example 8.1 is the different branching structure of the two automata. Whereas $\mathcal{A}_1$ forks firstly into runs with labels a and $(b \parallel c)$, $\mathcal{A}_2$ forks firstly into runs with labels $(a \parallel b)$ and c. These two different realizations cannot be captured by a local definition of new fork and join transitions. However, the situation would be different if we consider M-regular sp-series because in an M-run a parallel product of length n is realized by fork and join transitions of arity n. Therefore, we concentrate on the closure of M-regular sp-series under Hadamard product.

The Hadamard product of two regular sp-series can be interpreted as follows: For an arbitrary sp-poset t the automata $\mathcal{A}_1$ and $\mathcal{A}_2$ run on t one after the other and their results, *i.e.* the weights, are multiplied sequentially. To show that the Hadamard product of the behaviors of $\mathcal{A}_1$ and $\mathcal{A}_2$ is regular, the two runs of the automata on t have to be fused to one run on t in a product automaton $\mathcal{A}$ still to be defined. For this, the weights both of $\mathcal{A}_1$ and $\mathcal{A}_2$ have to be fused on the local level of sequential transitions. Therefore, even for word series over semirings, the sequential multiplication has to be commutative. Furthermore, now we have to shift the weights over the fork and join transitions away to the different depth levels of the automaton. Hence, it is not surprising that we have to impose heavy restrictions on the underlying bisemiring to get a positive result.

8.1 The Hadamard Product for Doubled Semirings

Firstly, we give the construction of the product automaton for an arbitrary bisemiring $\mathbb{K}$.

Construction 8.2. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two wba over Σ and $\mathbb{K}$ with $\mathcal{A}_i = (Q_i, \mu_{i_{\text{seq}}}, \mu_{i_{\text{fork}}}, \mu_{i_{\text{join}}}, \lambda_i, \gamma_i)$ for $i = 1, 2$. We define the *product wba* $\mathcal{A}_1 \times \mathcal{A}_2 = (Q, \mu_{\text{seq}}, \mu_{\text{fork}}, \mu_{\text{join}}, \lambda, \gamma)$ of $\mathcal{A}_1$ and $\mathcal{A}_2$ as follows:[1]

- $Q = Q_1 \times Q_2$,
- $\mu_{\text{seq}}\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, a, \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right) = \mu_{1_{\text{seq}}}(p^1, a, q^1) \circ \mu_{2_{\text{seq}}}(p^2, a, q^2)$,
- for $m \in \{2, \dots, |Q|\}$:[2]

$$\mu^m_{\text{fork}}\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, \left\{\begin{pmatrix} p^1_1 \\ p^2_1 \end{pmatrix}, \dots, \begin{pmatrix} p^1_m \\ p^2_m \end{pmatrix}\right\}\right) = \mu{1}^m_{\text{fork}}(p^1, \{p^1_1, \dots, p^1_m\}) \circ \mu{2}^m_{\text{fork}}(p^2, \{p^2_1, \dots, p^2_m\}),$$

[1] Here, we indicate by an upper index to which wba a state belongs to, *e.g.* $p^1 \in Q_1$.
[2] For $|Q| \geq m > |Q_1|$ we assume $\mu_{1_{\text{fork}}}$ to be identically zero, and similarly for $\mu_{2_{\text{fork}}}$. This assumption also applies to the join transition functions.

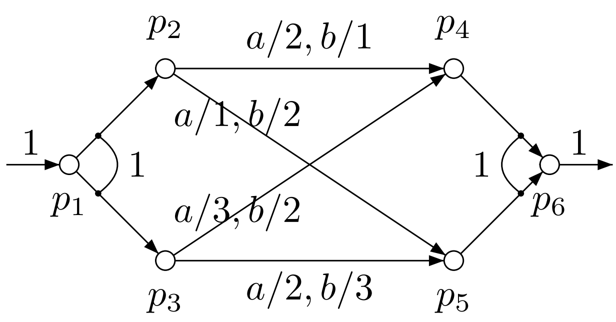


Figure 8.2: A wba $\mathcal{A}$ with auto-concurrency.

- for $m \in \{2, \ldots, |Q|\}$:

$$\mu_{\text{join}}^{m}\left(\left\{\binom{p_1^1}{p_1^2}, \ldots, \binom{p_m^1}{p_m^2}\right\}, \binom{p^1}{p^2}\right)$$
$$= \mu 1_{\text{join}}^{m}(\{p_1^1, \ldots, p_m^1\}, p^1) \circ \mu 2_{\text{join}}^{m}(\{p_1^2, \ldots, p_m^2\}, p^2),$$

- $\lambda \binom{p^1}{p^2} = \lambda_1(p^1) \circ \lambda_2(p^2)$, and $\gamma \binom{p^1}{p^2} = \gamma_1(p^1) \circ \gamma_2(p^2)$.

Note that a doubled semiring $\mathbb{K} = (K, \oplus, \circ, \circ, 0, 1)$ is always a commutative bisemiring. However, the next example shows that the product wba construction does not M-recognize the Hadamard product of two M-regular sp-series over arbitrary doubled semirings. We will see that auto-concurrency causes severe problems.

Example 8.3. Let $\mathcal{A}$ be the wba of Figure 8.2. The weights are taken from the doubled semiring of natural numbers with the usual addition and multiplication. Then $\mathcal{S}_M(\mathcal{A}) = 16(a \parallel b) + 7(a \parallel a) + 7(b \parallel b)$. Hence, $\mathcal{S}_M(\mathcal{A}) \odot \mathcal{S}_M(\mathcal{A}) = 256(a \parallel b) + 49(a \parallel a) + 49(b \parallel b)$. If we apply Construction 8.2 of the product automaton, we get the wba $\mathcal{A} \times \mathcal{A}$ with fork transitions $\binom{p_1}{p_1} \to_1 \{\binom{p_2}{p_2}, \binom{p_3}{p_3}\}$ and $\binom{p_1}{p_1} \to_1 \{\binom{p_2}{p_3}, \binom{p_3}{p_2}\}$, and join transitions $\{\binom{p_4}{p_4}, \binom{p_5}{p_5}\} \to_1 \binom{p_6}{p_6}$, $\{\binom{p_4}{p_5}, \binom{p_5}{p_4}\} \to_1 \binom{p_6}{p_6}$. Therefore, we get for $a \parallel a$ eight successful runs in $\mathcal{A} \times \mathcal{A}$ as shown in Figure 8.3. Hence, $(\mathcal{S}_M(\mathcal{A} \times \mathcal{A}), a \parallel a) = 98 \neq 49$. The problem arises from the following situation: In $\mathcal{A}$ there are only two successful runs on $a \parallel a$. Suppose $\mathcal{A}^{\odot}$

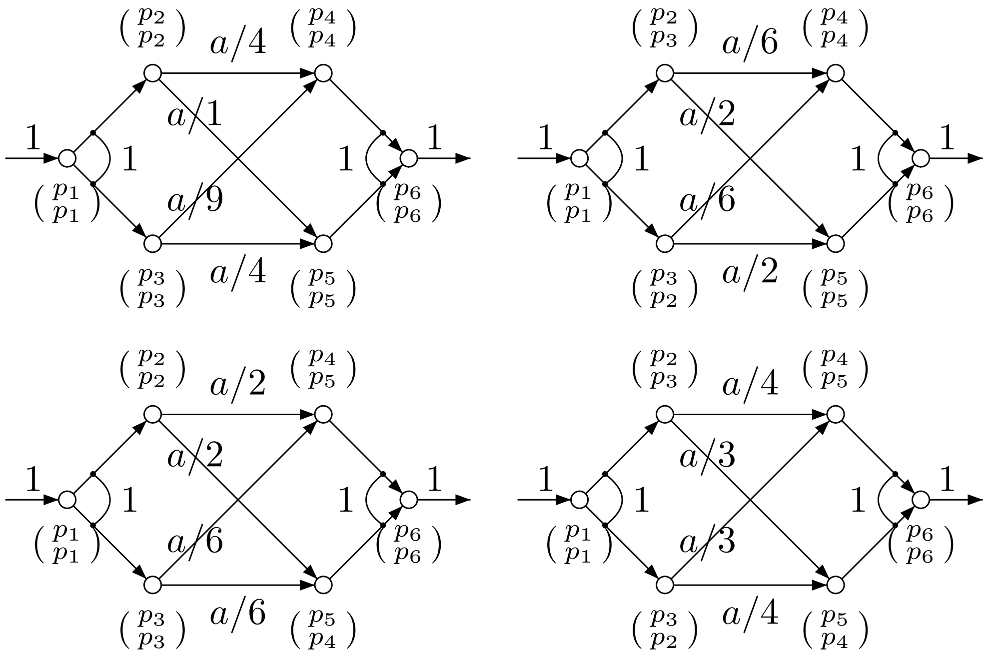


Figure 8.3: Eight successful runs on $a \parallel a$ in the product automaton $\mathcal{A} \times \mathcal{A}$.

is a wba M-recognizing $\mathcal{S}_M(\mathcal{A}) \odot \mathcal{S}_M(\mathcal{A})$. Hence, $\mathcal{A}^{\odot}$ should have only four instead of eight successful runs on $a \parallel a$. But on the other hand, the sp-poset $a \parallel b$ has four successful runs in $\mathcal{A}$, and should, therefore, have 16 successful runs in $\mathcal{A}^{\odot}$. But the number of successful runs has to be realized by the branching structure of that automaton. If we would allow the fork $\binom{p_1}{p_1} \to_1 \{\binom{p_2}{p_2}, \binom{p_3}{p_3}\}$ only (or, dually, the join $\{\binom{p_4}{p_4}, \binom{p_5}{p_5}\} \to_1 \binom{p_6}{p_6}$ only), we would get four successful runs on $a \parallel b$ only and would make a mistake for the weight of this sp-poset. Construction 8.2 realizes the right number of successful runs for the sp-poset $a \parallel b$ which has no auto-concurrency. As a consequence we get too many runs for sp-posets with auto-concurrency as we have seen above.

The next lemma gives a description of the behavior of the product automaton over doubled semirings. Let $n \in \mathbb{N}$. We abbreviate a sum of n equal elements $k \in \mathbb{K}$ as follows:

$$n.k = \underbrace{k \oplus k \oplus \ldots \oplus k}_{n}.$$

Lemma 8.4. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two wba over the doubled semiring $\mathbb{K}$, and let $t \in \mathrm{SP}$. Then there is $c(t) \in \mathbb{N}^{>0}$, depending on t only, such that:*

$$\Big(\mathcal{S}_M(\mathcal{A}_1 \times \mathcal{A}_2), t\Big) = c(t).\Big[\Big(\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2), t\Big)\Big].$$

Proof. Let $\mathcal{A}_i$ be a wba with $\mathcal{S}_M(\mathcal{A}_i) = S_i$ for $i = 1, 2$. Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ be the product wba as given in Construction 8.2. Firstly, we prove the following:

Claim 1. For every $t \in \mathrm{SP}$ there is a $c(t) \in \mathbb{N}^{>0}$ such that for every $p^1, q^1 \in Q_1$, $p^2, q^2 \in Q_2$ we have:

$$\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) = c(t).\Big[\mathbf{wgt}^1_M(p^1, t, q^1) \circ \mathbf{wgt}^2_M(p^2, t, q^2)\Big].$$

We proceed by structural induction on $t \in \mathrm{SP}$. Indeed, for $t = a \in \Sigma$ we get Claim 1 immediately by the definition of $\mathcal{A}$. Especially, $c(a) = 1$. Now, let $t = t_1 \cdot \ldots \cdot t_m$ ($m \geq 2$) be a sequential sp-poset with its maximal sequential decomposition. Then we get:

$$\begin{aligned}\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) &= \bigoplus_{G:\binom{p^1}{p^2} \xrightarrow{t} \binom{q^1}{q^2}} \mathbf{wgt}(G) \\ &= \bigoplus_{\substack{G_1,\ldots,G_m: \\ G = G_1 \cdot \ldots \cdot G_m, \\ \mathbf{lab}(G_i) = t_i}} \mathbf{wgt}(G_1) \circ \ldots \circ \mathbf{wgt}(G_m)\end{aligned}$$

and then by commutativity of $\oplus$ and distributivity of $\circ$ over $\oplus$

$$= \bigoplus_{\left[\binom{r^1_1}{r^2_1}, \ldots, \binom{r^1_{m-1}}{r^2_{m-1}}\right] \in Q^{m-1}} \mathbf{wgt}_M\left(\binom{p^1}{p^2}, t_1, \binom{r^1_1}{r^2_1}\right) \circ \ldots$$

$$\circ\, \mathbf{wgt}_M\left(\binom{r^1_{m-1}}{r^2_{m-1}}, t_m, \binom{q^1}{q^2}\right)$$

and now by induction hypothesis

$$= \bigoplus_{\left[\binom{r_1^1}{r_1^2},\ldots,\binom{r_{m-1}^1}{r_{m-1}^2}\right]\in Q^{m-1}} \Big(c(t_1).[\mathbf{wgt}_M^1(p^1,t_1,r_1^1)\circ \mathbf{wgt}_M^2(p^2,t_1,r_1^2)]$$

$$\circ \ldots \circ c(t_m).[\mathbf{wgt}_M^1(r_{m-1}^1,t_m,q^1)\circ \mathbf{wgt}_M^2(r_{m-1}^2,t_m,q^2)]\Big)$$

and then by distributivity of $\circ$ over $\oplus$ and commutativity of $\circ$ [3]

$$= \bigoplus_{\left[\binom{r_1^1}{r_1^2},\ldots,\binom{r_{m-1}^1}{r_{m-1}^2}\right]\in Q^{m-1}} \big(c(t_1)c(t_2)\ldots c(t_m)\big).\Big(\mathbf{wgt}_M^1(p^1,t_1,r_1^1)\circ \ldots$$

$$\circ\, \mathbf{wgt}_M^1(r_{m-1}^1,t_m,q^1)\circ \mathbf{wgt}_M^2(p^2,t_1,r_1^2)\circ \ldots \circ \mathbf{wgt}_M^2(r_{m-1}^2,t_m,q^2)\Big)$$

and, finally, by distributivity of $\circ$ over $\oplus$ and commutativity of $\oplus$

$$=\big(c(t_1)c(t_2)\ldots c(t_m)\big).$$

$$\Bigg[\Bigg(\bigoplus_{(r_1^1,\ldots,r_{m-1}^1)\in Q_1^{m-1}} [\mathbf{wgt}_M^1(p^1,t_1,r_1^1)\circ \ldots \circ \mathbf{wgt}_M^1(r_{m-1}^1,t_m,q^1)]\Bigg)$$

$$\circ \Bigg(\bigoplus_{(r_1^2,\ldots,r_{m-1}^2)\in Q_2^{m-1}} [\mathbf{wgt}_M^2(p^2,t_1,r_1^2)\circ \ldots \circ \mathbf{wgt}_M^2(r_{m-1}^2,t_m,q^2)]\Bigg)\Bigg]$$

$$=\big(c(t_1)c(t_2)\ldots c(t_m)\big).\Big(\mathbf{wgt}_M^1(p^1,t,q^1)\circ \mathbf{wgt}_M^2(p^2,t,q^2)\Big)$$

which shows the claim for a sequential sp-poset t. Note, that $c(t) = c(t_1)c(t_2)\ldots c(t_m)$ depends on t only because $c(t_i)$ depends on t_i only for $i = 1,\ldots,m$.

[3] In the next equation, $c(t_1)c(t_2)\ldots c(t_m)$ denotes the usual product of $c(t_1), c(t_2), \ldots, c(t_m)$ in $\mathbb{N}$.

