Weighted Logics and Weighted Simple Automata for Context-Free Languages of Infinite Words

Sven Dziadek

Weighted Logics and Weighted Simple Automata for Context-Free Languages of Infinite Words

Der Fakultät für Mathematik und Informatik
der Universität Leipzig
angenommene

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

zur Erlangung des akademischen Grades

Doctor rerum naturalium
(Dr. rer. nat.)

im Fachgebiet

Informatik

vorgelegt von

M. Sc. Sven Dziadek
geboren am 04. August 1989 in Ludwigshafen am Rhein

Die Annahme der Dissertation wurde empfohlen von:

1. Prof. Dr. Dr. h.c. Manfred Droste (Universität Leipzig)
   Prof. Dr. Dr. h.c. Werner Kuich (Technische Universität Wien)
2. Prof. Dr. Juha Honkala (University of Turku, Finnland)

Die Verleihung des akademischen Grades erfolgt mit Bestehen der Verteidigung am 21. Dezember 2020 mit dem Gesamtprädikat *summa cum laude*.

Abstract

Büchi, Elgot and Trakhtenbrot provided a seminal connection between monadic second-order logic and finite automata for both finite and infinite words. This BET-Theorem has been extended by Lautemann, Schwentick and Thérien to context-free languages by introducing a monadic second-order logic with an additional existentially quantified second-order variable. This new variable models the stack of pushdown automata. A fundamental study by Cohen and Gold extended the context-free languages to infinite words. *Our first main result* is a second-order logic in the sense of Lautemann, Schwentick and Thérien with the same expressive power as ω-context-free languages. For our argument, we investigate Greibach normal forms of ω-context-free grammars as well as a new type of Büchi pushdown automata, the simple pushdown automata. Simple pushdown automata do not use ϵ-transitions and can change the stack only by at most one symbol. We show that simple pushdown automata of infinite words suffice to accept all ω-context-free languages. This enables us to use Büchi-type results recently developed for infinite nested words.

In weighted automata theory, many classical results on formal languages have been extended into a quantitative setting. Weighted context-free languages of finite words trace back already to Chomsky and Schützenberger. Their work has been extended to infinite words by Ésik and Kuich. As in the theory of formal grammars, these weighted ω-context-free languages, or ω-algebraic series, can be represented as solutions of mixed ω-algebraic systems of equations and by weighted ω-pushdown automata.

In *our second main result*, we show that (mixed) ω-algebraic systems can be transformed into *Greibach normal form*.

We then investigate simple pushdown automata in the weighted setting. Here, we give *our third main result*. We prove that weighted simple pushdown automata of finite words recognize all weighted context-free languages, i.e., generate all algebraic series. Then, we show that weighted simple ω-pushdown automata generate all ω-algebraic series. This latter result uses the former result together with the Greibach normal form that we developed for ω-algebraic systems.

As a *fourth main result*, we prove that for weighted simple ω-pushdown automata, Büchi-acceptance and Muller-acceptance are expressively equivalent.

In our *fifth main result*, we derive a Nivat-like theorem for weighted simple ω-pushdown automata. This theorem states that the behaviors of our automata are precisely the projections of very simple ω-series restricted to ω-context-free languages.

The last result, *our sixth main result*, is a weighted logic with the same expressive power as weighted simple ω-pushdown automata. To prove the equivalence, we use a similar result for weighted nested ω-word automata and apply our present result of expressive equivalence of Muller and Büchi acceptance.

Acknowledgment

First, I want to thank Manfred Droste for supervising this thesis. He helped and guided me in many ways. I was happy to learn the fine art of research.

My second supervisor, Werner Kuich, introduced me to a very special kind of thinking. Thank you for your patience.

Dear reviewers, thank you for reviewing this thesis.

I want to thank my colleagues for a warm working climate, help and for many hours of company at the university canteen. I also want to thank the other QuantLA members, especially also the second generation, who welcomed me and who led me participate in many social events. Special thanks also to Karin.

I am indebted to the DFG (Deutsche Forschungsgemeinschaft) for financial support through the research training group QuantLA.

I am very grateful for having a supportive partner. Sarah, you are very important to me and you greatly shape my life. Thank you for everything!

My family helped me to get here and I know of all the benefits of a happy childhood.

Table of Contents

List of Terms

CHAPTER 1

Introduction

The aim of formal language theory is to understand and explain languages. This thesis tries to give new insights into a class that has not been investigated until recently. Only in 2004, Ésik and Kuich introduced weighted context-free languages of infinite words, i.e., ω-words (see Ésik and Kuich, 2007a, for a survey). They investigated weighted ω-context-free grammars called ω-algebraic systems. Droste, Ésik and Kuich (2017) and Droste and Kuich (2017) continued this research by investigating weighted ω-pushdown automata.

In this thesis, we give a logical characterization of weighted ω-context-free languages and therefore complete the trinity of formalisms for this language class: grammar, automaton and logic. The intermediate steps to the weighted logic may be of independent interest: We give a logical characterization of unweighted ω-context-free languages first. Then we establish a new normal form for pushdown automata: simple pushdown automata. Simple pushdown automata do not use ϵ-transitions and can change the stack only by at most one symbol. We compare and relate simple pushdown automata to context-free languages and specifically to the Greibach normal form of context-free grammars. This relation is investigated for infinite words in the unweighted case and for finite and infinite words in the weighted case. For weighted ω-algebraic systems, we first show the existence of the Greibach normal form.

Figure 1.1 gives an overview of the contributions in this thesis. It shows different extensions and generalizations of regular languages on its three axes: infinite words on the x-axis, context-free languages on the y-axis and weighted languages on the z-axis. Topics colored in burgundy are new in this thesis. As summarized in the figure, this thesis extends major contributions in the field (in green). We will give more details about related work in the following subsections.

1.1 Context-Free Languages

This thesis focuses on context-free languages. Chomsky (1959) classified the context-free languages as class 2 in the so-called "Chomsky hierarchy" (see Figure 1.2). Context-free languages are the basis for programming languages (e.g., see compiler construction, Aho, Sethi and Ullman, 1986; Waite and Goos, 1984; Wirth, 1996). Furthermore, many data structures are context-free because context-free languages can match opening and closing brackets and therefore describe tree-like structures. For example, arithmetic expressions and the Extensible Markup Language (XML) can be described by context-free grammars. Pushdown automata and context-free grammars are the most prominent

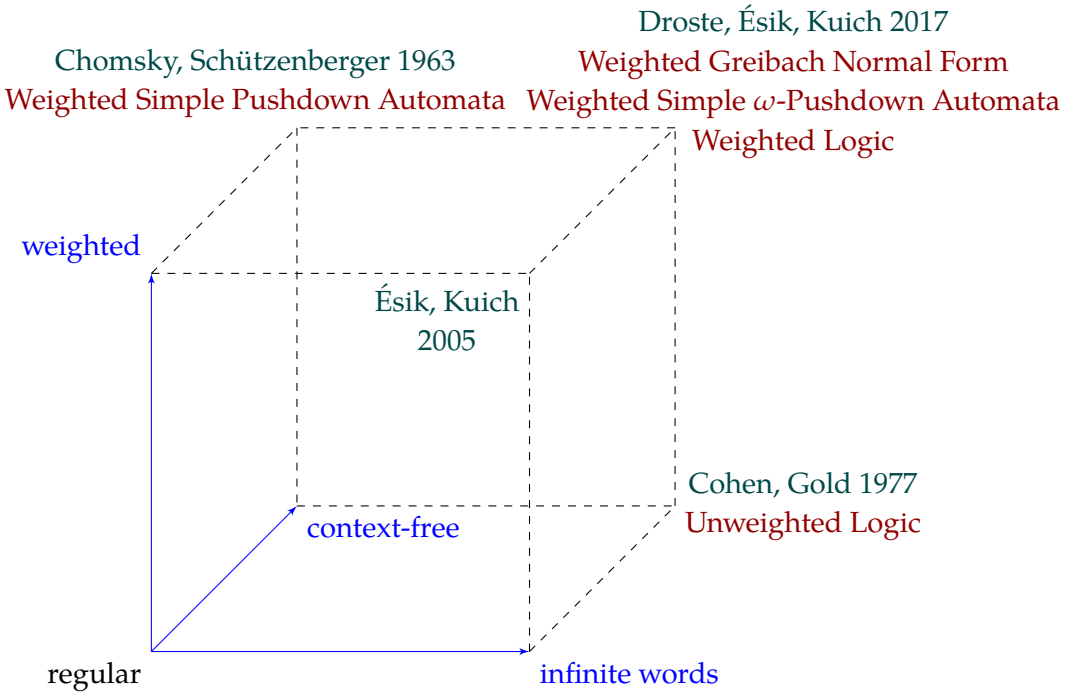


Figure 1.1: Overview: Extensions and generalizations of regular languages on three axes: languages of infinite words, context-free languages and weighted languages.
In green: Major contributions in the intersection of multiple extensions.
In burgundy: Contributions of this thesis.

examples of formalisms to describe exactly the class of context-free languages.