Finally, we consider a parallel sp-poset $t = t_1 \parallel \ldots \parallel t_m$ ($m \geq 2$) given in its maximal parallel decomposition. Since the wba considered are in M-running mode, we know that a run on t starts with a fork transition of arity m and ends with a join transition of arity m. Let S_m denote the permutation group of m elements. We calculate the weight of t from p^1 to $q^1 \in Q_1$ in $\mathcal{A}_1$: Firstly, we add over all possible fork transitions $p^1 \to \{r^1_1, \ldots, r^1_m\}$ starting in p^1 and all join transitions $\{s^1_1, \ldots, s^1_m\} \to q^1$ terminating in q^1. If the fork and join transition are fixed we have to add up over all possible executions of $t = t_1 \parallel \ldots \parallel t_m$ between the state sets $\{r^1_1, \ldots, r^1_m\}$ and $\{s^1_1, \ldots, s^1_m\}$. These executions are determined by the starting and ending state of a run on t_i for all $i = 1, \ldots, m$. The number of executions depends on the "rate of auto-concurrency" of t, *i.e.* the classes of equal factors in $t_1 \parallel \ldots \parallel t_m$. Let $\ker(t) \subseteq S_m$ be the subgroup of S_m consisting of all those permutations $\sigma \in S_m$ such that $t_i = t_{\sigma(i)}$ for all $i = 1, \ldots, m$. Let $\pi_1, \pi_2, \varrho_1, \varrho_2 \in S_m$. On one hand, we fix the starting state of the runs on t_i by $r^1_{\pi_1(i)}$ and the ending state by $s^1_{\varrho_1(i)}$ for $i = 1, \ldots, m$. This yields the execution $X_1(t) = \{(r^1_{\pi_1(i)}, t_i, s^1_{\varrho_1(i)}) \mid i = 1, \ldots, m\}$. On the other hand, we get another execution $X_2(t) = \{(r^1_{\pi_2(i)}, t_i, s^1_{\varrho_2(i)}) \mid i = 1, \ldots, m\}$ which makes use of the permutations π_2 and ϱ_2. Now, we have:

Claim 2. $X_1(t) = X_2(t)$ iff there is a $\sigma \in \ker(t)$ with $\pi_2 = \pi_1 \circ \sigma$ and $\varrho_2 = \varrho_1 \circ \sigma$.[4]

Indeed, if $\pi_2 = \pi_1 \circ \sigma$ and $\varrho_2 = \varrho_1 \circ \sigma$ for some $\sigma \in \ker(t)$ then $(r^1_{\pi_1(i)}, t_i, s^1_{\varrho_1(i)}) = (r^1_{\pi_2(j)}, t_j, s^1_{\varrho_2(j)})$ for $i = \sigma(j)$ and $j = 1, \ldots, m$. Vice versa, $X_1(t) = X_2(t)$ implies the existence of some $\sigma \in S_m$ with $(r^1_{\pi_2(i)}, t_i, s^1_{\varrho_2(i)}) = (r^1_{\pi_1(\sigma(i))}, t_{\sigma(i)}, s^1_{\varrho_1(\sigma(i))})$ for $i = 1, \ldots, m$. Hence, $\pi_2 = \pi_1 \circ \sigma$ and $\varrho_2 = \varrho_1 \circ \sigma$. Moreover, $t_i = t_{\sigma(i)}$ for $i = 1, \ldots, m$. Therefore, $\sigma \in \ker(t)$.

Let U_m be a system of representatives for the left cosets $\pi \ker(t)$ where $\pi \in S_m$. We put $X_{\pi,\varrho}(t) = \{(r^1_{\pi(i)}, t_i, s^1_{\varrho(i)}) \mid i = 1, \ldots, m\}$. Let $\pi_1, \pi_2 \in S_m$ and $\varrho_1, \varrho_2 \in U_m$. Then $X_{\pi_1,\varrho_1}(t) = X_{\pi_2,\varrho_2}(t)$ iff $\pi_1 = \pi_2$ and $\varrho_1 = \varrho_2$. Indeed, by Claim 2 $X_{\pi_1,\varrho_1}(t) = X_{\pi_2,\varrho_2}(t)$ implies $\pi_2 = \pi_1 \circ \sigma$ and $\varrho_2 = \varrho_1 \circ \sigma$ for some $\sigma \in \ker(t)$. Hence, ϱ_1 and ϱ_2 would be in the same left coset of $\ker(t)$. Thus, $\varrho_1 = \varrho_2$. This implies $\pi_1 = \pi_2$.

[4] Here, $\pi_1 \circ \sigma$ means: firstly, apply σ and then π_1.

Moreover, for $\pi, \varrho \in S_m$ there are $\pi' \in S_m, \varrho' \in U_m$ with $X_{\pi,\varrho}(t) = X_{\pi',\varrho'}(t)$. We just choose ϱ' as the representative from U_m which is in the same coset as ϱ. Hence, there is $\sigma \in \ker(t)$ with $\varrho' = \varrho \circ \sigma \in U_m$, and then we put $\pi' = \pi \circ \sigma$. Hence, to obtain the weight of $t = t_1 \parallel \ldots \parallel t_m$ we have to add over all $\pi \in S_m$ and all $\varrho \in U_m$. Thus, we get:

$$\begin{aligned}
&\mathbf{wgt}^1_M(p^1, t, q^1) \\
&= \bigoplus_{\substack{\{r^1_1,\ldots,r^1_m\} \in \mathfrak{P}_m(Q_1) \\ \{s^1_1,\ldots,s^1_m\} \in \mathfrak{P}_m(Q_1) \\ \pi \in S_m, \varrho \in U_m}} \Big[\mu 1^m_{\mathrm{fork}}(p^1, \{r^1_1, \ldots, r^1_m\}) \circ \mathbf{wgt}^1_M(r^1_{\pi(1)}, t_1, s^1_{\varrho(1)}) \circ \\
&\qquad \ldots \circ \mathbf{wgt}^1_M(r^1_{\pi(m)}, t_m, s^1_{\varrho(m)}) \circ \mu 1^m_{\mathrm{join}}(\{s^1_1, \ldots, s^1_m\}, q^1)\Big]
\end{aligned} \tag{8.1}$$

To calculate $\mathbf{wgt}^2_M(p^2, t, q^2)$ we do the same as for wba $\mathcal{A}_1$. But this time we let both permutations range over all S_m. But then, by Claim 2, every execution of $t = t_1 \parallel \ldots \parallel t_m$ is generated $|\ker(t)|$ times. Let $e(t) = |\ker(t)|$. Then we get for the wba $\mathcal{A}_2$:

$$\begin{aligned}
&e(t).\, \mathbf{wgt}^2_M(p^2, t, q^2) \\
&= \bigoplus_{\substack{\{r^2_1,\ldots,r^2_m\} \in \mathfrak{P}_m(Q_2) \\ \{s^2_1,\ldots,s^2_m\} \in \mathfrak{P}_m(Q_2) \\ \tau, \eta \in S_m}} \Big[\mu 2^m_{\mathrm{fork}}(p^2, \{r^2_1, \ldots, r^2_m\}) \circ \mathbf{wgt}^2_M(r^2_{\tau(1)}, t_1, s^2_{\eta(1)}) \circ \\
&\qquad \ldots \circ \mathbf{wgt}^2_M(r^2_{\tau(m)}, t_m, s^2_{\eta(m)}) \circ \mu 2^m_{\mathrm{join}}(\{s^2_1, \ldots, s^2_m\}, q^2)\Big]
\end{aligned} \tag{8.2}$$

For $\mathcal{A}$ we argue as for $\mathcal{A}_1$ and get:

$$\begin{aligned}
&\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \\
&= \bigoplus_{\left\{\binom{p_1^1}{p_1^2},\ldots,\binom{p_m^1}{p_m^2}\right\}\in\mathfrak{P}_m(Q)} \quad \bigoplus_{\left\{\binom{q_1^1}{q_1^2},\ldots,\binom{q_m^1}{q_m^2}\right\}\in\mathfrak{P}_m(Q)} \quad \bigoplus_{\alpha\in S_m,\beta\in U_m} \\
&\left[\mu_{\text{fork}}^m\left(\binom{p^1}{p^2}, \left\{\binom{p_1^1}{p_1^2}, \ldots, \binom{p_m^1}{p_m^2}\right\}\right) \circ \mathbf{wgt}_M\left(\binom{p_{\alpha(1)}^1}{p_{\alpha(1)}^2}, t_1, \binom{q_{\beta(1)}^1}{q_{\beta(1)}^2}\right) \circ \ldots \right. \\
&\left. \circ\, \mathbf{wgt}_M\left(\binom{p_{\alpha(m)}^1}{p_{\alpha(m)}^2}, t_m, \binom{q_{\beta(m)}^1}{q_{\beta(m)}^2}\right) \circ \mu_{\text{join}}^m\left(\left\{\binom{q_1^1}{q_1^2}, \ldots, \binom{q_m^1}{q_m^2}\right\}, \binom{q^1}{q^2}\right)\right]
\end{aligned} \tag{8.3}$$

In the next step, we want to add over pairs of m-sets[5] instead over m-sets of pairs. In Equation (8.3), the starting states of the runs on t_i are $\binom{p_{\alpha(i)}^1}{p_{\alpha(i)}^2}$ where $\alpha \in S_m$. There, firstly we choose a set of m pairs and consider for such a choice all possible permutations. Note, that we can restrict the choice of the m pairs as follows: the sets $\{p_1^1, \ldots, p_m^1\}$ and $\{p_1^2, \ldots, p_m^2\}$ should both contain m mutually different elements because otherwise

$$\mu_{\text{fork}}^m\left(\binom{p^1}{p^2}, \left\{\binom{p_1^1}{p_1^2}, \ldots, \binom{p_m^1}{p_m^2}\right\}\right) = 0.$$

Hence, we choose the m-sets $\{p_1^1, \ldots, p_m^1\}$ and $\{p_1^2, \ldots, p_m^2\}$, fix the pairs, and permute them. We have $m!$ possibilities to fix the pairs because we can fix the order of the states of the first component, and consider all possible permutations of the states of the second component. Therefore, we may permute each m-set independently of the other one by permutations $\pi, \tau \in S_m$ to get all pairs in any order. The same is true for the ending states $\binom{q_{\beta(m)}^1}{q_{\beta(m)}^2}$. But this time $\beta \in U_m$. To obtain all pairs in any order

[5] By an m-set we denote a set of cardinality m.

determined by a permutation of U_m we do the following: we permute the states of the first component by all $\varrho \in U_m$ (which gives the order), and the states of the second component by all $\eta \in S_m$ (which gives all pairs). Hence, we can continue Equation (8.3) as follows:

$$= \bigoplus_{\substack{(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}),\\ (\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\})\\ \in\mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \bigoplus_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \Bigg[\mu 1^m_{\mathrm{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) \circ \mu 2^m_{\mathrm{fork}}(p^2,\{r_1^2,\ldots,r_m^2\})$$
$$\circ\, \mathbf{wgt}_M\left(\begin{pmatrix} r^1_{\pi(1)} \\ r^2_{\tau(1)}\end{pmatrix}, t_1, \begin{pmatrix} s^1_{\varrho(1)} \\ s^2_{\eta(1)}\end{pmatrix}\right) \circ \ldots \circ \mathbf{wgt}_M\left(\begin{pmatrix} r^1_{\pi(m)} \\ r^2_{\tau(m)}\end{pmatrix}, t_m, \begin{pmatrix} s^1_{\varrho(m)} \\ s^2_{\eta(m)}\end{pmatrix}\right)$$
$$\circ \mu 1^m_{\mathrm{join}}(\{s_1^1,\ldots,s_m^1\},q^1) \circ \mu 2^m_{\mathrm{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Bigg] \tag{8.4}$$

By induction hypothesis we get next:

$$= \bigoplus_{\substack{(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}),\\ (\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\})\\ \in\mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \bigoplus_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \Bigg[\mu 1^m_{\mathrm{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) \circ \mu 2^m_{\mathrm{fork}}(p^2,\{r_1^2,\ldots,r_m^2\})$$
$$\circ \left[c(t_1).[\mathbf{wgt}^1_M(r^1_{\pi(1)},t_1,s^1_{\varrho(1)}) \circ \mathbf{wgt}^2_M(r^2_{\tau(1)},t_1,s^2_{\eta(1)})]\right]$$
$$\vdots$$
$$\circ \left[c(t_m).[\mathbf{wgt}^1_M(r^1_{\pi(m)},t_m,s^1_{\varrho(m)}) \circ \mathbf{wgt}^2_M(r^2_{\tau(m)},t_m,s^2_{\eta(m)})]\right]$$
$$\circ \mu 1^m_{\mathrm{join}}(\{s_1^1,\ldots,s_m^1\},q^1) \circ \mu 2^m_{\mathrm{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Bigg] \tag{8.5}$$

Now, we apply distributivity of $\circ$ over $\oplus$:

$$
\begin{aligned}
= & \bigoplus_{\substack{\left(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}\right),\\ \left(\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\}\right)\\ \in \mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \bigoplus_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \\
& \Big[\mu_1{}^m_{\mathrm{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) \circ \mu_2{}^m_{\mathrm{fork}}(p^2,\{r_1^2,\ldots,r_m^2\}) \\
& \qquad \circ \big(c(t_1)c(t_2)\ldots c(t_m)\big). \\
& \qquad \big[\mathbf{wgt}_M^1(r^1_{\pi(1)},t_1,s^1_{\varrho(1)}) \circ \mathbf{wgt}_M^2(r^2_{\tau(1)},t_1,s^2_{\eta(1)}) \\
& \qquad \vdots \\
& \qquad \circ \mathbf{wgt}_M^1(r^1_{\pi(m)},t_m,s^1_{\varrho(m)}) \circ \mathbf{wgt}_M^2(r^2_{\tau(m)},t_m,s^2_{\eta(m)})\big] \\
& \qquad \circ \mu_1{}^m_{\mathrm{join}}(\{s_1^1,\ldots,s_m^1\},q^1) \circ \mu_2{}^m_{\mathrm{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Big]
\end{aligned} \tag{8.6}
$$

By commutativity of $\circ$ and distributivity of $\circ$ over $\oplus$ we get:

$$
\begin{aligned}
= & \big(c(t_1)c(t_2)\ldots c(t_m)\big).\Bigg[\bigoplus_{\substack{\left(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}\right),\\ \left(\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\}\right)\\ \in \mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \bigoplus_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \\
& \Big[\mu_1{}^m_{\mathrm{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) \circ \mu_2{}^m_{\mathrm{fork}}(p^2,\{r_1^2,\ldots,r_m^2\}) \\
& \qquad \circ \mathbf{wgt}_M^1(r^1_{\pi(1)},t_1,s^1_{\varrho(1)}) \circ \ldots \circ \mathbf{wgt}_M^1(r^1_{\pi(m)},t_m,s^1_{\varrho(m)}) \\
& \qquad \circ \mathbf{wgt}_M^2(r^2_{\tau(1)},t_1,s^2_{\eta(1)}) \circ \ldots \circ \mathbf{wgt}_M^2(r^2_{\tau(m)},t_m,s^2_{\eta(m)}) \\
& \qquad \circ \mu_1{}^m_{\mathrm{join}}(\{s_1^1,\ldots,s_m^1\},q^1) \circ \mu_2{}^m_{\mathrm{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Big]\Bigg]
\end{aligned} \tag{8.7}
$$

Finally, we apply Equations (8.1) and (8.2) and get:

$$
= \big(c(t_1)c(t_2)\ldots c(t_m)\big).\Big[\mathbf{wgt}_M^1(p^1,t,q^1) \circ \big(e(t).\,\mathbf{wgt}_M^2(p^2,t,q^2)\big)\Big] \tag{8.8}
$$

Summarizing Equations (8.3) to (8.8) and applying distributivity of $\circ$ over $\oplus$, we get

$$\begin{aligned}&\mathbf{wgt}_M\left(\begin{pmatrix}p^1\\p^2\end{pmatrix}, t, \begin{pmatrix}q^1\\q^2\end{pmatrix}\right)\\&\quad = \big(c(t_1)c(t_2)\dots c(t_m)e(t)\big).\Big[\mathbf{wgt}^1_M(p^1,t,q^1)\circ\mathbf{wgt}^2_M(p^2,t,q^2)\Big]\end{aligned} \tag{8.9}$$

Since $e(t)$ depends on t and $c(t_i)$ on t_i only, the factor $c(t_1)c(t_2)\dots c(t_m)e(t)$ depends on t only. This proves Claim 1.

Now we can calculate the weight of $t \in \mathrm{SP}$ in $\mathcal{A}$:

$$\begin{aligned}
(\mathcal{S}(\mathcal{A}),t) &= \bigoplus_{\begin{pmatrix}p^1\\p^2\end{pmatrix},\begin{pmatrix}q^1\\q^2\end{pmatrix}\in Q} \lambda\begin{pmatrix}p^1\\p^2\end{pmatrix}\circ\mathbf{wgt}_M\left(\begin{pmatrix}p^1\\p^2\end{pmatrix}, t, \begin{pmatrix}q^1\\q^2\end{pmatrix}\right)\circ\gamma\begin{pmatrix}q^1\\q^2\end{pmatrix}\\
&= \bigoplus_{\begin{pmatrix}p^1\\p^2\end{pmatrix},\begin{pmatrix}q^1\\q^2\end{pmatrix}\in Q} \lambda\begin{pmatrix}p^1\\p^2\end{pmatrix}\circ\Big(c(t).\big[\mathbf{wgt}^1_M(p^1,t,q^1)\\
&\qquad\qquad \circ\,\mathbf{wgt}^2_M(p^2,t,q^2)\big]\Big)\circ\gamma\begin{pmatrix}q^1\\q^2\end{pmatrix}\\
&= c(t).\Big[\bigoplus_{\substack{p^1,q^1\in Q_1\\p^2,q^2\in Q_2}}\lambda_1(p^1)\circ\mathbf{wgt}^1_M(p^1,t,q^1)\circ\gamma_1(q^1)\\
&\qquad\qquad \circ\lambda_2(p^2)\circ\mathbf{wgt}^2_M(p^2,t,q^2)\circ\gamma_2(q^2)\Big]\\
&= c(t).\Big[\Big[\bigoplus_{p^1,q^1\in Q_1}\lambda_1(p^1)\circ\mathbf{wgt}^1_M(p^1,t,q^1)\circ\gamma_1(q^1)\Big]\\
&\qquad\qquad \circ\Big[\bigoplus_{p^2,q^2\in Q_2}\lambda_2(p^2)\circ\mathbf{wgt}^2_M(p^2,t,q^2)\circ\gamma_2(q^2)\Big]\Big]\\
&= c(t).\Big[\big(\mathcal{S}_M(\mathcal{A}_1),t\big)\circ\big(\mathcal{S}_M(\mathcal{A}_2),t\big)\Big]\\
&= c(t).\Big[\big(\mathcal{S}_M(\mathcal{A}_1)\odot\mathcal{S}_M(\mathcal{A}_2),t\big)\Big]
\end{aligned}$$

This shows our assertion. $\square$

Remark 8.5. The factor $c(t)$ for $t \in \mathrm{SP}$ used in Lemma 8.4 can be computed by the decomposition of t. More precisely, we have:

- If $t = a \in \Sigma$ then $c(t) = 1$.
- If $t = t_1 \cdot \ldots \cdot t_m$ $(m \geq 2)$ is sequentially decomposed then $c(t) = c(t_1) \ldots c(t_m)$.
- Let $t = t_1 \parallel \ldots \parallel t_m$ $(m \geq 2)$ admit a parallel decomposition. Moreover, let $e_1, \ldots, e_n \in \mathbb{N}^{>0}$ with $n \leq m$ be such that for each $i = 1, \ldots, n$ e_i factors of t are equal but no other factor is equal to one of them. Then

$$c(t) = (e_1! \ldots e_n!) c(t_1) \ldots c(t_m).$$

Example 8.6. Let $t = a\Big(b \parallel b \parallel \big(a(a \parallel b)\big)\Big)\Big(a \parallel a \parallel b \parallel b\Big)$. Then we get:

$$\begin{aligned} c(t) =& c(a)\, c\Big(b \parallel b \parallel \big(a(a \parallel b)\big)\Big)\, c(a \parallel a \parallel b \parallel b) \\ =& 2!\, c(b)\, c(b)\, c\big(a(a \parallel b)\big)\, 2!\, 2!\, c(a)\, c(a)\, c(b)\, c(b) \\ =& (2!)^3\, c(a)\, c(a \parallel b) = (2!)^3\, c(a)\, c(b) \\ =& 2^3 = 8. \end{aligned}$$

Lemma 8.4 has an immediate consequence for the supports of M-regular sp-series. We call a bisemiring $\mathbb{K}$ *torsion-free* if $(K, \oplus)$ is torsion-free, *i.e.* $n.k \neq 0$ for all $n \geq 1$ and $k \in K$ with $k \neq 0$.

Corollary 8.7. *Let $\mathbb{K}$ be a torsion-free doubled semiring, and let $\mathcal{A}_1, \mathcal{A}_2$ be two wba over $\mathbb{K}$. Then*

$$\operatorname{supp}\big[\mathcal{S}_M(\mathcal{A}_1 \times \mathcal{A}_2)\big] = \operatorname{supp}\big[\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2)\big].$$

Corollary 8.7 is especially true for all zero-sum-free doubled semirings.

The problem of a multiplication of successful runs for sp-posets with auto-concurrency can be compensated by idempotency of the underlying doubled semiring. Remember that $\mathbb{K}$ is idempotent if $k \oplus k = k$ for all $k \in \mathbb{K}$.

Theorem 8.8. *Let S_1 and S_2 be two M-regular sp-series over an idempotent doubled semiring $\mathbb{K}$. Then the Hadamard product $S_1 \odot S_2$ is again M-regular.*

Proof. Let $\mathcal{A}_i$ be a wba M-recognizing S_i for $i = 1, 2$. Moreover, let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. By Lemma 8.4, for every $t \in \mathrm{SP}$ there is a $c(t) \in \mathbb{N}^{>0}$ with $(\mathcal{S}_M(\mathcal{A}), t) = c(t).\big[(\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2), t)\big]$. Since $\mathbb{K}$ is idempotent, the wba $\mathcal{A}$ M-recognizes $S_1 \odot S_2$. □

Corollary 8.9. *Let $\mathbb{K}$ be an idempotent doubled semiring, and let $S_1, S_2 \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ two C-regular sp-series of bounded width. Then $S_1 \odot S_2$ is C-regular.*

Proof. By Theorem 7.6, S_1 and S_2 are M-regular. Hence, $S_1 \odot S_2$ is M-regular by Theorem 8.8. Since $S_1 \odot S_2$ is of bounded width again, $S_1 \odot S_2$ is C-regular by Theorem 7.6. □

Since the Boolean bisemiring $\mathbb{B}$ is an idempotent doubled semiring we get:

Corollary 8.10. *M-regular sp-languages are closed under intersection. C-regular sp-languages of bounded width are closed under intersection.*

The regular sp-languages as defined by Lodaya and Weil [LW00] are C-regular languages in our sense. In [LW00, Thm. 2.6], they noted that the recognizable sp-languages[6] are closed under intersection. But not every regular sp-language is recognizable. In [LW00, Thm. 4.6], they stated that regular languages are closed under intersection and refer to the classical construction. As we have seen in Example 8.1, this construction cannot be generalized straightforwardly in the C-running mode. Later on, they showed in [LW01, Thm. 5.8] that an sp-language L is regular iff L is the image of a recognizable subset of the series-Σ-algebra $S_\Sigma(A)$ under the natural projection π. In $S_\Sigma(A)$ the parallel multiplication is understood as a binary term operation. Again, the recognizable languages of $S_\Sigma(A)$ are closed under intersection. But the projection of the intersection of two $S_\Sigma(A)$-languages is in general not the intersection of the two projections.

[6] Here, *recognizable sp-languages* are those which are recognized by a morphism into a finite sp-algebra.

Thus, Corollary 8.10 seems to be the most general result for the closure of regular sp-languages under intersection known so far. It is an open question if the intersection of two C-regular sp-languages is C-regular also for unbounded width.

In the remainder of this section we will discuss the prerequisite of idempotency in Theorem 8.8. Idempotency was imposed on the bisemiring because of auto-concurrency. We could think of overcoming this problem by constructing wba where we know the "rate of auto-concurrency" when branching, *i.e.* the fork transitions would have a certain "type" of auto-concurrency. Then fork transitions of the same type only would be fused in a Hadamard product automaton. But how to fix the type of a fork transition with concern to auto-concurrency? If some parallel sp-poset $t = t_1 \parallel \ldots \parallel t_n$ $(n \geq 2)$ is given, the number of sequential factors of each t_i $(i = 1, \ldots, n)$ is finite but may be arbitrary large. Thus, to decide the right type of auto-concurrency it would be necessary to encode all the possible future of a state within this state. But this is impossible within the frame of finite-state systems. Such a construction would be possible only if there is an $N \in \mathbb{N}$ such that for every parallel sp-poset $t = t_1 \parallel \ldots \parallel t_n$ the number of sequential factors of each t_i is bounded by N. This would be a rather strong restriction and is not worked out any further.

However, if the supports of the sp-series $\mathcal{S}_M(\mathcal{A}_1)$ and $\mathcal{S}_M(\mathcal{A}_2)$ of two wba $\mathcal{A}_1$ and $\mathcal{A}_2$ do not contain sp-posets with a "direct concurrency", then the product automaton will M-recognize the Hadamard product $\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2)$ even if the underlying doubled semiring $\mathbb{K}$ is not idempotent. More precisely, we define the set of *sp-posets without direct auto-concurrency* $\mathrm{SP}_{\nparallel}(\Sigma)$ as follows:

- $a \in \mathrm{SP}_{\nparallel}(\Sigma)$ for every $a \in \Sigma$,
- if $t_1, t_2 \in \mathrm{SP}_{\nparallel}(\Sigma)$, then $t_1 \cdot t_2 \in \mathrm{SP}_{\nparallel}(\Sigma)$,
- if $t_1, t_2 \in \mathrm{SP}_{\nparallel}(\Sigma)$ where $t_1 = t_{11} \parallel \ldots \parallel t_{1m}$ and $t_2 = t_{21} \parallel \ldots \parallel t_{2n}$ with $m, n \geq 1$ are the maximal parallel decompositions of t_1 and t_2, respectively, and $t_{1i} \neq t_{2j}$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, then $t_1 \parallel t_2 \in \mathrm{SP}_{\nparallel}(\Sigma)$,
- $\mathrm{SP}_{\nparallel}(\Sigma)$ is the least subset of $\mathrm{SP}(\Sigma)$ satisfying the three properties given above.

Hence, an sp-poset t is in $\mathrm{SP}_{\not\parallel}(\Sigma)$ if in the composition of t two equal posets are never multiplied in parallel. For an sp-series $S \in \mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$ we put $S \in \mathbb{K}\langle\langle \mathrm{SP}_{\not\parallel}\rangle\rangle$ if $\operatorname{supp} S \subseteq \mathrm{SP}_{\not\parallel}(\Sigma)$. Then we get:

Theorem 8.11. *Let $S_1 \in \mathbb{K}\langle\langle \mathrm{SP}_{\not\parallel}\rangle\rangle$, $S_2 \in \mathbb{K}\langle\langle \mathrm{SP}\rangle\rangle$ be M-regular sp-series over the doubled semiring $\mathbb{K}$. Then $S_1 \odot S_2$ is M-regular.*

Proof. Let $\mathcal{A}_i$ be a wba M-recognizing S_i for $i = 1, 2$, and let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Then we get:

Claim 1. For every $t \in \mathrm{SP}_{\not\parallel}$ and every $p^1, q^1 \in Q_1$, $p^2, q^2 \in Q_2$ we have:

$$\mathbf{wgt}_M\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, t, \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right) = \mathbf{wgt}^1_M(p^1, t, q^1) \circ \mathbf{wgt}^2_M(p^2, t, q^2).$$

We refer to the proof of Claim 1 in the proof of Lemma 8.4. There, a factor $c(t)$ had to be considered additionally. We have $c(t) = 1$ for $t = a \in \Sigma$. For $t = t_1 \cdot \ldots \cdot t_m$ we get $c(t) = c(t_1) \ldots c(t_m)$. Hence, $c(t) = 1$ if $c(t_i) = 1$ for $i = 1, \ldots, m$. For $t = t_1 \parallel \ldots \parallel t_m$ Equation (8.9) implied $c(t) = c(t_1) \ldots c(t_m) e(t)$ where $e(t) = |\ker(t)|$. Since $t \in \mathrm{SP}_{\not\parallel}$, all factors t_i $(i = 1, \ldots, m)$ are mutually different. Hence, $\ker(t)$ contains the identity only and $e(t) = 1$. By induction hypothesis $c(t_i) = 1$ for $i = 1, \ldots, m$. Therefore, $c(t) = 1$. This proves Claim 1.

By Claim 1 we get immediately $(\mathcal{S}_M(\mathcal{A}), t) = (S_1 \odot S_2, t)$ for all $t \in \mathrm{SP}_{\not\parallel}$. Now, let $t \in \mathrm{SP} \setminus \mathrm{SP}_{\not\parallel}$. Then $t \notin \operatorname{supp} S_1$, and, therefore, $t \notin \operatorname{supp}(S_1 \odot S_2)$. By Lemma 8.4, there is a $c(t) \in \mathbb{N}^{>0}$ with

$$(\mathcal{S}_M(\mathcal{A}), t) = c(t).[(S_1 \odot S_2, t)] = c(t).0 = 0.$$

Hence, $t \notin \operatorname{supp} \mathcal{S}_M(\mathcal{A})$. This proves $\mathcal{S}_M(\mathcal{A}) = S_1 \odot S_2$. $\square$

Last but not least, the prerequisite of Theorem 8.11 is decidable assumed $\mathbb{K}$ is positive.

Lemma 8.12. *Let $\mathbb{K}$ be a positive doubled semiring, and let $\mathcal{A}$ be a wba with weights from $\mathbb{K}$. It is decidable whether $\mathcal{S}_M(\mathcal{A}) \in \mathbb{K}\langle\langle \mathrm{SP}_{\not\parallel}\rangle\rangle$.*

Proof. Since $\mathbb{K}$ is positive, $\mathcal{S}_M(\mathcal{A}) \in \mathbb{K}\langle\langle \mathrm{SP}_{\not\parallel}\rangle\rangle$ if and only if there is no successful run with a parallel sub-run $G = \parallel_{p,q}(G_1, \ldots, G_m)$ such that there are $i \neq j \in \{1, \ldots, m\}$ with $\mathbf{lab}(G_i) = \mathbf{lab}(G_j)$. The idea is to compute the set P of all those pairs (p, q) such that there is a parallel non-atomic M-run from p to q as a sub-run of a successful run. Then we test whether $\mathbf{lab}(G_i) = \mathbf{lab}(G_j)$ by building appropriate product automata, using Corollary 8.7 and deciding empty support of weighted automata over the positive bisemiring $\mathbb{K}$.

Firstly, we show how to compute P. Let M be the set of all pairs (p, q) with $p, q \in Q$ such that there is an M-run from p to q, and let $\overline{M} \subseteq M$ be the set of those pairs such that there is a sequential run between the two states. Both M and $\overline{M}$ are computable as shown in the proof of Lemma 5.14.

Let $B_M \subseteq M$ contain only those pairs (p, q) such that there is a parallel non-atomic M-run between p and q. Since there are finitely many fork and join transitions only and $\overline{M}$ is computable, also B_M is computable.