A major application of the theory of regular formal languages is *model checking* of software and hardware systems. Model checking plays an important role in software verification to review programs and to seek for mistakes (see McMillan, 1993; Baier and Katoen, 2008; Clarke et al., 2016, for background). The input of the model checking problem is the model, i.e., an abstraction of the software (or hardware) system, and a specification. The model checking problem then asks whether the system satisfies the specification. Both, the model and the specification must be effectively given. In general, the model will be given in the form of a Büchi automaton and the specification as a logical formula.

A seminal contribution to the fundamentals of model checking is the work of Büchi (1960), Elgot (1961) and Trakhtenbrot (1961). Their famous *BET-Theorem* provides the connection between finite automata and monadic second-order logic for finite words. It was extended to various other structures, like infinite words (Büchi, 1966, see also next section), finite trees (Thatcher and Wright, 1968), finite pictures (Giammarresi et al., 1996), finite and infinite nested words (Alur and Madhusudan, 2009), and context-free languages (Lautemann, Schwentick and Thérien, 1994).

Most common model checking algorithms work on regular models and regular spec-

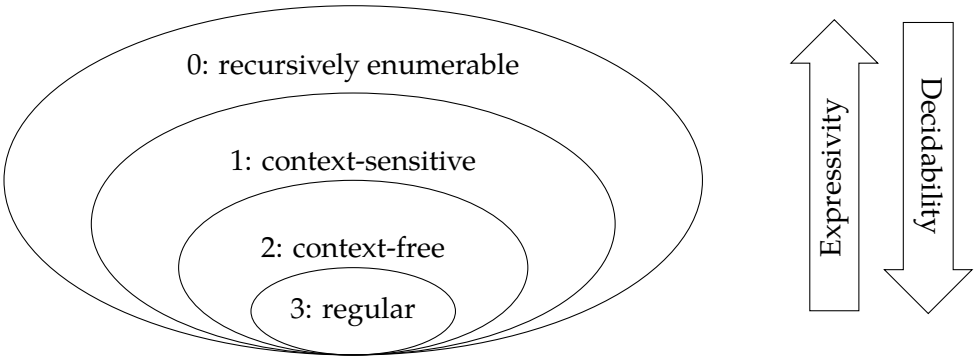


Figure 1.2: Chomsky hierarchy

ifications. But this is not always sufficient. Functional and procedural programming use function calls and even recursion; both cannot be described by regular languages, but they could be described by context-free languages. On the other hand, the use of context-free languages for both the model and its specification raises another problem: context-free languages are not closed under intersection. A compromise is to use either a context-free model or a context-free specification and restrict the other input to regular languages. For example, Esparza et al. (2000) describe how to check for reachability in context-free systems. Another solution is the restriction to subclasses of context-free languages. Examples of this are nested word languages or visibly pushdown languages (Alur and Madhusudan, 2004, 2009) and CaRet (Alur, Etessami and Madhusudan, 2004). An important contribution to model checking of non-regular properties was given by the BET-Theorem for context-free languages by Lautemann, Schwentick and Thérien (1994). They presented a monadic second-order logic with one additional existential dyadic first-order variable that describes exactly the context-free languages.

1.2 Infinite Words

In this thesis, we investigate mainly infinite words as input. Infinite strings mostly occur in logics and model checking. In fact, temporal logics, such as LTL or CTL, argue over time which is assumed to be infinite into the future (cf. Emerson, 1990, for a survey) but also other program logics, e.g., dynamic logics, consider infinite traces of programs (cf. Kozen and Tiuryn, 1990, for a survey). Similarly, model checking can be used for checking interactive and background processes. As opposed to batch processing (where programs calculate a result and then halt), many modern processes either wait for the user or for other programs to interact with them. For instance, web services and the Internet of things are two fields in which most processes only halt exceptionally. In these cases, it makes sense to abstract from physical running time and assume the program does not stop. Consequently, we have infinite runs of programs and are dealing with their infinite traces. Sets of such traces are then called languages of infinite words or ω-languages.

Automata of infinite words have been investigated to prove decidability of monadic second-order logic with one successor (Büchi, 1966; McNaughton, 1966) and with two successors (Rabin, 1969). Cohen and Gold (1977) extended ω-automata to the next layer of the Chomsky hierarchy, ω-context-free languages. The *ω-pushdown automata* behave similarly to their counterparts on finite words, but they are extended by a Büchi or Muller acceptance condition. Büchi acceptance means that at least one state of a designated set of states must occur infinitely often along an infinite run of the automaton. Equivalently, ω-context-free languages can be described by ω-context-free grammars. Cohen and Gold (1977) also developed fundamental results like closure under Kleene-like ω-rational operations for instance. The *first main contribution of this thesis* is a logic with the same expressive power as ω-context-free languages (see Chapter 2). This logic extends the work of Lautemann, Schwentick and Thérien (1994) to infinite words.

1.2.1 Simple Pushdown Automata

For the equivalence proof between logic and ω-context-free languages, we use a special kind of ω-pushdown automaton that we call *simple ω-pushdown automata*. Simple pushdown automata are realtime pushdown automata, i.e., without ϵ-transitions. Additionally, the stack access is restricted to the top symbol, i.e., the automaton can either pop that symbol, ignore the stack or push a new symbol. Read access to the stack is not needed except when popping the top symbol. Finally, simple pushdown automata start with an empty stack without the bottom of stack symbol.

For finite words, simple pushdown automata have already occurred hidden in a proof by Blass and Gurevich (2006); they used simple automata (without calling them in this way) to prove that general projections of regular nested word languages are context-free. Apart from that, simple pushdown automata with clocks were utilized by Droste and Perevoshchikov (2015a). Droste and Perevoshchikov do not discuss any comparisons of their automaton model with more general pushdown automata. They do, however, show a BET-Theorem for these automata. As they define a logic that has similarities to the logic of Lautemann, Schwentick and Thérien (1994) and they prove that this logic is equivalent to their automaton model, one could conclude that their automaton model could all context-free languages. We present here a more direct approach to prove this.

Otherwise, simple pushdown automata seem to have been overlooked although they provide a very natural model and work already for finite words. Note that realtime as well as stack-restricted pushdown automata have been considered in the literature but their combination is non-trivial. We extend the result of Blass and Gurevich (2006) and prove that simple ω-pushdown automata recognize all ω-context-free languages. For the proof, we use ω-context-free grammars in Greibach normal form and convert them directly into simple ω-pushdown automata. Alternatively, we show how we can restrict the stack access if the ω-pushdown automaton is already realtime.