We recall the construction of the directed graph Γ_M with vertex set M and two kinds of directed edges from the proof of Lemma 5.14. For all $(p, q) \in B_M$, fork transitions $p \to \{p_1, \ldots, p_m\}$, and join transitions $\{q_1, \ldots, q_m\} \to q$, $\alpha \in S_m$ with $(p_i, q_{\alpha(i)}) \in \overline{M}$ for $i = 1, \ldots, m$ we add a red arrow from (p, q) to $(p_i, q_{\alpha(i)})$ for every $i = 1, \ldots, m$. Furthermore, for every $(p, q) \in M$ and every $(\tilde{p}, \tilde{q}) \in B_M$ we put a blue arrow from (p, q) to $(\tilde{p}, \tilde{q})$ if

- $p = \tilde{p}$ and $(\tilde{q}, q) \in M$, or
- $q = \tilde{q}$ and $(p, \tilde{p}) \in M$, or
- $(p, \tilde{p}) \in M$ and $(\tilde{q}, q) \in M$.

Now $(\tilde{p}, \tilde{q}) \in B_M$ is in P if and only if there is a cycle-free path in Γ_M from some $(\mathfrak{i}, \mathfrak{f}) \in M$ with $\mathfrak{i}$ initial and $\mathfrak{f}$ final to $(\tilde{p}, \tilde{q})$ such that the path is either empty, or consists of red and blue edges in turn ending with a blue edge. Hence, P is computable.

Now let $(p, q) \in P$. Then any M-run from p to q is a sub-run of some successful M-run. Let $f : p \to \{p_1, \ldots, p_m\}$ be any fork starting in p, and $j : \{q_1, \ldots, q_m\} \to q$ a matching join to f ending in q. For any $\alpha \in S_m$ with $(p_i, q_{\alpha(i)}) \in \overline{M}$ for all $i = 1, \ldots, m$, we test whether there are runs $p_i \xrightarrow{t} q_{\alpha(i)}$ and $p_j \xrightarrow{t} q_{\alpha(j)}$ for $i \neq j$ with the same label t as follows:

For $i = 1, \ldots, m$ let $\mathcal{A}(p_i, q_{\alpha(i)})$ be the wba with the same states and transitions as $\mathcal{A}$, but p_i the only initial and $q_{\alpha(i)}$ the only final state both with initial and final weight 1, respectively. Since $\mathbb{K}$ is positive, we have by Corollary 8.7

$$\begin{aligned}
&\operatorname{supp}\Big[\mathcal{S}_M\big(\mathcal{A}(p_i, q_{\alpha(i)}) \times \mathcal{A}(p_j, q_{\alpha(j)})\big)\Big] \\
&= \operatorname{supp}\Big[\mathcal{S}_M\big(\mathcal{A}(p_i, q_{\alpha(i)})\big) \odot \mathcal{S}_M\big(\mathcal{A}(p_j, q_{\alpha(j)})\big)\Big] \\
&= \operatorname{supp}\Big[\mathcal{S}_M\big(\mathcal{A}(p_i, q_{\alpha(i)})\big)\Big] \cap \operatorname{supp}\Big[\mathcal{S}_M\big(\mathcal{A}(p_j, q_{\alpha(j)})\big)\Big].
\end{aligned}$$

Hence, there is no $t \in \mathrm{SP}$ with $p_i \xrightarrow{t} q_{\alpha(i)}$ and $p_j \xrightarrow{t} q_{\alpha(j)}$ if and only if $\operatorname{supp}\Big[\mathcal{S}_M\big(\mathcal{A}(p_i, q_{\alpha(i)}) \times \mathcal{A}(p_j, q_{\alpha(j)})\big)\Big] = \emptyset$. So we construct for all $i \neq j \in \{1, \ldots, m\}$ the product automaton $\mathcal{A}_{i,j} = \mathcal{A}(p_i, q_{\alpha(i)}) \times \mathcal{A}(p_j, q_{\alpha(j)})$. By Lemma 5.14 empty support is decidable. Hence, we can decide whether there are $i \neq j$ and $t \in \mathrm{SP}$ with $p_i \xrightarrow{t} q_{\alpha(i)}$ and $p_j \xrightarrow{t} q_{\alpha(j)}$. Continuing this procedure for all matching pairs (f, j) such that f starts in p, and j ends in q for some $(p, q) \in P$, and for all suitable permutations $\alpha \in S_m$ ($2 \leq m \leq |Q|$), we can decide whether $\mathcal{S}_M(\mathcal{A}) \in \mathbb{K}\langle\langle \mathrm{SP}_{\nparallel} \rangle\rangle$. $\square$

8.2 The Hadamard Product for Distributive Bisemirings

In this section, we will see that the situation concerning the closure of M-regular sp-series under Hadamard product is not as favorable for distributive bisemirings as it was for doubled semirings. The following example shows that the product automaton does not yield the correct result.

Example 8.13. In this example, we consider the tropical bisemiring $\mathbb{T} = (\mathbb{N} \cup \{+\infty\}, \min, +, \max, +\infty, 0)$. Note that $\mathbb{T}$ is idempotent, commutative, and distributive. Now, two wba $\mathcal{A}_1$ and $\mathcal{A}_2$ over $\mathbb{T}$ are given as shown in Figure 8.4. We have $\mathcal{S}_M(\mathcal{A}_1) = 5[b \parallel c \parallel (ca)]$ and $\mathcal{S}_M(\mathcal{A}_2) = 4[b \parallel c \parallel (ca)]$. Thus, $\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2) = 9[b \parallel c \parallel (ca)]$.

But if we apply the product automaton construction as given by Construction 8.2 we would get as the only run of $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ on $b \parallel c \parallel (ca)$ the

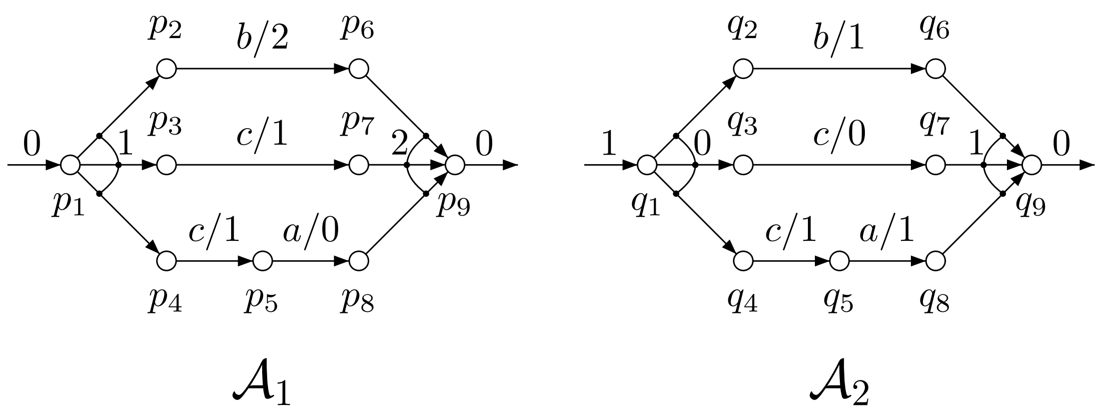


Figure 8.4: Two wba M-recognizing $b \parallel c \parallel (ca)$ with different weights.

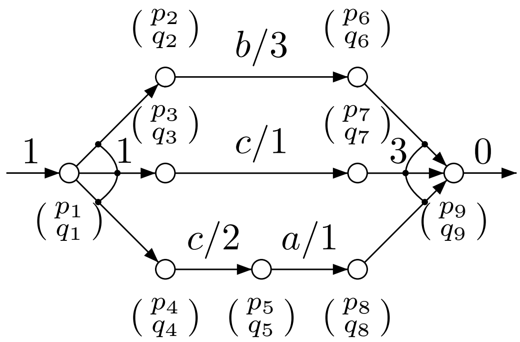


Figure 8.5: The only run on $b \parallel c \parallel (ca)$ in the product automaton $\mathcal{A}_1 \times \mathcal{A}_2$.

one indicated in Figure 8.5. Hence, $(\mathcal{S}_M(\mathcal{A}), b \parallel c \parallel (ca)) = 8 \leq 9$. What is the problem right here? The parallel sp-poset $b \parallel c \parallel (ca)$ is one sequential step in the automaton. This step has without considering the weights for fork and join transitions the weight 2 both in $\mathcal{A}_1$ and $\mathcal{A}_2$. Therefore, this sequential step must have weight $2 + 2 = 4$ in an automaton recognizing the Hadamard product. But unfortunately, the weight of this sequential step is realized in $\mathcal{A}_1$ and $\mathcal{A}_2$ in different sub-runs. In $\mathcal{A}_1$ the weight arises from the sub-run executing b, but in $\mathcal{A}_2$ from the sub-run executing (ca). In the construction of the product automaton the respective weights of the single transitions are multiplied, *i.e.* in $\mathbb{T}$ added by the usual addition. Thus, none of the sub-runs with label b, c, and (ca)

has weight 4, but a weight less than 4 (cf. Figure 8.5). Consequently, $\big(\mathcal{S}_M(\mathcal{A}), b \,\|\, c \,\|\, (ca)\big) \leq \big(\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2), b \,\|\, c \,\|\, (ca)\big)$.

The following lemma states that the situation noted in the last example can be generalized. The lemma is formulated for the family of tropical bisemirings. Later on, we discuss that this result may be generalized to bisemirings which are commutative, idempotent, distributive, and carry an order which is compatible with the operations of the bisemiring.

Lemma 8.14. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be two wba over the $\mathbb{N}$-, $\mathbb{Z}$-, $\mathbb{Q}$-, or $\mathbb{R}$-tropical bisemiring, and let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$. Further on, let $\leq$ be the usual order on $\mathbb{R}$ with $r \leq +\infty$ for all $r \in \mathbb{R}$. Then $\operatorname{supp}[\mathcal{S}_M(\mathcal{A})] = \operatorname{supp}[\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2)]$ and $(\mathcal{S}_M(\mathcal{A}), t) \leq (\mathcal{S}_M(\mathcal{A}_1) \odot \mathcal{S}_M(\mathcal{A}_2), t)$ for all $t \in \mathrm{SP}$.*

Proof. We put $S = \mathcal{S}_M(\mathcal{A})$ and $S_i = \mathcal{S}_M(\mathcal{A}_i)$ for $i = 1, 2$.

By structural induction on $t \in \mathrm{SP}$ we show the following:

Claim 1. For every $t \in \mathrm{SP}$, $p^1, q^1 \in Q_1$, and $p^2, q^2 \in Q_2$:

1. $\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \leq \mathbf{wgt}^1_M(p^1, t, q^1) + \mathbf{wgt}^2_M(p^2, t, q^2)$, and
2. if $\mathbf{wgt}^1_M(p^1, t, q^1)$ or $\mathbf{wgt}^2_M(p^2, t, q^2)$ equals $+\infty$, then
$$\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) = +\infty.$$

The second part of Claim 1 is clear. If $\mathbf{wgt}^1_M(p^1, t, q^1) = +\infty$, there is no run on t from p^1 to q^1 in $\mathcal{A}_1$. Hence, there is no run on t in $\mathcal{A}$ from $\binom{p^1}{q^2}$ to $\binom{q^1}{q^2}$ where p^2, q^2 are arbitrary states of $\mathcal{A}_2$. This is because a transition in $\mathcal{A}$ is defined only if the projections of this transition are defined in $\mathcal{A}_1$ and $\mathcal{A}_2$. Therefore, $\mathbf{wgt}_M\left(\binom{p^1}{q^2}, t, \binom{q^1}{q^2}\right) = +\infty$. For $\mathbf{wgt}^2_M(p^1, t, q^1) = +\infty$ the claim follows similarly.

Now, we prove the first part of Claim 1. Let $t = a \in \Sigma$. Then the claim follows by the definition of $\mathcal{A}$ immediately. In this case we get even equality. Now, assume $t = t_1 \cdot \ldots \cdot t_m$ ($m \geq 2$) is sequential with its maximal sequential decomposition. We get:

$$
\begin{aligned}
&\mathbf{wgt}_M\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, t, \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right)\\
=&\min\left\{\mathbf{wgt}_M(G) \;\middle|\; G : \begin{pmatrix} p^1 \\ p^2 \end{pmatrix} \xrightarrow{t} \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right\}\\
=&\min\Big\{\mathbf{wgt}_M(G_1) + \cdots + \mathbf{wgt}_M(G_m) \;\Big|\; G = G_1 \cdot \ldots \cdot G_m,\\
&\qquad\qquad \mathbf{lab}(G_i) = t_i \quad \text{for} \quad i = 1, \ldots, m\Big\}\\
=&\min\Big\{\mathbf{wgt}_M\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, t_1, \begin{pmatrix} r_1^1 \\ r_1^2 \end{pmatrix}\right) + \cdots + \mathbf{wgt}_M\left(\begin{pmatrix} r_{m-1}^1 \\ r_{m-1}^2 \end{pmatrix}, t_m, \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right) \;\Big|\\
&\qquad\qquad \left[\begin{pmatrix} r_1^1 \\ r_1^2 \end{pmatrix}, \ldots, \begin{pmatrix} r_{m-1}^1 \\ r_{m-1}^2 \end{pmatrix}\right] \in (Q_1 \times Q_2)^{m-1}\Big\}
\end{aligned}
$$

(by distributivity of + over min)

$$
\begin{aligned}
\leq&\min\Big\{\mathbf{wgt}_M^1(p^1, t_1, r_1^1) + \cdots + \mathbf{wgt}_M^1(r_{m-1}^1, t_m, q^1)\\
&\quad + \mathbf{wgt}_M^2(p^2, t_1, r_1^2) + \cdots + \mathbf{wgt}_M^2(r_{m-1}^2, t_m, q^2) \;\Big|\\
&\qquad\qquad \left[\begin{pmatrix} r_1^1 \\ r_1^2 \end{pmatrix}, \ldots, \begin{pmatrix} r_{m-1}^1 \\ r_{m-1}^2 \end{pmatrix}\right] \in (Q_1 \times Q_2)^{m-1}\Big\}
\end{aligned}
$$

(by induction hypothesis)

$$
\begin{aligned}
=&\min\Big\{\mathbf{wgt}_M^1(p^1, t_1, r_1^1) + \cdots + \mathbf{wgt}_M^1(r_{m-1}^1, t_m, q^1) \;\Big|\\
&\qquad\qquad (r_1^1, \ldots, r_{m-1}^1) \in Q_1^{m-1}\Big\}\\
&+\min\Big\{\mathbf{wgt}_M^2(p^2, t_1, r_1^2) + \cdots + \mathbf{wgt}_M^2(r_{m-1}^2, t_m, q^2) \;\Big|\\
&\qquad\qquad (r_1^2, \ldots, r_{m-1}^2) \in Q_2^{m-1}\Big\}
\end{aligned}
$$

(by distributivity of + over min)

$$= \mathbf{wgt}^1_M(p^1, t, q^1) + \mathbf{wgt}^2_M(p^2, t, q^2).$$

Now, let $t = t_1 \parallel \ldots \parallel t_m$ $(m \geq 2)$ be a parallel sp-poset with its maximal parallel decomposition. We refer to the proof of Lemma 8.4 for most of the details of the following calculation. Here, we only point out the differences to the calculations of the mentioned proof. Firstly, we stress the following two differences to the proof of Lemma 8.4: the induction hypothesis is an inequality instead of an equality, and sequential and parallel multiplication are not the same, but the sequential multiplication + distributes over the parallel multiplication max.

Let U_m be a system of representatives of the left cosets $\pi \ker(t)$ with $\pi \in S_m$ as defined in the proof of Lemma 8.4. In order to calculate $\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right)$ we take the minimum of all possible runs on t from state $\binom{p^1}{p^2}$ to state $\binom{q^1}{q^2}$. Thus, we get:

$$\begin{aligned}
&\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right)\\
&= \min_{\left\{\binom{p^1_1}{p^2_1},\ldots,\binom{p^1_m}{p^2_m}\right\} \in \mathfrak{P}_m(Q)} \; \min_{\left\{\binom{q^1_1}{q^2_1},\ldots,\binom{q^1_m}{q^2_m}\right\} \in \mathfrak{P}_m(Q)} \; \min_{\alpha \in S_m, \beta \in U_m}\\
&\qquad \Bigg\{ \mu^m_{\text{fork}}\left(\binom{p^1}{p^2}, \left\{\binom{p^1_1}{p^2_1}, \ldots, \binom{p^1_m}{p^2_m}\right\}\right)\\
&\qquad + \max_{i=1,\ldots,m} \left\{ \mathbf{wgt}_M\left(\binom{p^1_{\alpha(i)}}{p^2_{\alpha(i)}}, t_i, \binom{q^1_{\beta(i)}}{q^2_{\beta(i)}}\right)\right\}\\
&\qquad + \mu^m_{\text{join}}\left(\left\{\binom{q^1_1}{q^2_1}, \ldots, \binom{q^1_m}{q^2_m}\right\}, \binom{q^1}{q^2}\right)\Bigg\} \qquad (8.10)
\end{aligned}$$

We refer to the proof of Lemma 8.4 for the correctness of this calculation. Now, we proceed by replacing m-sets of states from Q by pairs of m-sets of states from Q_1 and Q_2, respectively. In return we have to establish new permutations. As in the proof of Lemma 8.4 we conclude:

$$= \min_{\substack{\left(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}\right),\\ \left(\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\}\right)\\ \in\mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \left\{ {\mu_1}^m_{\text{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) + {\mu_2}^m_{\text{fork}}(p^2,\{r_1^2,\ldots,r_m^2\}) \right. \\ + \max_{i=1,\ldots,m}\left\{\mathbf{wgt}_M\left(\begin{pmatrix} r^1_{\pi(i)} \\ r^2_{\tau(i)}\end{pmatrix}, t_i, \begin{pmatrix} s^1_{\varrho(i)} \\ s^2_{\eta(i)}\end{pmatrix}\right)\right\} \\ \left. + {\mu_1}^m_{\text{join}}(\{s_1^1,\ldots,s_m^1\},q^1) + {\mu_2}^m_{\text{join}}(\{s_1^2,\ldots,s_m^2\},q^2) \right\} \tag{8.11}$$

Note that the order $\leq$ is compatible with all bisemiring operations, *i.e.* $k \leq k'$ and $l \leq l'$ imply $k * l \leq k' * l'$ for $* \in \{\min, +, \max\}$. By this and the induction hypothesis we get:

$$\leq \min_{\substack{\left(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}\right),\\ \left(\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\}\right)\\ \in\mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \left\{ {\mu_1}^m_{\text{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) + {\mu_2}^m_{\text{fork}}(p^2,\{r_1^2,\ldots,r_m^2\}) \right. \\ + \max_{i=1,\ldots,m}\left\{\mathbf{wgt}^1_M(r^1_{\pi(i)}, t_i, s^1_{\varrho(i)}) + \mathbf{wgt}^2_M(r^2_{\tau(i)}, t_i, s^2_{\eta(i)})\right\} \\ \left. + {\mu_1}^m_{\text{join}}(\{s_1^1,\ldots,s_m^1\},q^1) + {\mu_2}^m_{\text{join}}(\{s_1^2,\ldots,s_m^2\},q^2) \right\} \tag{8.12}$$

Next, we extend the set of values over which the maximum is taken, using the relation $\max(k,l) \leq \max(k,l,h)$ for $k,l,h \in \mathbb{R} \cup \{+\infty\}$:

$$\leq \min_{\substack{\left(\{r_1^1,\dots,r_m^1\},\{r_1^2,\dots,r_m^2\}\right),\\ \left(\{s_1^1,\dots,s_m^1\},\{s_1^2,\dots,s_m^2\}\right)\\ \in \mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \Bigg\{ \mu 1^m_{\text{fork}}(p^1,\{r_1^1,\dots,r_m^1\}) + \mu 2^m_{\text{fork}}(p^2,\{r_1^2,\dots,r_m^2\}) + \max_{\substack{i=1,\dots,m\\ j=1,\dots,m}} \left\{ \mathbf{wgt}^1_M(r^1_{\pi(i)},t_i,s^1_{\varrho(i)}) + \mathbf{wgt}^2_M(r^2_{\tau(j)},t_j,s^2_{\eta(j)}) \right\} + \mu 1^m_{\text{join}}(\{s_1^1,\dots,s_m^1\},q^1) + \mu 2^m_{\text{join}}(\{s_1^2,\dots,s_m^2\},q^2) \Bigg\} \quad (8.13)$$

Now, by distributivity of + over max we get:

$$= \min_{\substack{\left(\{r_1^1,\dots,r_m^1\},\{r_1^2,\dots,r_m^2\}\right),\\ \left(\{s_1^1,\dots,s_m^1\},\{s_1^2,\dots,s_m^2\}\right)\\ \in \mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \Bigg\{ \mu 1^m_{\text{fork}}(p^1,\{r_1^1,\dots,r_m^1\}) + \mu 2^m_{\text{fork}}(p^2,\{r_1^2,\dots,r_m^2\}) + \max_{i=1,\dots,m} \left\{ \mathbf{wgt}^1_M(r^1_{\pi(i)},t_i,s^1_{\varrho(i)}) + \max_{j=1,\dots,m} \left\{ \mathbf{wgt}^2_M(r^2_{\tau(j)},t_j,s^2_{\eta(j)}) \right\} \right\} + \mu 1^m_{\text{join}}(\{s_1^1,\dots,s_m^1\},q^1) + \mu 2^m_{\text{join}}(\{s_1^2,\dots,s_m^2\},q^2) \Bigg\} \quad (8.14)$$

We apply distributivity of + over max once more, and get:

$$= \min_{\substack{\left(\{r_1^1,\ldots,r_m^1\},\{r_1^2,\ldots,r_m^2\}\right),\\ \left(\{s_1^1,\ldots,s_m^1\},\{s_1^2,\ldots,s_m^2\}\right)\\ \in\mathfrak{P}_m(Q_1)\times\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}} \Bigg\{ {\mu_1}^m_{\text{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) + {\mu_2}^m_{\text{fork}}(p^2,\{r_1^2,\ldots,r_m^2\})$$
$$+ \max_{i=1,\ldots,m}\left\{\mathbf{wgt}^1_M(r^1_{\pi(i)},t_i,s^1_{\varrho(i)})\right\} + \max_{j=1,\ldots,m}\left\{\mathbf{wgt}^2_M(r^2_{\tau(j)},t_j,s^2_{\eta(j)})\right\}$$
$$+{\mu_1}^m_{\text{join}}(\{s_1^1,\ldots,s_m^1\},q^1) + {\mu_2}^m_{\text{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Bigg\} \quad (8.15)$$

By commutativity of + we get:

$$= \min_{\substack{\{r_1^1,\ldots,r_m^1\},\{s_1^1,\ldots,s_m^1\}\\ \in\mathfrak{P}_m(Q_1)}} \min_{\substack{\{r_1^2,\ldots,r_m^2\},\{s_1^2,\ldots,s_m^2\}\\ \in\mathfrak{P}_m(Q_2)}} \min_{\substack{\pi,\tau,\eta\in S_m\\ \varrho\in U_m}}$$
$$\Bigg\{ {\mu_1}^m_{\text{fork}}(p^1,\{r_1^1,\ldots,r_m^1\}) + \max_{i=1,\ldots,m}\left\{\mathbf{wgt}^1_M(r^1_{\pi(i)},t_i,s^1_{\varrho(i)})\right\}$$
$$+ {\mu_1}^m_{\text{join}}(\{s_1^1,\ldots,s_m^1\},q^1)$$
$$+ {\mu_2}^m_{\text{fork}}(p^2,\{r_1^2,\ldots,r_m^2\}) + \max_{j=1,\ldots,m}\left\{\mathbf{wgt}^2_M(r^2_{\tau(j)},t_j,s^2_{\eta(j)})\right\}$$
$$+ {\mu_2}^m_{\text{join}}(\{s_1^2,\ldots,s_m^2\},q^2)\Bigg\} \quad (8.16)$$

By distributivity of the sequential multiplication + over the addition min we get:

$$
\begin{aligned}
&= \min_{\substack{\{r^1_1,\dots,r^1_m\},\{s^1_1,\dots,s^1_m\}\\ \in\mathfrak{P}_m(Q_1)}} \min_{\pi\in S_m,\varrho\in U_m} \Bigg\{ \mu 1^m_{\text{fork}}(p^1,\{r^1_1,\dots,r^1_m\}) \\
&\quad + \max_{i=1,\dots,m} \left\{ \mathbf{wgt}^1_M(r^1_{\pi(i)},t_i,s^1_{\varrho(i)}) \right\} + \mu 1^m_{\text{join}}(\{s^1_1,\dots,s^1_m\},q^1) \Bigg\} \\
&\quad + \min_{\substack{\{r^2_1,\dots,r^2_m\},\{s^2_1,\dots,s^2_m\}\\ \in\mathfrak{P}_m(Q_2)}} \min_{\tau,\eta\in S_m} \Bigg\{ \mu 2^m_{\text{fork}}(p^2,\{r^2_1,\dots,r^2_m\}) \\
&\quad + \max_{j=1,\dots,m} \left\{ \mathbf{wgt}^2_M(r^2_{\tau(j)},t_j,s^2_{\eta(j)}) \right\} + \mu 2^m_{\text{join}}(\{s^2_1,\dots,s^2_m\},q^2) \Bigg\}
\end{aligned}
\tag{8.17}
$$

With a similar argument as for Equations (8.1) and (8.2) in the proof of Lemma 8.4 and by idempotency of min we get:

$$
\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2} \right) \leq \mathbf{wgt}^1_M(p^1,t,q^1) + \mathbf{wgt}^2_M(p^2,t,q^2). \tag{8.18}
$$

This proves Claim 1. By this we get for every $t \in \mathrm{SP}$ and $S = \mathcal{S}_M(\mathcal{A})$:

$$
(S,t) = \min_{\binom{p^1}{p^2},\binom{q^1}{q^2}\in Q} \left(\lambda\binom{p^1}{p^2} + \mathbf{wgt}_M\left(\binom{p^1}{p^2},t,\binom{q^1}{q^2}\right) + \gamma\binom{q^1}{q^2} \right) \tag{8.19}
$$

$$
\begin{aligned}
&\leq \min_{\substack{p^1,q^1\in Q_1\\ p^2,q^2\in Q_2}} \Big(\lambda_1(p^1) + \mathbf{wgt}^1_M(p^1,t,q^1) + \gamma_1(q^1) \\
&\qquad + \lambda_2(p^2) + \mathbf{wgt}^2_M(p^2,t,q^2) + \gamma_2(q^2) \Big)
\end{aligned}
\tag{8.20}
$$

$$\begin{aligned} &= \min_{p^1,q^1\in Q_1}\left(\lambda_1(p^1)+\mathbf{wgt}^1_M(p^1,t,q^1)+\gamma_1(q^1)\right) \\ &\qquad + \min_{p^2,q^2\in Q_2}\left(\lambda_2(p^2)+\mathbf{wgt}^2_M(p^2,t,q^2)+\gamma_2(q^2)\right) \end{aligned} \tag{8.21}$$

$$= (\mathcal{S}_M(\mathcal{A}_1),t)+(\mathcal{S}_M(\mathcal{A}_2),t) \tag{8.22}$$

$$= (S_1 \odot S_2, t) \tag{8.23}$$

Finally, we show $\operatorname{supp}\mathcal{S}_M(\mathcal{A}) = \operatorname{supp}(\mathcal{S}_M(\mathcal{A}_1)\odot\mathcal{S}_M(\mathcal{A}_2))$. Since $(S,t) \le (S_1 \odot S_2, t)$ for all $t \in \mathrm{SP}$ and $r \le +\infty$ for all $r \in \mathbb{R}$, $t \in \operatorname{supp}(S_1 \odot S_2)$ implies $t \in \operatorname{supp} S$. Vice versa, if $t \in \operatorname{supp} S$ then there are $\binom{p^1}{p^2}, \binom{q^1}{q^2} \in Q$ such that

$$\lambda\binom{p^1}{p^2} + \mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) + \gamma\binom{q^1}{q^2} < +\infty. \tag{8.24}$$

Hence, $\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) < +\infty$, and, by Claim 1, also $\mathbf{wgt}^1_M(p^1,t,q^1) < +\infty$ and $\mathbf{wgt}^2_M(p^2,t,q^2) < +\infty$. Moreover, Equation (8.24) implies $\lambda_i(p^i), \gamma^i(q^i) < +\infty$ for $i = 1,2$. Hence, $t \in \operatorname{supp}(S_i)$ for $i = 1,2$. This implies $t \in \operatorname{supp}(S_1 \odot S_2)$ because the sequential multiplication $+$ is zero-divisor-free. □

A careful analysis of the last proof shows that we have used the following properties of the underlying bisemiring $\mathbb{K} = (K, \oplus, \circ, \diamond, 0, 1)$:

1. $\mathbb{K}$ is idempotent, commutative, and distributive,
2. the sequential multiplication is zero-divisor-free,
3. the order $\le$ on $\mathbb{K}$ is compatible with all three operations of $\mathbb{K}$,
4. $k \le k \diamond l$ for all $k, l \in \mathbb{K}$, and
5. $k \le 0$ for all $k \in \mathbb{K}$.

Another example of a bisemiring having all these properties is the one of Example 2.10 $(\mathbb{R}^{>0} \cup \{+\infty\}, \min, \cdot, +, +\infty, 1)$ together with the usual

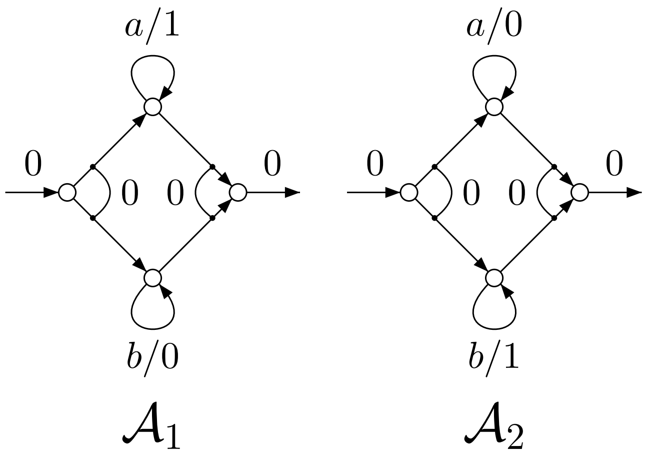


Figure 8.6: Two wba M-recognizing S_1 and S_2 from Example 8.15.

order on the reals and $r \leq +\infty$ for all $r \in \mathbb{R}^{>0}$. Note that we used condition (5) "$k \leq 0$ for all $k \in \mathbb{K}$" only to show that $t \in \operatorname{supp}(S_1 \odot S_2)$ implies $t \in \operatorname{supp} S$. We will see later on that this implication can be shown for positive bisemirings without using condition (5). Thus, Lemma 8.14 is also valid for the polar bisemiring $(\mathbb{N} \cup \{-\infty\}, \max, +, \mathrm{nax}, -\infty, 0)$ from Example 2.7 with the usual order and $-\infty \leq n$ for all $n \in \mathbb{N}$.

We will close this section with an example showing that M-regular sp-series over idempotent, commutative, and distributive bisemirings are in general not closed under Hadamard product.