1.3 Weighted Languages

Traditional formal language theory investigates qualitative questions like: "Is the word in the language or not?" In many cases, we also want to determine quantitative properties. Quantitative languages or *weighted languages* relate words to resources such as probabilities, costs, gains, energy consumption, consumption of other resources, counts, transductions and even averages, optimal values or discountings of the above. Depending on the question asked, one might consider different weight structures. To abstract from the specific question, weight structures with similar properties are clustered and investigated together. The most prominent investigated weight structures are semirings. Semiring weighted automata multiply weights of transitions along the run and add the weights of all runs to gain the weight of an input word. There exist multiple variants like complete semirings, bimonoids or valuation monoids. Complete semirings are semirings that allow infinite sums. Automata over valuation monoids likewise sum up the weights of all possible runs but the multiplication is different; the weights along the run are not consecutively multiplied but given as a sequence to the valuation function. Valuation monoids include complete semirings but also allow discounted and average behavior. They were first introduced by Droste and Meinecke (2012) but their idea is based on Chatterjee, Doyen and Henzinger (2008). In this thesis, we consider both, complete semirings and valuation monoids.

Additionally to possible weight structures, many language classes were generalized to the weighted setting (e.g., regular, context-free, star-free languages). Even various input structures were considered (like e.g., words, trees, pictures, nested words, infinite words, etc.). See the books by Salomaa and Soittola (1978), Kuich and Salomaa (1986), Ésik and Kuich (2007a) and Droste, Kuich and Vogler (2009) for an overview.

Weighted context-free languages already date back to the work of Chomsky and Schützenberger (1963). The theory of weighted pushdown automata has been extensively studied (see Kuich, 1997, for a survey). On the other hand, weighted languages of infinite words were first considered by Ésik and Kuich (2005a,b)[1]. They investigated weighted ω-regular languages first and then, Ésik and Kuich (2004) extended their work to weighted ω-context-free languages. Current developments in the area are given by e.g., Droste and Vogler (2014) who established a Chomsky-Schützenberger type result for weighted pushdown automata. Droste and Perevoshchikov (2015b) developed a weighted logic with the same expressive power as weighted pushdown automata. For infinite words, weighted ω-pushdown automata were considered only recently by Droste, Ésik and Kuich (2017) and Droste and Kuich (2017).

Important applications for weighted ω-languages are for instance probabilistic model checking but also model checking with other quantitative questions. One might ask whether some program always runs with less energy consumption (cf. e.g. Bouyer et al., 2008) or whether it has an average cost below some threshold. In Chapter 5, we give an

[1] This was beforehand published as a technical report of the Technical University Vienna in 2003.

example on how a basic web server and its average response time can be modeled by a weighted ω-pushdown automaton. For model checking, it makes sense to consider not only automata and grammars but more prominently logic. Therefore, since the 1960s, the BET-Theorem has been extended and generalized to multiple input structures (like e.g., trees, pictures, nested words, infinite words, etc., see above, Section 1.1). Droste and Gastin (2007) generalized the BET-Theorem to quantitative languages. They opened a very new branch of research. The weighted BET-Theorem was extended to algebraic power series by Mathissen (2008), to weighted languages of infinite words (Droste and Rahonis, 2006; Droste and Gastin, 2009; Droste and Meinecke, 2012), to weighted higher-order pushdown automata by Vogler, Droste and Herrmann (2016), and to weighted timed pushdown automata by Droste and Perevoshchikov (2015b). See Droste, Kuich and Vogler (2009) for an overview.

The final goal of the thesis is the extension of our first main result to the weighted setting, i.e., to find a weighted logic for weighted ω-context-free languages. To reach this goal, we use a similar way as discussed before for the unweighted case. Only, for the weighted case, much more groundwork has to be laid. In fact, some open problems remain (see Chapter 6, page 122, for details). As the *second main result*, we develop a new normal form for weighted ω-context-free grammars, i.e., for ω-algebraic systems, namely the Greibach normal form (see Chapter 3). Besides the Chomsky normal form, the normal form invented by Greibach (1965) is the most well-known normal form for context-free languages. The Greibach normal form is usually used for the construction of realtime, i.e., ϵ-free pushdown automata (cf. e.g. Kuich and Salomaa, 1986). The proof that ω-algebraic systems can be transformed into Greibach normal form is given in Chapter 3.

As our *third main result*, we define weighted simple ω-pushdown automata and prove that they recognize all weighted ω-context-free languages (see Chapter 4). As a subresult, we also solve this case for languages of finite words, i.e., weighted simple pushdown automata recognize all weighted context-free languages. This generalizes the corresponding result of Chapter 2 from the unweighted to the weighted case. The generalization shows that the basic model is very natural. Both, the result on the Greibach normal form and the result on weighted simple ω-pushdown automata use complete semirings as weight structure because algebraic systems heavily rely on distributivity. For the Greibach normal form, we have to assume that the semiring is additionally commutative.

For the rest of this thesis, we consider ω-valuation monoids. This is a valuation monoid that processes infinite sequences that naturally occur in the context of infinite words. As ω-valuation monoids are a generalization of complete semirings, one can easily combine the following with the preceding results when restricting the weight structure to complete semirings. Our *fourth main result* is the expressive equivalence of Büchi and Muller acceptance for weighted simple ω-pushdown automata over ω-valuation monoids.

As a *fifth main result*, we show a Nivat-like theorem for weighted simple ω-pushdown automata over ω-valuation monoids. This theorem states that the weighted languages recognized by weighted simple ω-pushdown automata are induced by an unweighted ω-context-free language and a very simple weighted finite automaton with only one state; the two components can be intersected and a projection of this intersection gives us the original language. Such a theorem was first given by Nivat (1968) for rational transductions. Nivat hereby showed how to combine a rational language together with homomorphisms and inverse homomorphisms into a rational transducer. Droste and Kuske (to appear) extended Nivat's theorem to weighted automata of finite words over semirings.

The *sixth main contribution of this thesis* is a BET-Theorem for weighted simple ω-pushdown automata (see Chapter 5). This generalizes the BET-Theorem for unweighted ω-context-free languages that is also newly given in this thesis (Chapter 2), and this extends the BET-Theorem for context-free languages of Lautemann, Schwentick and Thérien (1994) and the BET-Theorem for weighted regular ω-languages (Droste and Rahonis, 2006; Ésik and Kuich, 2004) and the BET-Theorem for weighted nested ω-words (Droste and Dück, 2017). We will be using ω-valuation monoids for our logical characterization.

1.4 Structure of This Thesis

After the introduction, we show a logical characterization of unweighted context-free languages of infinite words *in Chapter 2*. An overview over this chapter is given in Figure 1.3; in the following, the connections depicted in the figure are discussed in more detail. The chapter introduces ω-context-free grammars. It gives a detailed proof of the existence of the Greibach normal form for ω-context-free grammars (the proof idea was already given in Cohen and Gold, 1977). Then we introduce simple ω-pushdown automata and prove that they recognize all ω-context-free languages. The chapter also gives an insight into simple pushdown automata of finite words, and we explain how to convert ω-pushdown automata that are already realtime, i.e., that have no ϵ-transitions, into simple ω-pushdown automata. After the excursion, we present the *matching ω-logic*. To prove its expressive equivalence with ω-context-free languages, we use a corresponding result on nested ω-words. Therefore, we recall known results from Alur and Madhusudan (2004) on visibly pushdown ω-languages. Here, we present a projection from nested ω-word languages to ω-context-free languages that is new for infinite words but is similar to an idea of Blass and Gurevich (2006) for finite words. Then we finally prove the BET-theorem for ω-context-free languages.

The subsequent chapters are devoted to a similar result for weighted ω-context-free languages. Figure 1.4 gives an overview. Results on the Greibach normal form and results on simple pushdown automata have been moved to their own chapter. Open problems are indicated by dashed blue arrows; see Chapter 6, page 122, for details.