Example 8.15. We work with the polar bisemiring $\mathbb{P} = (\mathbb{N} \cup \{-\infty\}, \max, +, \mathrm{nax}, -\infty, 0)$ from Example 2.7 where $\mathrm{nax}(k,l) = \max(k,l)$ if $k, l \neq -\infty$ and equal to $-\infty$ otherwise. Recall that $\mathbb{P}$ is idempotent, commutative, and distributive. We define two sp-series S_1, S_2 over $\mathbb{P}$ by:

$$(S_1, t) = \begin{cases} m & \text{if } t = a^m \parallel b^n \ (m, n \geq 1), \\ -\infty & \text{otherwise,} \end{cases}$$

$$(S_2, t) = \begin{cases} n & \text{if } t = a^m \parallel b^n \ (m, n \geq 1), \\ -\infty & \text{otherwise.} \end{cases}$$

Clearly, S_1 and S_2 are M-regular (cf. Figure 8.6) and of bounded width.

Moreover, we have

$$(S_1 \odot S_2, t) = \begin{cases} m+n & \text{if } t = a^m \parallel b^n \ (m, n \geq 1), \\ -\infty & \text{otherwise.} \end{cases}$$

We will show that $S_1 \odot S_2$ is not M-regular, and, hence, also not C-regular by Theorem 7.6.

Assume $S_1 \odot S_2$ is M-regular. Then there is a wba $\mathcal{A}$ with $\mathcal{S}_M(\mathcal{A}) = S_1 \odot S_2$. By Proposition 7.4, $\mathcal{A}$ can be chosen as a normalized 0-1-branching wba. Let Q be the state set of $\mathcal{A}$, and let W be the maximal weight sequential transitions take in $\mathcal{A}$. We put $t = a^m \parallel b^n$ where $m > W \cdot |Q|$ and $n > (W-1) \cdot m$. By assumption $(\mathcal{S}_M(\mathcal{A}), t) = m + n$. Since the addition of $\mathbb{P}$ is max and $\mathcal{A}$ is normalized, there is a run $G : p \xrightarrow{t} q$ with p initial, q final, and $\mathbf{wgt}_M(G) = m + n$. G starts with a fork transition $p \rightarrow_0 \{p_1, p_2\}$ and ends with a join transition $\{q_1, q_2\} \rightarrow_0 q$ because $\mathcal{A}$ is 0-1-branching. Moreover, $G' : p_1 \xrightarrow{a^m} q_1$ and $G'' : p_2 \xrightarrow{b^n} q_2$ are runs of $\mathcal{A}$. We get:

$$m + n = \mathrm{nax}(\mathbf{wgt}_M(G'), \mathbf{wgt}_M(G'')) = \max(\mathbf{wgt}_M(G'), \mathbf{wgt}_M(G'')).$$

Now, $n > (W-1) \cdot m$ implies $m \cdot W < m+n$. But $\mathbf{wgt}_M(G') \leq m \cdot W$ because G' is composed of m sequential transitions. Hence, $\mathbf{wgt}_M(G'') = m + n$. The run G'' is a sequence of n sequential transitions. Now, we will show that G'' contains a cycle whose weight is greater than its length.

Claim 1. G'' contains a sub-run $C : r \xrightarrow{b^l} r$ with $\mathbf{wgt}_M(C) > l \geq 1$.

We assume G'' does not contain such a sub-run C, *i.e.* for all sub-runs $C{:}r \xrightarrow{b^l} r$ $(r \in Q)$ of G'' we have $\mathbf{wgt}_M(C) \leq l$. We can give a decomposition of G'' by

$$G'' = C_0 \cdot H_1 \cdot C_1 \cdot H_2 \ldots \cdot C_{k-1} \cdot H_k \cdot C_k \tag{8.25}$$

where $H_1 \cdot \ldots \cdot H_k$ is a run of $\mathcal{A}$ from $r_0 = p_2$ to $r_k = q_2$, and where $C_i : r_i \rightarrow r_i$ is a run of $\mathcal{A}$ for $i = 0, \ldots, k$. Moreover, we may assume that no state used in $H_1 \cdot \ldots \cdot H_k$ occurs twice. Indeed, we choose C_0 maximal so that r_0 does not occur in the remaining of the run, then H_1 is a single transition from r_0 to r_1, and we proceed for $C_1, \ldots, C_k$ similarly. Note that both $H_1 \cdot \ldots \cdot H_k$ and the C_i can be the *empty run* ε_{r_i} *at* r_i, *i.e.* the graph with one vertex labeled by r_i but no edges.[7] This decomposition is illustrated by Figure 8.7.

Figure 8.7: A decomposition of run G''.

Hence, $H_1 \cdot H_2 \cdot \ldots \cdot H_k$ is composed of less than $|Q|$ sequential transitions. Hence, $\mathbf{wgt}_M(H_1 \cdot \ldots \cdot H_k) < W \cdot |Q|$. Since the weight of each cycle is assumed to be less than or equal to the number of its transitions and $m > W \cdot |Q|$, we get

$$\begin{aligned} \mathbf{wgt}_M(G'') =& \sum_{i=0}^{k} \mathbf{wgt}_M(C_i) + \mathbf{wgt}_M(H_1 \cdot \ldots \cdot H_k) \\ <& n + W \cdot |Q| \\ <& n + m \end{aligned}$$

in contradiction to $\mathbf{wgt}_M(G'') = m+n$. Hence, there is a sub-run C:$r \xrightarrow{b^l} r$ of G'' with $l \geq 1$ and $\mathbf{wgt}_M(C) > l$. This proves Claim 1.

By Claim 1, G'' may be decomposed as follows: $G'' = H \cdot C \cdot H'$ where $C : r \xrightarrow{b^l} r$ with $\mathbf{wgt}_M(C) > l$ and H, H' runs or empty runs of $\mathcal{A}$. We put $G^* = H \cdot C \cdot C \cdot H'$ and $\tilde{G} = \parallel_{p,q}(G', G^*)$. Then $\mathbf{wgt}_M(G^*) > \mathbf{wgt}_M(G'')+l$, and we get:

$$\begin{aligned} \mathbf{lab}(\tilde{G}) =& \mathbf{lab}(G') \parallel \mathbf{lab}(G^*) = a^m \parallel b^{n+l}, \quad \text{and} \\ \mathbf{wgt}_M(\tilde{G}) =& \max(\mathbf{wgt}_M(G'), \mathbf{wgt}_M(G^*)) = \mathbf{wgt}_M(G^*) > m + n + l. \end{aligned}$$

Hence, there is a successful run $\tilde{G}$ on $a^m \parallel b^{n+l}$ with $\mathbf{wgt}_M(G') > m+n+l$. But then $\big(\mathcal{S}_M(\mathcal{A}), a^m \parallel b^{n+l}\big) > m + n + l = \big(S_1 \odot S_2, a^m \parallel b^{n+l}\big)$. Hence, $S_1 \odot S_2$ is not M-regular.

[7] The sequential product of runs can be extended to empty runs in a natural way. If $G : p \to q$, then $\varepsilon_p \cdot G = G = G \cdot \varepsilon_q$.

Thus, we showed that M-regular sp-series (even of bounded width) over idempotent, commutative, and distributive bisemirings are not closed under Hadamard product.

9 Series and Languages

In this chapter, we investigate the relations between regular sp-series and sp-languages, especially characteristic series and supports. We will show that there are regular sp-languages with a non-regular characteristic series. But for a restricted class of idempotent bisemirings the characteristic series of a regular language will be regular. The last result states that the restriction of M-regular sp-series to M-regular languages is regular.

An sp-language L is *C-regular* or *M-regular* if there is a wba $\mathcal{A}$ over the Boolean bisemiring $\mathbb{B}$ with $\operatorname{supp} \mathcal{S}_C(\mathcal{A}) = L$ or $\operatorname{supp} \mathcal{S}_M(\mathcal{A}) = L$, respectively. The class of sequential-rational sp-languages was already introduced in Section 3.2.

Let $\mathbb{K}, \mathbb{K}'$ be bisemirings, $h : \mathbb{K} \to \mathbb{K}'$ a bisemiring homomorphism, and $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$. The sp-series $\bar{h}(S) \in \mathbb{K}'\langle\langle \mathrm{SP} \rangle\rangle$ is defined by $(\bar{h}(S), t) = h(S, t)$ for any $t \in \mathrm{SP}$.

Proposition 9.1. *Let $h : \mathbb{K} \to \mathbb{K}'$ be a bisemiring homomorphism. If $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ is C-regular (M-regular), then $\bar{h}(S) \in \mathbb{K}'\langle\langle \mathrm{SP} \rangle\rangle$ is C-regular (M-regular).*

Proof. Let $\mathcal{A} = (Q, \mu_{\mathrm{seq}}, \mu_{\mathrm{fork}}, \mu_{\mathrm{join}}, \lambda, \gamma)$ be a wba over $\mathbb{K}$ recognizing S. Obviously, $\bar{h}\mathcal{A} = (Q, h \circ \mu_{\mathrm{seq}}, h \circ \mu_{\mathrm{fork}}, h \circ \mu_{\mathrm{join}}, h \circ \lambda, h \circ \gamma)$ is a wba over $\mathbb{K}'$ recognizing $\bar{h}(S)$ because h is a bisemiring homomorphism. $\square$

Remember that a bisemiring $\mathbb{K}$ is *positive* if it is zero-sum-free and zero-divisor-free for both products. For a positive bisemiring $\mathbb{K}$ the mapping $h_{\mathbb{B}} : \mathbb{K} \to \mathbb{B}$ with

$$h_{\mathbb{B}}(k) = \begin{cases} 1 & \text{if } k \neq 0, \\ 0 & \text{if } k = 0 \end{cases}$$

is a bisemiring homomorphism.

Lemma 9.2. *Let* $\mathbb{K}$ *be a positive bisemiring.* $L \subseteq \mathrm{SP}$ *is C-regular (M-regular) iff* $L = \operatorname{supp} S$ *for some C-regular (M-regular) sp-series* $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$*.*

Proof. Suppose L is regular. Then there is a wba $\mathcal{A}$ over $\mathbb{B}$ with support $\operatorname{supp} \mathcal{S}(\mathcal{A}) = L$. We consider the same wba $\mathcal{A}$ as a wba $\mathcal{A}_{\mathbb{K}}$ over $\mathbb{K}$. This means if there is a transition in $\mathcal{A}$ there is a transition in $\mathcal{A}_{\mathbb{K}}$ with weight 1, and there are no other transitions in $\mathcal{A}_{\mathbb{K}}$. Moreover, initial and final states are the same as in $\mathcal{A}$, again with weight 1. Since $\mathbb{K}$ is positive, $(\mathcal{S}(\mathcal{A}_{\mathbb{K}}), t) \neq 0$ iff $(\mathcal{S}(\mathcal{A}), t) \neq 0$. Hence, $\operatorname{supp} \mathcal{S}(\mathcal{A}_{\mathbb{K}}) = L$.

Vice versa, let $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ be a regular sp-series with $\operatorname{supp} S = L$. As noted above $h_{\mathbb{B}} : \mathbb{K} \to \mathbb{B}$ is a bisemiring homomorphism. By Proposition 9.1, $\overline{h_{\mathbb{B}}}(S) \in \mathbb{B}\langle\langle \mathrm{SP} \rangle\rangle$ is regular, and $\operatorname{supp} \overline{h_{\mathbb{B}}}(S) = \operatorname{supp} S = L$. Hence, L is a regular sp-language. □

With Theorem 6.2 and Corollary 6.4 we get immediately:

Corollary 9.3. *Let* $\mathbb{K}$ *be a positive bisemiring.* $L \subseteq \mathrm{SP}$ *is sequential-rational iff* $L = \operatorname{supp} S$ *for some sequential-rational sp-series* $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$*.*

Next, we state a result about the support of the product automaton already indicated in Chapter 8.

Lemma 9.4. *Let* $\mathbb{K}$ *be a positive bisemiring,* $\mathcal{A}_1, \mathcal{A}_2$ *two wba over* $\mathbb{K}$*, and* $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$*. Then*

$$\operatorname{supp} \mathcal{S}_M(\mathcal{A}) = \operatorname{supp}[\mathcal{S}_M(\mathcal{A}_1)] \cap \operatorname{supp}[\mathcal{S}_M(\mathcal{A}_2)].$$

Proof. Let $S_i = \mathcal{S}_M(\mathcal{A}_i)$ for $i = 1, 2$ and $S = \mathcal{S}_M(\mathcal{A})$. Now, we put $T_i = \overline{h_{\mathbb{B}}}(S_i)$ for $i = 1, 2$ and $T = \overline{h_{\mathbb{B}}}(S)$. T_1, T_2 and T are sp-series over the idempotent doubled semiring $\mathbb{B}$. Moreover, T_i is M-recognized by the wba $\overline{h_{\mathbb{B}}}\mathcal{A}_i$ for $i = 1, 2$, and T by $\overline{h_{\mathbb{B}}}\mathcal{A} = \overline{h_{\mathbb{B}}}\mathcal{A}_1 \times \overline{h_{\mathbb{B}}}\mathcal{A}_2$. By Corollary 8.7,

$$\operatorname{supp} T = \operatorname{supp}(T_1 \odot T_2) = \operatorname{supp} T_1 \cap \operatorname{supp} T_2.$$

Clearly, $\operatorname{supp} S = \operatorname{supp} T$ and $\operatorname{supp} S_i = \operatorname{supp} T_i$ for $i = 1, 2$ because $\mathbb{K}$ is positive. Hence, $\operatorname{supp} S = \operatorname{supp} S_1 \cap \operatorname{supp} S_2$. □

The reversed concept to the support is the characteristic series of an sp-language. Let $L \subseteq \mathrm{SP}$. The *characteristic series* $\mathbb{1}_L : \mathrm{SP} \to \mathbb{K}$ of L over $\mathbb{K}$ is defined by

$$(\mathbb{1}_L, t) = \begin{cases} 1 & \text{if } t \in L, \\ 0 & \text{otherwise.} \end{cases}$$

The next example is concerned with a characteristic series over the doubled semiring $\mathbb{N}$. It shows that the characteristic series $\mathbb{1}_L$ of a regular language L does not have to be regular.

Example 9.5. Let Σ be a finite alphabet, and $\mathbb{N} = (\mathbb{N}, +, \cdot, \cdot, 0, 1)$ the doubled semiring of the natural numbers with the usual addition and multiplication. $L = (\Sigma^+) \parallel (\Sigma^+)$ is a regular sp-language because it is sequential-rational. Let $\mathbb{1}_L$ be the characteristic series of L over $\mathbb{N}$. Assume there is a wba $\mathcal{A}$ over $\mathbb{N}$ recognizing $\mathbb{1}_L$. Consider the infinite set $W = \{w \parallel w \mid w \in \Sigma^+\} \subseteq L$. Hence, each $w \parallel w \in L$ has a successful run in $\mathcal{A}$ limited by some matching pair. For a fixed matching pair (f, j) with $f : p \to \{p_1, p_2\}$ and $j : \{q_1, q_2\} \to q$, there are two possibilities for the sub-runs on w: either $G_1 : p_1 \xrightarrow{w} q_1$, $G_2 : p_2 \xrightarrow{w} q_2$, or $G_1 : p_1 \xrightarrow{w} q_2$, $G_2 : p_2 \xrightarrow{w} q_1$. Hence, the set of matching pairs and of the possibilities for sub-runs on w is finite. But since W is infinite, there are $u, v \in \Sigma^+$ with $u \neq v$ such that

- there is a matching pair (f, j) with $f : p \to \{p_1, p_2\}$, $j : \{q_1, q_2\} \to q$, p initial, q final, and
- there are runs $G_i : p_i \xrightarrow{u} q_i$ and $H_i : p_i \xrightarrow{v} q_i$ in $\mathcal{A}$ for $i = 1, 2$.

But then $F = \parallel_{p,q}(G_1, H_2)$ and $F' = \parallel_{p,q}(G_2, H_1)$ are two different successful runs of $\mathcal{A}$ on $u \parallel v$. Hence,

$$\begin{aligned} (\mathcal{S}(\mathcal{A}), u \parallel v) &\geq \mathbf{wgt}(F) + \mathbf{wgt}(F') \\ &\geq 2 \\ &> (\mathbb{1}_L, u \parallel v) \end{aligned}$$

because $\mathbb{N}$ is zero-sum-free. Thus, $\mathcal{A}$ does not recognize $\mathbb{1}_L$.

Nevertheless, for a certain class of idempotent bisemirings characteristic series of regular languages are regular.

Theorem 9.6. *Let $L \subseteq \mathrm{SP}$ be C-regular (M-regular), and $\mathbb{K}$ an idempotent bisemiring with $1 \Diamond 1 = 1$. Then $\mathbb{1}_L$ over $\mathbb{K}$ is C-regular (M-regular).*

Proof. There is a wba $\mathcal{A}$ over $\mathbb{B}$ with $\operatorname{supp} \mathcal{S}(\mathcal{A}) = L$. Let $h : \mathbb{B} \to \mathbb{K}$ be the mapping with $0_{\mathbb{B}} \mapsto 0_{\mathbb{K}}$ and $1_{\mathbb{B}} \mapsto 1_{\mathbb{K}}$. We have:

- $h(1_{\mathbb{B}} \oplus 1_{\mathbb{B}}) = h(1_{\mathbb{B}}) = 1_{\mathbb{K}} = 1_{\mathbb{K}} \oplus 1_{\mathbb{K}} = h(1_{\mathbb{B}}) \oplus h(1_{\mathbb{B}})$ because $\mathbb{K}$ is idempotent,
- $h(1_{\mathbb{B}} \circ 1_{\mathbb{B}}) = 1_{\mathbb{K}} = h(1_{\mathbb{B}}) \circ h(1_{\mathbb{B}})$, and
- $h(1_{\mathbb{B}} \Diamond 1_{\mathbb{B}}) = 1_{\mathbb{K}} = 1_{\mathbb{K}} \Diamond 1_{\mathbb{K}} = h(1_{\mathbb{B}}) \Diamond h(1_{\mathbb{B}})$.

Hence, h is a bisemiring homomorphism. By Proposition 9.1, $\mathbb{1}_L = \bar{h}(\mathcal{S}(\mathcal{A}))$ is regular. ☐

The *restriction* $S_{|L} \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ *of an sp-series* $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ *to some sp-language* L is defined by

$$(S_{|L}, t) = \begin{cases} (S, t) & \text{if } t \in L, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 9.7. *Let $\mathbb{K}$ be an idempotent bisemiring, $S \in \mathbb{K}\langle\langle \mathrm{SP} \rangle\rangle$ M-regular, and L an M-regular sp-language. Then $S_{|L}$ is M-regular.*

Proof. Let $\mathcal{A}_1$ be a wba M-recognizing S, and $\mathcal{A}_L$ a wba over $\mathbb{B}$ with $\operatorname{supp} \mathcal{S}_M(\mathcal{A}_L) = L$. Let $h : \mathbb{B} \to \mathbb{K}$ be the mapping with $0_{\mathbb{B}} \mapsto 0_{\mathbb{K}}$ and $1_{\mathbb{B}} \mapsto 1_{\mathbb{K}}$. We put $\mathcal{A}_2 = \bar{h}\mathcal{A}_L$.[1] Let $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$.

Now, we proceed similarly as in the proof of Lemma 8.4.

Claim 1. Let $p^1, q^1 \in Q_1$, $p^2, q^2 \in Q_2$, and $t \in \mathrm{SP}$. Then

$$\begin{aligned} &\mathbf{wgt}_M\left(\begin{pmatrix} p^1 \\ p^2 \end{pmatrix}, t, \begin{pmatrix} q^1 \\ q^2 \end{pmatrix}\right) \\ &= \begin{cases} \mathbf{wgt}^1_M(p^1, t, q^1) & \text{if } \mathbf{wgt}_M(p^2, t, q^2) = 1_{\mathbb{B}} \text{ in } \mathcal{A}_L, \\ 0_{\mathbb{K}} & \text{otherwise.} \end{cases} \end{aligned}$$

[1] Note that h is not necessarily a homomorphism and, therefore, in general not $\mathcal{S}_M(\mathcal{A}_2) = \mathbb{1}_L$.

For $t = a \in \Sigma$, Claim 1 follows immediately from the definition of $\mathcal{A}$. Let $t = t_1 \cdot \ldots \cdot t_m$ ($m \geq 2$) be sequential in its sequential decomposition. Then we get:

$$\begin{aligned}
&\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \\
&= \bigoplus_{\substack{G_1,\ldots,G_m: \\ G=G_1\cdot\ldots\cdot G_m, \\ \mathbf{lab}(G_i)=t_i}} \mathbf{wgt}(G_1) \circ \ldots \circ \mathbf{wgt}(G_m) \\
&= \bigoplus_{\left[\binom{r_1^1}{r_1^2},\ldots,\binom{r_{m-1}^1}{r_{m-1}^2}\right] \in Q^{m-1}} \mathbf{wgt}_M\left(\binom{p^1}{p^2}, t_1, \binom{r_1^1}{r_1^2}\right) \circ \ldots \\
&\qquad\qquad \circ \mathbf{wgt}_M\left(\binom{r_{m-1}^1}{r_{m-1}^2}, t_m, \binom{q^1}{q^2}\right)
\end{aligned}$$

We put $r_0^i = p^i$ and $r_m^i = q^i$ for $i = 1, 2$. By induction hypothesis, we get:

$$\begin{aligned}
&\mathbf{wgt}_M\left(\binom{r_i^1}{r_i^2}, t_{i+1}, \binom{r_{i+1}^1}{r_{i+1}^2}\right) \\
&\qquad = \begin{cases} \mathbf{wgt}_M^1(r_i^1, t_{i+1}, r_{i+1}^1) & \text{if } \mathbf{wgt}_M(r_i^2, t_{i+1}, r_{i+1}^2) = 1_{\mathbb{B}} \text{ in } \mathcal{A}_L, \\ 0_{\mathbb{K}} & \text{otherwise} \end{cases}
\end{aligned}$$

for $i = 0, \ldots, m-1$. By idempotency of $\mathbb{K}$, we conclude

$$\begin{aligned}
&\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \\
&\qquad = \bigoplus_{(r_1^1,\ldots,r_{m-1}^1) \in Q_q^{m-1}} \mathbf{wgt}_M^1(p^1, t_1, r_1^1) \circ \ldots \circ \mathbf{wgt}_M^1(r_{m-1}^1, t_m, q^1)
\end{aligned}$$

if there are $(r_1^2, \ldots, r_{m-1}^2) \in Q_L^{m-1}$ such that there is a run $p^2 \xrightarrow{t_1} r_1^2 \xrightarrow{t_2} \ldots \xrightarrow{t_m} q^2$, *i.e.* $\mathbf{wgt}_M(p^2, t, q^2) = 1_{\mathbb{B}}$ in $\mathcal{A}_L$. On the other hand, if $\mathbf{wgt}_M(p^2, t, q^2) = 0_{\mathbb{B}}$ in $\mathcal{A}_L$, then such a run does not exist and

$$\mathbf{wgt}_M\left(\binom{r_i^1}{r_i^2}, t_{i+1}, \binom{r_{i+1}^1}{r_{i+1}^2}\right) = 0_{\mathbb{K}}$$

for some $i \in \{0, \ldots, m-1\}$. In this case, $\mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) = 0_{\mathbb{K}}$. This shows Claim 1 for $t = t_1 \cdot \ldots \cdot t_m$.

For $t = t_1 \parallel \ldots \parallel t_m$ $(m \geq 2)$ we proceed similarly but use Equations (8.1), (8.3), and (8.4) from the proof of Lemma 8.4. We omit the details.

Now we have:

$$(\mathcal{S}(\mathcal{A}), t) = \bigoplus_{\binom{p^1}{p^2}, \binom{q^1}{q^2} \in Q} \lambda\binom{p^1}{p^2} \circ \mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \circ \gamma\binom{q^1}{q^2}.$$

By Claim 1 and the construction of $\mathcal{A}$ we conclude

$$\lambda\binom{p^1}{p^2} \circ \mathbf{wgt}_M\left(\binom{p^1}{p^2}, t, \binom{q^1}{q^2}\right) \circ \gamma\binom{q^1}{q^2}$$

$$= \begin{cases} \bigoplus\limits_{p^1, q^1 \in Q_1} \lambda_1(p^1) \circ \mathbf{wgt}^1_M(p^1, t, q^1) \circ \gamma_1(q^1) & \text{if } p^2 \text{ initial, } q^2 \text{ final, and} \\ & \mathbf{wgt}_M(p^2, t, q^2) = 1_{\mathbb{B}} \\ & \text{in } \mathcal{A}_L, \\ 0_{\mathbb{K}} & \text{otherwise.} \end{cases}$$

Hence, by idempotency of $\mathbb{K}$:

$$(\mathcal{S}_M(\mathcal{A}), t) = \begin{cases} (\mathcal{S}_M(\mathcal{A}_1), t) & \text{if } t \in L, \\ 0_{\mathbb{K}} & \text{otherwise.} \end{cases}$$

Therefore, $S_{|L}$ is M-regular. □

10 Conclusion

Maybe the most fascinating aspect about Schützenberger's theorem on the coincidence of regular and rational formal power series over words [Sch61b] is its universal validity. It holds true for every underlying semiring. Therefore, we are gratified that our corresponding result for series over sp-posets and bisemirings is of the same universality. Moreover, it could be obtained by elementary constructions true to the original spirit of Kleene. No doubt, we paid a price by using sometimes rather sophisticated constructions and quite technical proofs. In the literature other proofs of Schützenberger's result than combinatorial ones can be found. Mainly, they base on a matrix representation of weighted automata [SS78, BR88, KS86]. If $\mathbb{K}$ is the underlying semiring and Q the state set of the automaton then a matrix $\mu(a) \in \mathbb{K}^{Q \times Q}$ can be associated to every $a \in \Sigma$. This mapping is extended to a monoid homomorphism $\mu : \Sigma^\star \to \mathbb{K}^{Q \times Q}$. Combined with the initial and final weight function, a weighted automaton can be seen as a finitely generated semimodule together with a linear form. The initial state vector $\lambda \in \mathbb{K}^Q$ is a fixed element m_0 of the semimodule on which an endomorphism $\mu(w)$ is acting, resulting in another element $m = \mu(w)(m_0)$. Finally, the linear form $\gamma \in \mathbb{K}^Q$, given by the final weights, is applied to m and yields an element of $\mathbb{K}$. Thus, Berstel and Reutenauer [BR88] use stable finitely generated submodules to prove Schützenberger's theorem. For $\mathbb{K}$ a commutative ring, Reutenauer defines even an analogue to the syntactic monoid of a formal language: the syntactic algebra [Reu80]. For $\mathbb{K}$ a field, these results are used to determine a reduced linear representation of a regular series which is the analogue to the minimal automaton of a formal language. The reduced linear representation of a rational series was already studied by Schützenberger [Sch61b, Sch61a]. This raises the question which other methods could have been applied to studying rational series over sp-posets and bisemirings than pure combinatorial ones. Unfortunately, no straightforward generalization of the matrix representation of word series is possible.

Certainly, a weighted branching automaton can also be understood as a finitely generated $\mathbb{K}$-semimodule where the scalar multiplication is the sequential multiplication. This is because initial and final weights are multiplied sequentially. Every $a \in \Sigma$ determines $\mu_{\text{seq}}(a) \in \mathbb{K}^{Q \times Q}$ which can be seen as an endomorphism of the $\mathbb{K}$-semimodule. Sequential composition of letters translates into the usual composition of endomorphisms. But what about parallel composition? The first problem arises from the fact that the cascade branching mode does not reflect the associativity of the parallel product. But even when overcoming this problem by considering the maximally branching mode, what kind of endomorphism is, for example, defined by $a \parallel b$? Firstly, $a \parallel b$ acts actually on two states instead of one. Hence, we had to switch to a higher dimensional semimodule by the fork transitions, let $a \parallel b$ act in this higher dimensional semimodule, and come back to the original semimodule by join transitions. The acting of $a \parallel b$ could be seen as the Kronecker product of $\mu_{\text{seq}}(a)$ and $\mu_{\text{seq}}(b)$ but this time with respect to the parallel product. Hence, we switch to a semimodule with another scalar operation. For semirings there is a theory of semimodules and linear mappings. This theory is even more powerful when the semiring is actually a ring or a field. Then all the apparatus of ring and field theory and their linear spaces is at our disposal. This is different with bisemirings because now we have to deal with two products. Possibly, an approach of "linked semimodules" as indicated above, could be successful. Here, it may also be of interest to explore the work done by Bozapalidis *et al.* for tree series. In [BA89] a matrix representation for tree series $S : T_\Sigma \to \mathbb{K}$ where $\mathbb{K}$ is a field was defined and the equivalence of the notions of representable and recognizable tree series was shown. This is not true for arbitrary semirings. Nevertheless, in [Boz94] a characterization of representable tree series by finitely generated stable submodules was reached for arbitrary semirings. The notion of representable tree series may be of interest for sp-series because multilinear functions were used. However, the problem of two scalar multiplications arising from a bisemiring remains. A more algebraic understanding of regular sp-series is subject of further research. Altogether, the basic combinatorial approach we have chosen seems to be the most self-evident so far.

Another approach of giving a Myhill-Nerode-like characterization of regular series over words and trees is by using congruences. This was done for instance by Mohri [Moh97] for word series and Borchardt [Bor03] for tree series. However, those approaches use deterministic devices. The concept of determinism is not really explored even for branching automata without weights. In [LW01, Rem. 3.6] Lodaya and Weil consider the free Σ-algebra with an additional associative operation over some alphabet A, called a series-Σ-algebra $S_\Sigma(A)$. There, they note that the construction of an appropriate branching automaton out of a finite series-Σ-algebra yields something like a deterministic complete device, in the sense that for every element $x \in S_\Sigma(A)$ the automaton has one and only one run on x starting at every state. But this global condition does not translate into a local one for branching automata over sp-posets. This is due to the commutativity of the parallel product. As we have seen in Example 9.5 the regular sp-language $(\Sigma^+) \parallel (\Sigma^+)$ is not recognized by an unambiguous branching automaton. So far it is not understood what concept of determinism could be useful for branching automata over sp-posets.

Obviously, our results are generalizations of those achieved by Lodaya and Weil for sp-languages of bounded width [LW00]. Further on, any semiring may be seen as a bisemiring by adding a trivial parallel multiplication which is identical zero. By considering a weighted branching automaton over such a semiring without any fork and join transitions we get almost a classical weighted automaton over words, but not entirely. Usually, weighted word automata may contain the empty word ε within their support. If $\lambda, \gamma \in \mathbb{K}^Q$ are the initial and final weight functions the weight of ε is obtained by

$$\bigoplus_{p,q \in Q} \lambda(p) \circ \gamma(q).$$

The iteration and the Kleene star operation are then restricted to proper series, *i.e.* those without ε in their support, in order to be well-defined. Since we excluded ε from our considerations, our results are not full generalizations of the word case and, especially, of Schützenberger's theorem. However, there are good reasons to consider proper sp-series only. If ε is allowed in the support we would implicitly include a parallel scalar multiplication by parallel Cauchy product. Therefore, we would have also to show closure of regular sp-series under parallel scalar products. This could be fairly done by ε-transitions only which would make the whole model as well as the proofs much more complicated. Another alternative would be the restriction of the parallel product to proper sp-series. Hence, we prefer to study only proper sp-series from the very beginning.