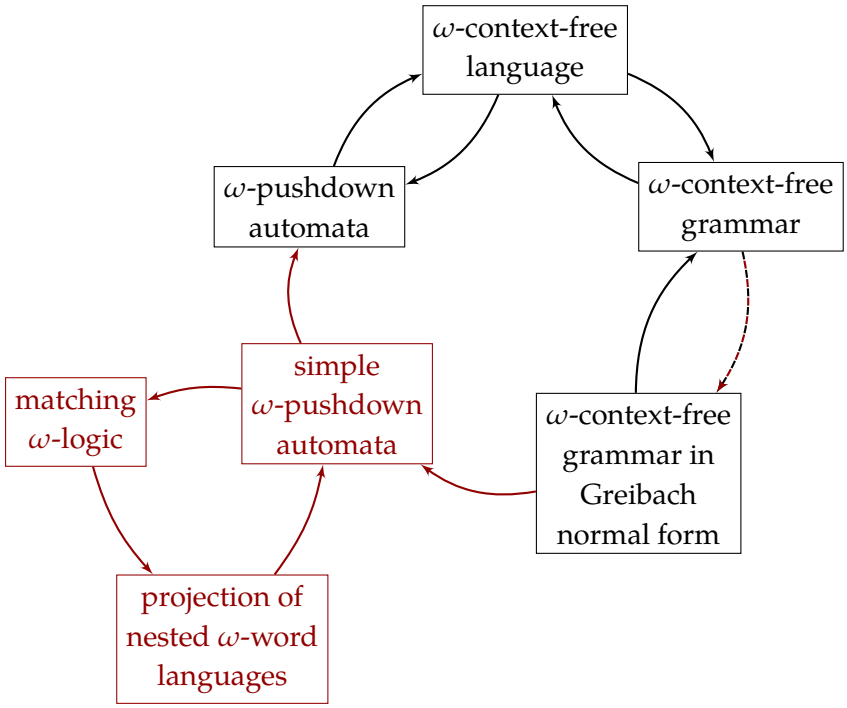


Figure 1.3: Overview Chapter 2: Logics for unweighted ω-context-free languages. In burgundy: New contributions of this thesis.

Chapter 3 proves that ω-algebraic systems can be transformed into Greibach normal form; ω-algebraic systems are basically systems of equations that generalize ω-context-free grammars. We will be using continuous star-omega semirings and sometimes even commutative semirings as a weight structure in this chapter. The chapter gives many preliminaries on complete and continuous semirings as well as semiring-semimodule pairs; the preliminaries will also be used in the subsequent chapter. First, we give a simple but important theorem on ω-powers of matrices considering Büchi-acceptance. Then, we introduce ω-algebraic systems together with mixed ω-algebraic systems. We use an existing result on the ω-Kleene closure for weighted context-free languages to build a mixed ω-algebraic system in Greibach normal form for every ω-algebraic series. Furthermore, as the main result of this chapter, we show how to extend this result from mixed ω-algebraic systems to ω-algebraic systems.

Weighted simple pushdown automata are discussed in great depth in *Chapter 4*. The chapter is divided in two parts, the first one focuses on finite words, the second on infinite words. As these automata specialize reset pushdown automata, they will be called simple reset pushdown automata. We introduce reset pushdown matrices and in particular, simple reset pushdown matrices. Many important basic results on these matrices follow. Then, we define simple reset pushdown automata and present a construction on how to construct these simple automata from algebraic systems in Greibach normal form. The Greibach normal form for finite words can be gained

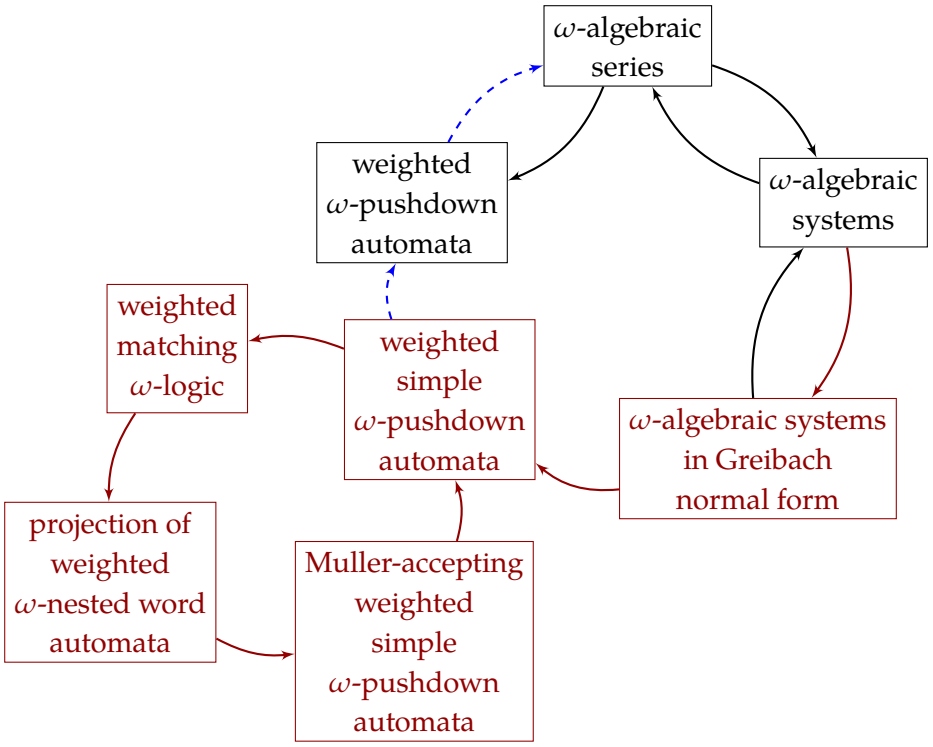


Figure 1.4: Overview Chapters 3 to 5: Logics for weighted ω-context-free languages.
In burgundy: New contributions of this thesis.
In dashed blue: The direction from weighted simple ω-pushdown automata to ω-algebraic series is currently open. A weaker result is provided by Droste, Ésik and Kuich (2017) (see Chapter 6, page 122, for details).

from an already existing result of Kuich and Salomaa (1986). We then prove that the construction is correct and thus, every algebraic series is the behavior of a simple reset pushdown automaton. We also present a small excursion on normal forms for algebraic systems.

In the second part of Chapter 4, we continue with simple reset pushdown automata of infinite words, the simple ω-reset pushdown automata. We first prove some results on infinite applications of simple reset pushdown matrices. Then we introduce simple ω-reset pushdown automata. Afterwards, we present an extension of the construction that we used for finite words. This construction uses ω-algebraic systems in Greibach normal form that can be gained from Chapter 3 and the construction is adapted to the structure of mixed ω-algebraic systems. We show that we can apply the results from the first part of the chapter. Then we investigate the canonical solutions of ω-algebraic systems and prove that they are equal to the behaviors of our constructed simple ω-reset pushdown automata.

Chapter 5 continues the work of Chapters 3 and 4 in the sense that it uses weighted simple ω-pushdown automata and shows a logical characterization for these automata.

However, we generalize the weight structure as we consider ω-valuation monoids in this chapter. Our weight structure, the ω-valuation monoids, are described in the chapter together with some of their properties. We then introduce weighted simple ω-pushdown automata over ω-valuation monoids. We show how they relate to the simple ω-reset pushdown automata in the preceding chapter. Our first main result of this chapter is the expressive equivalence of Büchi and Muller acceptance for weighted simple ω-pushdown automata. Then, we prove several closure properties for these automata and a Nivat-like theorem. We present the weighted matching ω-logic. Furthermore, we recall results of Droste and Dück (2017) on weighted nested ω-word languages. The projection from weighted nested ω-word languages to weighted ω-context-free languages uses our present result of expressive equivalence of Muller and Büchi acceptance. Finally, we prove the expressive equivalence between the weighted matching ω-logic and weighted simple ω-pushdown automata for various restrictions on our logic. In this chapter, we include two examples on how we can model average response times of a basic web server by a weighted simple ω-pushdown automaton and by our weighted logic, respectively.

The last chapter, *Chapter 6* summarizes our results and indicates possible future work.

CHAPTER 2

Unweighted Context-Free ω-Languages

This chapter shows a BET-Theorem for unweighted context-free languages of infinite words. Büchi (1960), Elgot (1961) and Trakhtenbrot (1961) proved that regular languages are exactly those languages definable by monadic-second-order logic. Here we will generalize their result to context-free languages (like Lautemann, Schwentick and Thérien, 1994) and extend it to infinite words (like Büchi, 1966).

For the following outline of the proof, please revert to Figure 1.3. Recall that in formal language theory, grammars in Greibach normal forms are of basic importance for context-free languages of finite words. Here, we will first use a Kleene-type result of Cohen and Gold (1977) to show that each ω-context-free language has a Büchi-accepting grammar in quadratic Greibach normal form. This enables us to show, as our first main new result, that each ω-context-free language can be accepted by a simple ω-pushdown automaton. A similar construction for context-free languages of finite words occurred within an argument of Blass and Gurevich (2006). Simple ω-pushdown automata are ω-pushdown automata without ϵ-transitions and with very specific access to the stack; the simple ω-pushdown automaton can either push one symbol, pop one symbol or ignore the stack altogether. As a technical difference to usual pushdown automata, simple ω-pushdown automata start with an empty stack and cannot read the top of the stack — except when popping it.

We show that the languages of simple pushdown automata are, in a natural way, projections of visibly pushdown languages investigated by Alur and Madhusudan (2004). Now we can use their expressive equivalence result for visibly pushdown languages and monadic second-order logic to derive our second main result, the logical description of ω-context-free languages. Since our proof is constructive and the emptiness problem for ω-pushdown automata is decidable (cf. Lei et al., 2017), we can also decide the emptiness of simple ω-pushdown automata and therefore the satisfiability for our matching ω-logic.

The chapter is structured so that we first deal with ω-context-free grammars in Greibach normal form in Section 2.1. Then we use those grammars in Section 2.2 for our simple ω-pushdown automata. A small excursion (Section 2.3) investigates simple pushdown automata also for finite words and constructs them if ϵ-transitions are already non-existent. Our language-theoretic results for simple ω-pushdown automata can be read independently of the rest of the chapter. Afterwards, in Section 2.4, we define our second-order logic. We recall some results of visibly pushdown ω-languages in Section 2.5 and then use them to prove our BET-Theorem in Section 2.6.

This chapter is based on Droste, Dziadek and Kuich (2020a).

2.1 Greibach Normal Form

Let us first make two conventions for the entire thesis. An *alphabet* denotes a finite set of symbols. The set $\mathbb{N} = \{0, 1, 2, \ldots\}$ of *natural numbers* are the non-negative integers.

Now, let us recall some background. The concept of ω-context-free languages has been defined by Cohen and Gold (1977) (for an overview cf. Thomas, 1990). They are defined to be the languages generated by ω-context-free grammars with Muller-acceptance condition (cf. Muller, 1963). It is shown that these languages coincide with the class of languages recognized by general ω-pushdown automata, both for Büchi- and Muller-acceptance condition.

The ω-context-free grammars are similar to context-free grammars for finite words but we consider only infinite leftmost derivations and define both Büchi- and Muller-acceptance conditions.

Definition 2.1 (Cohen and Gold, 1977)**.** An ω-context-free grammar is a tuple $G = (N, \Sigma, P, S, F)$ where (N, Σ, P, S) is an ordinary context-free grammar for finite words and F defines the acceptance condition: If G is Muller-accepting, we have $F \subseteq 2^N$. If G is Büchi-accepting, we have $F \subseteq N$.

Let $\delta\colon S \to \ldots$ be an infinite derivation of G. We write $\delta\colon S \to_G^\omega w$ if $w \in \Sigma^\omega$ is the infinite word of terminals occurring in the production rules of δ. For $i \geq 0$, let $\delta_N(i) = A_i$ be the non-terminal which is the left-hand side of the rule applied in step i of derivation δ. We define $\mathrm{Inf}(\delta) = \{A \mid A = A_i \text{ for infinitely many } i \geq 0\}$.

For a Muller-accepting ω-context-free grammar G, the language *generated* by G is defined as

$$\mathcal{L}(G) = \{w \in \Sigma^\omega \mid \exists \text{ leftmost derivation } \delta\colon S \to_G^\omega w \text{ with } \mathrm{Inf}(\delta) \in F\}.$$

For Büchi-accepting ω-context-free grammars,

$$\mathcal{L}(G) = \{w \in \Sigma^\omega \mid \exists \text{ leftmost derivation } \delta\colon S \to_G^\omega w \text{ with } \mathrm{Inf}(\delta) \cap F \neq \emptyset\}.$$

A language $L \subseteq \Sigma^\omega$ is said to be an ω-context-free language if $L = \mathcal{L}(G)$ for a Muller-accepting ω-context-free grammar G. ▼

Clearly, every Büchi-accepting ω-context-free grammar can be translated in a Muller-accepting one. The inverse is not so easily seen.

Lemma 2.2 (Ésik and Iván, 2011)**.** *Every ω-context-free language is generated by a Büchi-accepting ω-context-free grammar.*

This has been shown by Ésik and Iván (2011) by using automata of countable words, i.e., also words with multiple ω-operators are allowed. But it can be shown directly by using a standard idea used already to translate Muller-automata into Büchi-automata (cf. Cohen and Gold, 1977, Theorem 4.1.4). In the sequel, a stronger result will be needed and is provided by Lemma 2.4 below.

For finite words, the following definition is standard, cf. Autebert, Berstel and Boasson (1997) for an overview.

Definition 2.3. An ω-context-free grammar $G = (N, \Sigma, P, S, F)$ is in *Greibach normal form* if $P \subseteq N \times \Sigma N^*$. More specifically, G is in *quadratic Greibach normal form* if

$$P \subseteq N \times (\Sigma \cup \Sigma N \cup \Sigma N N)\,. \qquad \blacktriangledown$$

Note that this is the only chapter where we differentiate between the quadratic and the more general Greibach normal form. In the subsequent chapters, we will only consider the quadratic version and therefore denote it simply by Greibach normal form.

Lemma 2.4. *Let L be an ω-context-free language. There exists a Büchi-accepting ω-context-free grammar G in quadratic Greibach normal form with $\mathcal{L}(G) = L$.*

The idea of the proof is similar to the idea given by Cohen and Gold (1977, Theorem 4.2.2), which shows that for Muller-accepting ω-context-free grammars one can remove rules of the type $A \to \epsilon$. Cohen and Gold (1977) claim in Theorem 4.2.4 that the same idea can be used to prove that for every ω-context-free language there exists a Muller-accepting ω-context-free grammar in Greibach normal form. We show it here for Büchi-acceptance and for the stricter quadratic Greibach normal form.

Proof. By Theorem 4.1.8 of Cohen and Gold (1977), L can be expressed as the Kleene-closure of context-free languages of finite words, i.e., for some $l \in \mathbb{N}$, there exist context-free grammars G_i, G_i' $(1 \leq i \leq l)$ such that

$$L = \bigcup_{i=1}^{l} \mathcal{L}(G_i)\mathcal{L}(G_i')^{\omega}\,.$$

For $1 \leq i \leq l$, let $G_i = (N_i, \Sigma, P_i, S_i)$ and $G_i' = (N_i', \Sigma, P_i', S_i')$ and we assume all N_i and N_i' to be pairwise distinct. As the G_i and G_i' are context-free grammars for finite words, we can assume they are in quadratic Greibach normal form (cf. e.g. Autebert, Berstel and Boasson, 1997, p. 16).