Our main result, Theorem 6.2, holds for sp-series of bounded width. Lodaya and Weil [LW98, LW01] gave a rational characterization of regular sp-languages in general without a boundedness assumption for the width of the language. For this, they applied another concept of generalized rationality closer to that of tree languages and elsewhere also known as equationally defined languages. Instead of a parallel iteration they considered ξ-substitution and ξ-exponentiation. To ensure the closure of regular sp-languages under ξ-exponentiation they had to restrict the application of the exponentiation. They succeeded in [LW01] with such a characterization which there was considered for languages of term algebras with an additional associative operation. Afterwards, this result was "projected" to sp-languages. Is this concept of generalized rationality also suitable for arbitrary regular sp-series over bisemirings? Not in general, as it seems. The main obstacle is to define ξ-substitution for sp-series. Obviously, it is no problem to substitute the support of one series into that of another one. But what about weights? Having in mind that we want to show the closure of regular sp-series under ξ-substitution we would plug in one automaton at all places of the other one where a sequential transition labeled by ξ occurs. But then the behavior of the resulting automaton very much depends on the place where ξ appears because the way the weight of some $t \in \mathrm{SP}$ is calculated reflects the structure of t. Therefore, it does not make sense to define substitution for instance by multiplying sequentially all weights involved. Nevertheless, we could succeed if the bisemiring is actually a doubled commutative semiring because then the calculation can be done in any order with one multiplication only. For bisemirings in general one could perhaps show a Kleene-like result for "weighted sp-languages" similar to weighted tree languages as introduced by Pech [Pec03a]. There, the nodes of a tree are labeled with weights instead of computing one weight for the whole tree. However, this does not yet solve the problem of defining a suitable substitution for sp-series. Nevertheless, this is a promising line of further research.

As already noted, it would be fine to get a more algebraic characterization for rational and regular sp-series. It may be convenient firstly to confine to the case of two associative operations and forgetting about commutativity even if this way we leave the subject of concurrency. Ésik and Németh [ÉN02b, ÉN02a] and Hashigushi *et al.* [HIJ00] explored languages of biposets that are built of singletons by two independent associative operations. Ésik and Németh defined finite-state devices for accepting those languages. These so-called parenthesizing automata are composed mainly of two classical finite-state automata linked by parenthesizing transitions. Among other results, for the class of languages of bounded alternation depth Ésik and Németh also get the equivalence of birationality and regularity. Their concept of birationality corresponds to our concept of rationality. Moreover, they obtain similar results as Lodaya and Weil for series-rational and parallel-rational languages where the application of the parallel iteration, and the sequential iteration respectively, is forbidden. Since the model of parenthesizing automata is more balanced and in our opinion closer to classical finite automata than that of branching automata, an enrichment of parenthesizing automata with weights may deepen the understanding of models of weighted automata with several compositions.

Another way of broadening our results is the extension to infinite sp-posets. Here, Kuske [Kus03] introduced branching Büchi-automata and extended the results of Lodaya and Weil to languages of infinite sp-posets. When adding weights to these automata the problem arises that infinite products of weights must be handled. Then either one has to demand completeness properties for the bisemiring or one makes use of a deflation parameter. The latter one guarantees that the further an event is in the future the less its weight counts. This was done by Droste and Kuske [DK03] for weighted Büchi automata over words with values in the real max-plus semiring. In a different approach, Ésik and Kuich [ÉK03a, ÉK03b] used complete semirings with infinitary sum and product operations to overcome this problem. Something similar could be investigated for weighted Büchi automata over infinite sp-posets.

The basic idea of this work was that different compositions of processes may give rise to different compositions of their weights. This situation cannot be handled by one semiring alone. Therefore, we introduced bisemirings which arise naturally when considering sp-posets. We achieved first results for finite-state devices over sp-posets with weights from bisemirings, the weighted branching automata. However, as we hope to have indicated in this conclusion, such systems are by far not fully understood, especially in an algebraic sense. This may initiate more fruitful research within this area.

Bibliography

[BA89] S. Bozapalidis and A. Alexandrakis, *Représentations matricielles des séries d'arbre reconnaissables*, Informatique Théorique et Applications **23** (1989), no. 4, 449–459.

[BCOQ92] F.L. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity – An Algebra for Discrete Event Systems*, John Wiley & Sons, 1992.

[BÉ96] S. Bloom and Z. Ésik, *Free shuffle algebras in language varieties*, Theoret. Comp. Sc. **163** (1996), 55–98.

[BÉ03] S.L. Bloom and Z. Ésik, *An extension theorem with an application to formal tree series*, J. of Automata, Languages and Combinatorics **8** (2003), no. 2, 145–185.

[Bir73] G. Birkhoff, *Lattice Theory*, 3 ed., American Mathematical Society, 1973, 1 ed. 1940.

[BLB83] S. Bozapalidis and O. Louscou-Bozapalidou, *The rank of a formal tree power series*, Theoret. Comp. Sc. **27** (1983), 211–215.

[Bor03] B. Borchardt, *The Myhill-Nerode theorem for recognizable tree series*, Proc. of Develop. in Lang. Theory (7th DLT), Lecture Notes in Comp. Sc., vol. 2710, Springer, 2003, pp. 146–158.

[Boz91] S. Bozapalidis, *Effective construction of the syntactic algebra of a recognizable series on trees*, Acta Informatica **28** (1991), 351–363.

[Boz94] S. Bozapalidis, *Representable tree series*, Fundamenta Informaticae **21** (1994), 367–389.

[BR82] J. Berstel and C. Reutenauer, *Recognizable formal power series on trees*, Theoret. Comp. Sc. **18** (1982), 115–148.

[BR88] J. Berstel and C. Reutenauer, *Rational Series and Their Languages*, EATCS Monographs on Theoret. Comp. Sc., vol. 12, Springer, 1988.

[CGQ99] G. Cohen, S. Gaubert, and J.P. Quadrat, *Max-plus algebra and system theory: where we are and where to go now*, Annual Reviews in Control **23** (1999), 207–219.

[CK93] K. Culik and J. Kari, *Image compression using weighted finite automata*, Computer and Graphics **17** (1993), no. 3, 305–313.

[CK97] K. Culik and J. Kari, *Digital images and formal languages*, Handbook of Formal Languages: Beyond Words (G. Rozenberg and A. Salomaa, eds.), vol. 3, Springer, 1997, pp. 599–616.

[DG99] M. Droste and P. Gastin, *The Kleene-Schützenberger theorem for formal power series in partially commuting variables*, Information and Computation **153** (1999), 47–80.

[DG00] M. Droste and P. Gastin, *On aperiodic and star-free formal power series in partially commuting variables*, Formal Power Series and Algebraic Combinatorics (Moscow 2000), Springer Berlin, 2000, pp. 158–169.

[DGK00] M. Droste, P. Gastin, and D. Kuske, *Asynchronous cellular automata for pomsets*, Theoret. Comp. Sc. **247** (2000), 1–38.

[DK03] M. Droste and D. Kuske, *Skew and infinitary formal power series*, Automata Languages and Programming (30th ICALP, Eindhoven), Lecture Notes in Comp. Sc., vol. 2719, Springer, 2003, pp. 426–438.

[DPV04] M. Droste, Ch. Pech, and H. Vogler, *A Kleene theorem for weighted tree automata*, Theory of Computing Systems (2004), to appear.

[DR95] V. Diekert and G. Rozenberg (eds.), *The Book of Traces*, World Scientific, 1995.

[Dro92] M. Droste, *Concurrent automata and domains*, Intern. J. of Found. of Computer Sc. **3** (1992), 389–418.

[Eil74] S. Eilenberg, *Automata, Languages, and Machines*, vol. A, Academic Press, 1974.

[ÉK03a] Z. Ésik and W. Kuich, *A semiring-semimodule generalization of ω-regular languages*, Tech. Report I, Technische Universität Wien, 2003.

[ÉK03b] Z. Ésik and W. Kuich, *A semiring-semimodule generalization of ω-regular languages*, Tech. Report II, Technische Universität Wien, 2003.

[ÉN02a] Z. Ésik and Z. L. Németh, *Higher dimensional automata*, Tech. Report RS-02-24, BRICS, November 2002.

[ÉN02b] Z. Ésik and Z.L. Németh, *Automata on series-parallel biposets*, Proc. of Develop. in Lang. Theory 2001 (W. Kuich, G. Rozenberg, and A. Salomaa, eds.), Lecture Notes in Comp. Sc., vol. 2295, Springer, 2002, pp. 217–227.

[Gis88] J.L. Gischer, *The equational theory of pomsets*, Theoret. Comp. Sc. **61** (1988), 199–224.

[Gol99] J.S. Golan, *Semirings and their Applications*, Kluwer Academic Publishers, 1999.

[GP97] S. Gaubert and M. Plus, *Methods and applications of* $(\max, +)$ *linear algebra*, Tech. Report 3088, INRIA, Rocquencourt, January 1997.

[Gra81] J. Grabowski, *On partial languages*, Ann. Soc. Math. Pol. IV: Fundamenta Informaticae **4** (1981), no. 2, 427–498.

[Gun98] J. Gunawardena (ed.), *Idempotency*, Publications of the Newton Institute, Cambridge University Press, 1998.

[Haf99] U. Hafner, *Low bit-rate image and video coding with weighted finite automata*, Dissertation, Universität Würzburg, Germany, 1999.

[HIJ00] K. Hashiguchi, S. Ichihara, and S. Jimbo, *Formal languages over free binoids*, J. of Automata, Languages and Combinatorics **5** (2000), no. 3, 219–234.

[HMU00] J.E. Hopcroft, R. Motwani, and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed., Pearson Addison Wesley, 2000.

[HW99] U. Hebisch and H.J. Weinert, *Semirings: Algebraic Theory and Application*, World Scientific, 1999.

[Kat01] F. Katritzke, *Refinements of data compression using weighted finite automata*, Dissertation, Universität Siegen, Germany, 2001.

[Kle56] S.E. Kleene, *Representations of events in nerve nets and finite automata*, Automata Studies (C.E. Shannon and J. McCarthy, eds.), Princeton University Press, 1956, pp. 3–42.

[KM03] D. Kuske and I. Meinecke, *Branching automata with costs – a way of reflecting parallelism in costs*, Implementation and Application of Automata (8th CIAA 2003) (O. H. Ibarra and Z. Dang, eds.), Lecture Notes in Comp. Sc., vol. 2759, Springer, 2003, pp. 150–162.

[KN01] B. Khoussainov and A. Nerode, *Automata Theory and its Applications*, Progress in Computer Sc. and Applied Logic, vol. 21, Birkhäuser Boston, 2001.

[KS86] W. Kuich and A. Salomaa, *Semirings, Automata, Languages*, EATCS Monographs on Theoret. Comp. Sc., vol. 5, Springer, 1986.

[Kui97a] W. Kuich, *Formal power series over trees*, Proc. of the 3rd International Conference Developments in Language Theory, Aristotle University of Thessaloniki, 1997, pp. 60–101.

[Kui97b] W. Kuich, *Semirings and Formal Power Series*, Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds.), vol. 1, Springer, 1997, pp. 609–677.

[Kui99] W. Kuich, *Linear systems of equations and automata on distributive multioperator monoids*, Contributions to General Algebra (Proceedings of the Vienna Conference), no. 12, Verlag Johannes Heyn, 1999, pp. 1–10.

[Kui00] W. Kuich, *Formal series over algebras*, Math. Found. of Comp. Sc. (25th MFCS Bratislava), Lecture Notes in Comp. Sc., vol. 1893, Springer, 2000, pp. 488–496.

[Kus03] D. Kuske, *Towards a language theory for infinite N-free pomsets*, Theoret. Comp. Sc. **299** (2003), 347–386.

[LW98] K. Lodaya and P. Weil, *A Kleene iteration for parallelism*, Proc. FST & TCS 98, Lecture Notes in Comp. Sc., vol. 1530, Springer, 1998, pp. 355–366.

[LW00] K. Lodaya and P. Weil, *Series-parallel languages and the bounded-width property*, Theoret. Comp. Sc. **237** (2000), 347–380.

[LW01] K. Lodaya and P. Weil, *Rationality in algebras with a series operation*, Inform. and Comput. **171** (2001), 269–293.

[Maz87] A. Mazurkiewicz, *Trace theory*, Advances in Petri Nets 1986, Part II on Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes in Comp. Sc., vol. 255, Springer, 1987, pp. 279–324.

[Moh97] M. Mohri, *Finite-state transducers in language and speech processing*, Computational Linguistics **23** (1997), no. 2, 269–311.

[MPR00] M. Mohri, F. Pereira, and M. Riley, *The design principles of a weighted finite-state transducer library*, Theoret. Comp. Sc. **231** (2000), 17–32.

[NW95] M. Nielsen and G. Winskel, *Models for Concurrency*, Handbook of Logic in Computer Science: Semantic Modelling (S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, eds.), vol. 4, Oxford University Print, 1995, pp. 1–148.

[Pec03a] Ch. Pech, *Kleene-type results for weighted tree-automata*, Ph.D. thesis, TU Dresden, 2003.

[Pec03b] Ch. Pech, *Kleene's theorem for weighted tree-automata*, Fundamentals of Computation Theory (A. Lingas and B.J. Nilsson, eds.), Lecture Notes in Comp. Sc., vol. 2751, Springer, 2003, pp. 387–399.

[Pra86] V. Pratt, *Modelling concurrency with partial orders*, Int. J. of Parallel Programming **15** (1986), 33–71.

[Reu80] Ch. Reutenauer, *Séries formelles et algébres syntactiques*, J. of Algebra **66** (1980), 448–483.

[Sch61a] M.P. Schützenberger, *On a special class of recurrents events*, Annals Math. Stat. **32** (1961), 1201–1213.

[Sch61b] M.P. Schützenberger, *On the definition of a family of automata*, Information and Control **4** (1961), 245–270.

[Sco82] D. Scott, *Domains for denotational semantics*, Automata, Languages, and Programming (9th ICALP), Lecture Notes in Comp. Sc., vol. 140, Springer, 1982, pp. 577–613.

[SGG04] M.K. Sen, Sh. Ghosh, and S. Ghosh, *An introduction to bisemirings*, Preprint, University of Calcutta, Jadavpur University, 2004.

[SS78] A. Salomaa and M. Soittola, *Automata-Theoretic Aspects of Formal Power Series*, Texts and Monographs in Computer Science, Springer, 1978.

[TW68] J.W. Thatcher and J.B. Wright, *Generalized finite automata theory with application to a decision problem of second-order logic*, Math. Systems Theory **2** (1968), no. 1, 57–81.

[Wec92] W. Wechler, *Universal Algebra for Computer Scientists*, EATCS Monographs on Theoret. Comp. Sc., vol. 25, Springer, 1992.

[Win87] G. Winskel, *Event structures*, Advances in Petri Nets 1986, Part II on Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes in Comp. Sc., vol. 255, Springer, 1987, pp. 325–392.

[Zie87] W. Zielonka, *Notes on finite asynchronous automata*, R.A.I.R.O. – Informatique Théorique et Applications **21** (1987), 99–135.

[Zie89] W. Zielonka, *Safe executions of recognizable trace languages by asynchronous automata*, Logical Foundations of Computer Science (A.R. Meyer et al., ed.), Lecture Notes in Comp. Sc., vol. 363, Springer, 1989, pp. 278–289.

Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.

Die Promotionsordnung der Fakultät Mathematik und Naturwissenschaften der Technischen Universität Dresden vom 20. März 2000 erkenne ich an.

Die vorliegende Dissertation wurde an der Technischen Universität Dresden unter der wissenschaftlichen Betreuung von Prof. Dr. Manfred Droste angefertigt.

Affirmation

I affirm that I have written this dissertation without any inadmissible help from any third person and without recourse to any other aids; all sources are clearly referenced. The dissertation has never been submitted in this or similar form before, neither in Germany nor in any foreign country.

I accept the regulations for the conferral of a doctorate of the Faculty of Mathematics and Sciences of Dresden University of Technology (published on March, 20th, 2000).

I have written this dissertation at Dresden University of Technology under the scientific supervision of Prof. Dr. Manfred Droste.

Dresden, 29. Juni 2004 Ingmar Meinecke