We construct the Büchi-accepting ω-context-free grammar $G = (N, \Sigma, P, S, F)$ where $N = \{S\} \cup \bigcup_{i=1}^{l}(N_i \cup N_i' \cup \{\bar{S}_i\})$ and $F = \{\bar{S}_i \mid 1 \leq i \leq l\}$ as follows; here the symbols S and $\bar{S}_i$ are new symbols and are assumed to be neither in N_i nor in N_i'. We define as an intermediate step

$$P_{\text{tmp}} = \{S \to S_i\bar{S}_i,\ \bar{S}_i \to S_i'\bar{S}_i \mid 1 \leq i \leq l\} \cup \bigcup_{i=1}^{l} \left(P_i \cup P_i'\right).$$

The grammar G with P_{tmp} as set of production rules accepts the language L but is not yet in Greibach normal form. To achieve this, we first substitute S_i and S_i' with all possible right-hand sides of their productions to obtain G in Greibach normal form:

$$P = \{S \to \alpha\bar{S}_i,\ \bar{S}_i \to \alpha'\bar{S}_i \mid S_i \to \alpha \in P_i,\ S_i' \to \alpha' \in P_i', 1 \leq i \leq l\} \cup \bigcup_{i=1}^{l} \left(P_i \cup P_i'\right)$$

Unfortunately, α and α' can already contain two non-terminals and therefore, G can contain up to three non-terminals on the right-hand sides of its productions. Thus, G is not yet in quadratic Greibach normal form.

The Büchi-acceptance condition and the definition of F ensure that for every word accepted by G, one of the G'_i is applied infinitely many times. Hence

$$\mathcal{L}(G) = \bigcup_{i=1}^{l} \mathcal{L}(G_i)\mathcal{L}(G'_i)^\omega = L\,.$$

Now, we apply a standard algorithm (see e.g., Harrison, 1978, Theorem 4.7.1) to convert P into quadratic Greibach normal form $\bar{G} = (\bar{N}, \Sigma, \bar{P}, \bar{S}, F)$ with $\bar{N} = N \cup N \times N$ and

$$\begin{aligned}
\bar{P} = \bigcup_{i=1}^{l} (P_i \cup P'_i) & \\
\cup \{\bar{S} \to \alpha\bar{S}_i & \mid S_i \to \alpha \in P_i, |\alpha| \leq 2, 1 \leq i \leq l\} \\
\cup \{\bar{S} \to aB(C, \bar{S}_i) & \mid S_i \to aBC \in P_i, 1 \leq i \leq l\} \\
\cup \{\bar{S}_i \to \alpha\bar{S}_i & \mid S'_i \to \alpha \in P'_i, |\alpha| \leq 2, 1 \leq i \leq l\} \\
\cup \{\bar{S}_i \to aB(C, \bar{S}_i) & \mid S'_i \to aBC \in P'_i, 1 \leq i \leq l\} \\
\cup \{(A, \bar{S}_i) \to \alpha\bar{S}_i & \mid A \to \alpha \in P_i \cup P'_i, |\alpha| \leq 2, 1 \leq i \leq l\} \\
\cup \{(A, \bar{S}_i) \to aB(C, \bar{S}_i) & \mid A \to aBC \in P_i \cup P'_i, 1 \leq i \leq l\}\,.
\end{aligned}$$

The new non-terminals are pairs (A, B) with production rules that apply a production rule of non-terminal A and add the non-terminal B to its right-hand side.

Technically, because the G_i and G'_i are grammars for finite words, they could derive the empty word ϵ. In this special case, whenever $|\alpha| = 0$, we substitute in the above construction $\alpha\bar{S}_i$ by all right-hand sides of productions for $\bar{S}_i$ and we simply omit rules $\bar{S}_i \to \alpha\bar{S}_i$.

This shortens the production rules to at most two non-terminals on the right-hand side. As only an occurrence of the non-terminal $\bar{S}_i$ in $\bar{G}$ implies an occurrence of $\bar{S}_i$ in a derivation of G, the set of Büchi states F is not changed. It follows that

$$\mathcal{L}(\bar{G}) = \mathcal{L}(G) = L$$

and $\bar{G}$ is in quadratic Greibach normal form. □

Note that the above construction for $\bar{P}$ needs one case less than the original construction by Harrison (1978, Theorem 4.7.1). In the original construction, there is a special case, in which for (A, B) there is already a production rule for A with three non-terminals on the right hand side. In $\bar{P}$, such production rules only occur for $\bar{S}$ and for $\bar{S}_i$. But neither $\bar{S}$ nor any of the $\bar{S}_i$ occur in the first position of any pair (A, B).

Example 2.5. Let $\Sigma = \{a, b, c\}$, $L_1 = c^+$ and $L_1' = \{w \in \{a, b\}^+ \mid |w|_a = |w|_b\}$. The corresponding grammars are $G_1 = (\{S_1\}, \Sigma, P_1, S_1)$ where $P_1 = \{S_1 \to c \mid cS_1\}$ and $G_1' = (N_1', \Sigma, P_1', S_1')$ where $N_1' = \{S_1', M, N, A, B\}$ and P_1' contains the rules

$$\begin{aligned}
S_1' &\to aB \mid bA \mid aS_1'B \mid bS_1'A \mid aBS_1' \mid bAS_1' \mid aS_1'M \mid bS_1'N \\
M &\to bS_1' \\
N &\to aS_1' \\
A &\to a \\
B &\to b\,.
\end{aligned}$$

The corresponding grammar for $L = L_1 L_1'^{\omega}$ is $\bar{G} = (\bar{N}, \Sigma, \bar{P}, \bar{S}, F)$ with $F = \{\bar{S}_1\}$ and $\bar{P}$ contains the rules of P_1, of P_1', and additionally

$$\begin{aligned}
\bar{S} &\to c\bar{S}_1 \mid cS_1\bar{S}_1 \\
\bar{S}_1 &\to aB\bar{S}_1 \mid bA\bar{S}_1 \mid aS_1'(B, \bar{S}_1) \mid bS_1'(A, \bar{S}_1) \mid aB(S_1', \bar{S}_1) \mid bA(S_1', \bar{S}_1) \mid \\
&\qquad aS_1'(M, \bar{S}_1) \mid bS_1'(N, \bar{S}_1) \\
(B, \bar{S}_1) &\to b\bar{S}_1 \\
(A, \bar{S}_1) &\to a\bar{S}_1 \\
(S_1', \bar{S}_1) &\to aB\bar{S}_1 \mid bA\bar{S}_1 \mid aS_1'(B, \bar{S}_1) \mid bS_1'(A, \bar{S}_1) \mid aB(S_1', \bar{S}_1) \mid bA(S_1', \bar{S}_1) \mid \\
&\qquad aS_1'(M, \bar{S}_1) \mid bS_1'(N, \bar{S}_1) \\
(M, \bar{S}_1) &\to bS_1'\bar{S}_1 \\
(N, \bar{S}_1) &\to aS_1'\bar{S}_1\,.
\end{aligned}$$

▽

2.2 Simple ω-Pushdown Automata

Common definitions of ω-pushdown automata (cf. e.g., Cohen and Gold, 1977) extend pushdown automata of finite words by a set of Muller- or Büchi-accepting final states. We do not directly work with this automaton definition because the equivalence proof for this automaton and the logic we will define in Section 2.4 is not easily possible.

Instead, we propose another automaton model, the simple ω-pushdown automaton. As Droste and Perevoshchikov (2015a), we restrict the access to the stack to only allow either to keep the stack unaltered, to push one symbol or to pop one symbol. This will later allow us to employ a simple translation from nested word automata to our automaton model. We believe that this automaton model is also of independent interest.

In this section, we prove that the simple ω-pushdown automata recognize all ω-context-free languages. This means that the restrictions in the automaton model do not change the expressiveness of ω-pushdown automata. Even though this property seems to be very basic, we could not find many results in the literature on this property.

For the case of finite words, Blass and Gurevich (2006) show how to translate nested-word automata into pushdown automata and only use three stack commands for their automata. As we borrowed the restrictions of our automaton model from nested-word

automata, we use a similar strategy in the following to prove the expressive equivalence also in the infinite case.

For an alphabet Γ, let $\mathcal{S}(\Gamma) = (\{\downarrow\} \times \Gamma) \cup \{\#\} \cup (\{\uparrow\} \times \Gamma)$ be the set of *stack commands*.

Definition 2.6. A *simple ω-pushdown automaton* (ωPDA) over the alphabet Σ denotes a tuple $M = (Q, \Gamma, T, I, F)$ where

- Q is a finite set of states,
- Γ is a finite stack alphabet,
- $T \subseteq Q \times \Sigma \times Q \times \mathcal{S}(\Gamma)$ is a set of transitions,
- $I \subseteq Q$ is a set of initial states,
- $F \subseteq Q$ is a set of (Büchi-accepting) final states. ▼

A *configuration* of an ωPDA $M = (Q, \Gamma, T, I, F)$ is a pair (q, γ), where $q \in Q$ and $\gamma \in \Gamma^*$. We define the transition relation between configurations as follows. Let $\gamma \in \Gamma^*$ and $t \in T$. If $t = (q, \sigma, q', (\downarrow, A))$, we let $(q, \gamma) \vdash_M^t (q', A\gamma)$. If $t = (q, \sigma, q', \#)$, we put $(q, \gamma) \vdash_M^t (q', \gamma)$. Finally, if $t = (q, \sigma, q', (\uparrow, A))$, we let $(q, A\gamma) \vdash_M^t (q', \gamma)$. These three types of transitions are called *push*, *internal* and *pop* transitions, respectively. Note that the stack here grows to the left.

We denote by $\text{state}(q, \sigma, q', s) = q$ the state and by $\text{label}(q, \sigma, q', s) = \sigma$ the *label* of a transition. Both will be extended to infinite sequence of transitions by letting $\text{state}((t_i)_{i\geq 0}) = (\text{state}(t_i))_{i\geq 0} \in Q^\omega$ for the infinite sequence of states and similarly $\text{label}((t_i)_{i\geq 0}) = (\text{label}(t_i))_{i\geq 0} \in \Sigma^\omega$ for the infinite word constructed from the labels of the transitions.

We call an infinite sequence of transitions $\rho = (t_i)_{i\geq 0}$ with $t_i \in T$ a *run* of $M = (Q, \Gamma, T, I, F)$ on $w = \text{label}(\rho)$ iff there exists an infinite sequence of configurations $(q_i, \gamma_i)_{i\geq 0}$ with $q_0 \in I$ and $\gamma_0 = \epsilon$ such that $(q_i, \gamma_i) \vdash_M^{t_i} (q_{i+1}, \gamma_{i+1})$ for each $i \geq 0$.

For the sequence of states $(q_i)_{i\geq 0}$, let $\text{Inf}((q_i)_{i\geq 0}) = \{q \mid q = q_i \text{ for infinitely many } i \geq 0\}$. The run ρ is called *successful* if $\text{Inf}(\text{state}(\rho)) \cap F \neq \emptyset$.

Definition 2.7. For an ωPDA $M = (Q, \Gamma, T, I, F)$ over Σ, the language *accepted* by M is denoted by $\mathcal{L}(M) = \{w \in \Sigma^\omega \mid \exists \text{ successful run of } M \text{ on } w\}$. A language $L \subseteq \Sigma^\omega$ is called *ωPDA-recognizable* if there exists an ωPDA M with $\mathcal{L}(M) = L$. ▼

For clarity, we abbreviate a run $\rho = (t_i)_{i\geq 0}$ with $(q_0, \gamma_0) \vdash_M^{t_0} (q_1, \gamma_1) \vdash_M^{t_1} \cdots$ where $\text{label}(t_i) = a_i$ by $\rho\colon (q_0, \gamma_0) \xrightarrow{a_0} (q_1, \gamma_1) \xrightarrow{a_1} \cdots$ such that the word becomes visible.

Example 2.8. We define an example automaton $\mathcal{A} = (Q, \{S, B\}, T, S, \{S\})$ over $\Sigma = \{a, b\}$ with $Q = \{S, M, B\}$ and the transitions T as depicted in Fig. 2.1. In state M, the automaton reads a and pushes B. For every B that is popped from the stack, the automaton reads b. When there are no more B on the stack, S is remaining on the stack and b brings the automaton to start from the beginning. As S is the only final state, we have $\mathcal{L}(\mathcal{A}) = \{a^n b^n \mid n \geq 1\}^\omega$. ▽

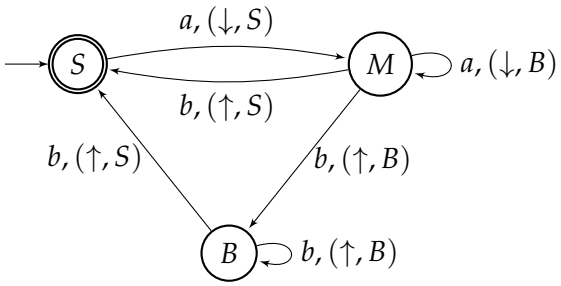


Figure 2.1: Example 2.8: Automaton

We call intermediate steps in a derivation $\delta\colon S \to \alpha_1 \to \alpha_2 \to \dots$ *sentential forms*. Thus, the i^{th} sentential form of δ is α_i. Similarly, the i^{th} configuration of a run $\rho\colon \gamma_0 \vdash \gamma_1 \vdash \dots$ is defined to be γ_i.

Clearly, every ωPDA-recognizable language is ω-context-free. We will now show the inverse which will be the first main result of this chapter.

Theorem 2.9. *Every ω-context-free language is ωPDA-recognizable.*

Proof. Let L be an ω-context-free language. By Lemma 2.4, L is generated by some Büchi-accepting ω-context-free grammar $G = (N, \Sigma, P, S, F)$ in quadratic Greibach normal form. We construct an ωPDA $M = (Q, \Gamma, T, I, F)$ with $Q = \Gamma = N$, $I = \{S\}$, and

$$T = \{(A, a, B, (\downarrow, C)) \mid A \to aBC \in P\} \cup \tag{2.1}$$
$$\{(A, a, B, \#) \mid A \to aB \in P\} \cup \tag{2.2}$$
$$\{(A, a, B, (\uparrow, B)) \mid A \to a \in P, B \in N\} \tag{2.3}$$

for $a \in \Sigma$ and $A, B, C \in N$.

Intuitively, the non-terminals in the grammar are simulated by states in the automaton. The second non-terminal on the right side of the productions is pushed to the stack to store it for later. Whenever a final production is processed (Equation (2.3)), it is checked which non-terminal is waiting on the stack to be processed. The Büchi-accepting final states of M are the same as in G. As the grammar only allows derivations where non-terminals in F occur infinitely often, the automaton will only allow runs where the same is true for states in F.

Claim: There exists a derivation $\delta\colon S \to \dots \to a_1 \dots a_i A_1 \dots A_j \to \dots$ of G if and only if there exists a run $\rho\colon (S, \epsilon) \xrightarrow{a_1} \dots \xrightarrow{a_i} (A_1, A_2 \dots A_j) \to \dots$ of M, with $i \geq 0$.

We prove the claim by an inductive construction of steps i in the derivation δ and in the run ρ:

Let $i = 0$. Then the derivation $\delta\colon S$ is still in start state. As $I = \{S\}$, the start of the corresponding run is $\rho\colon (S, \epsilon)$. The same argument holds for the other direction.

Let $i > 0$. We distinguish three cases:

1. Let the i^{th} rule be $A \to a_i$. To get the sentential form $a_1 \dots a_i A_1 \dots A_j$ in the i^{th} step, the $(i-1)^{\text{th}}$ sentential form has to be $a_1 \dots a_{i-1} A A_1 \dots A_j$. Then, by induction hypothesis, there exists the $(i-1)^{\text{th}}$ configuration $(A, A_1 \dots A_j)$ in the run ρ.
   By construction, there exists a transition $(A, a_i, A_1, (\uparrow, A_1)) \in T$. It follows that a possible i^{th} configuration is $(A_1, A_2 \dots A_j)$ and the word read until then is $a_1 \dots a_i$.
   The direction from run to derivation works similarly.
2. Let the i^{th} rule be $A \to a_i A_1$. To get the i^{th} sentential form as assumed in the claim, the $(i-1)^{\text{th}}$ sentential form has to be $a_1 \dots a_{i-1} A A_2 \dots A_j$. Then, by induction hypothesis, there exists the $(i-1)^{\text{th}}$ configuration $(A, A_2 \dots A_j)$ in the run ρ.
   By construction, there exists a transition $(A, a_i, A_1, \#) \in T$. It follows that a possible i^{th} configuration is $(A_1, A_2 \dots A_j)$ and the word read until then is $a_1 \dots a_i$.
   The other direction works similarly.
3. Let the i^{th} rule be $A \to a_i A_1 A_2$. To get the i^{th} sentential form as assumed in the claim, the $(i-1)^{\text{th}}$ sentential form has to be $a_1 \dots a_{i-1} A A_3 \dots A_j$. Then, by induction hypothesis, there exists the $(i-1)^{\text{th}}$ configuration $(A, A_3 \dots A_j)$ in the run ρ.
   By construction, there exists a transition $(A, a_i, A_1, (\downarrow, A_2)) \in T$. It follows that a possible i^{th} configuration is $(A_1, A_2 \dots A_j)$ and the word read until then is $a_1 \dots a_i$.
   The other direction works similarly.

This proves the claim.

Let $w = a_1 a_2 \dots \in \Sigma^\omega$. Now,

$$
\begin{aligned}
w \in \mathcal{L}(M) \text{ iff } & \exists \text{ run } \rho \text{ of } M \text{ on } w \text{ and } \mathrm{Inf}(\mathrm{state}(\rho)) \cap F \neq \emptyset \\
& \text{iff } \exists \text{ run } \rho \colon (S, \epsilon) \xrightarrow{a_1} \dots \xrightarrow{a_i} (A_1, A_2 \dots A_j) \to \dots \text{ of } M \\
& \qquad\qquad \text{and } \mathrm{Inf}(\mathrm{state}(\rho)) \cap F \neq \emptyset \\
& \text{iff } \exists \text{ derivation } \delta \colon S \to \dots \to a_1 \dots a_i A_1 \dots A_j \to \dots \text{ of } G \\
& \qquad\qquad \text{and } \mathrm{Inf}(\delta) \cap F \neq \emptyset \\
& \text{iff } \exists \text{ successful derivation } \delta \text{ of } G \text{ on } w \\
& \text{iff } w \in \mathcal{L}(G)\,.
\end{aligned}
$$

The equivalence of the second and third line is due to the claim above. □

Example 2.10. Let $G = (N, \Sigma, P, S, F)$ be a Büchi-accepting ω-context-free grammar with $N = \{S, M, B\}$, $\Sigma = \{a, b\}$, $F = \{S\}$ and P contains the following rules:

$$
\begin{aligned}
S &\to aMS \\
M &\to b \mid aMB \\
B &\to b
\end{aligned}
$$

Then G is in quadratic Greibach normal form. Note that the non-terminal M derives a string $a^n b^{n+1}$ for $n \in \mathbb{N}$ and the non-terminal S prepends another a. Thus, $\mathcal{L}(G) =$

$\{a^n b^n \mid n \geq 1\}^\omega$. By the construction of the proof of Theorem 2.9, G can be transformed into the ωPDA of Example 2.8. Note that Equation (2.3) generates a rule for every non-terminal in the grammar. As there are no transitions that push M on the stack, we omit its pop-rule here. ▽

Now we summarize our results obtained so far.

Corollary 2.11. *Let $L \subseteq \Sigma^\omega$. The following are equivalent:*

(*i*) *L is ωPDA-recognizable,*
(*ii*) *L is accepted by some ω-pushdown automaton,*
(*iii*) *L is contained in the ω-Kleene closure of context-free languages,*
(*iv*) *L is generated by some Muller-accepting ω-context-free grammar,*
(*v*) *L is generated by some Büchi-accepting ω-context-free grammar in quadratic Greibach normal form.*

Proof. The notions of ω-Kleene closure and general ω-pushdown automata were given by Cohen and Gold (1977) (see also the ϵ-transition-free variant in Definition 2.14); ω-Kleene closure was also used in the proof of Lemma 2.4.

The proof is performed by the following steps:
$(i) \Rightarrow (ii)$*:* trivial by definition,
$(ii) \Leftrightarrow (iii) \Leftrightarrow (iv)$*:* shown by Cohen and Gold (1977, Theorem 4.1.8),
$(iii) \Rightarrow (v)$*:* follows from Lemma 2.4,
$(v) \Rightarrow (i)$*:* follows from Theorem 2.9. □

2.3 Excursion: Simple Pushdown Automata

This section investigates some other possibilities to construct simple pushdown automata. Namely, the fist subsection treats simple pushdown automata of finite words. The second subsection shows how to transform ω-pushdown automata that are already realtime into simple ω-pushdown automata.

2.3.1 Finite Words

A similar idea as in Theorem 2.9 was used by Blass and Gurevich (2006) to show that every context-free language is the projection of some regular nested word language. We give here an idea how the construction for finite words looks like. We first need some definitions. Most of them are analogous to the definitions of ωPDA above, but for finite words, we accept by final state.

A *simple pushdown automaton* (PDA) (of finite words) over the alphabet Σ is a tuple $M = (Q, \Gamma, T, I, F)$ which syntactically is defined exactly as ωPDAs.

For the semantics, the notions *configuration* and *label* and the *successor relation* $\vdash_M^t$ are directly inherited from ωPDA.

We recall here the definition of a run. A finite sequence of transitions $\rho = (t_i)_{0 \le i \le m}$ with $t_i \in T$ is called a *run* of $M = (Q, \Gamma, T, I, F)$ on $w = \text{label}(\rho)$ iff there exists a finite sequence of configurations $(q_i, \gamma_i)_{0 \le i \le m}$ such that $(q_i, \gamma_i) \vdash_M^{t_i} (q_{i+1}, \gamma_{i+1})$ for each $0 \le i < m$ with $\gamma_0 = \epsilon$. Now, different to infinite words, this run is called *successful* if $q_m \in F$ and if $\gamma_m = \epsilon$.

For a PDA $M = (Q, \Gamma, T, I, F)$ over Σ, the language *accepted* by M is denoted by $\mathcal{L}(M) = \{w \in \Sigma^* \mid \exists \text{ successful run of } M \text{ on } w\}$. A language $L \subseteq \Sigma^*$ is called *PDA-recognizable* if there exists a PDA M with $\mathcal{L}(M) = L$.

Theorem 2.12. *Every context-free language is PDA-recognizable.*

The proof is in some sense similar to the one for Theorem 2.9. But we need a new dedicated final state and therefore also transitions leading to that state.

Proof. Let L be a context-free language. As shown by Autebert, Berstel and Boasson (1997), we can assume that L is generated by a context-free grammar $G = (N, \Sigma, P, S)$ in *quadratic Greibach normal form* (this is defined exactly as for infinite words, cf. Definition 2.3). For finite words, we exceptionally accept one ϵ-production for the start symbol S if S is never used on the right side of a production. This allows $\epsilon \in \mathcal{L}(G)$.

Assuming $\boldsymbol{F} \notin N$, we construct a PDA $A = (Q, \Gamma, T, I, F)$ over Σ with $Q = N \uplus \{\boldsymbol{F}\}$, $\Gamma = N$, $I = \{S\}$,

$$F = \begin{cases} \{\boldsymbol{F}, S\}, & \text{if } S \to \epsilon \in P \\ \{\boldsymbol{F}\}, & \text{otherwise}, \end{cases}$$

and

$$\begin{aligned} T &= \{(A, a, B, (\downarrow C)) \mid A \to aBC \in P\} \cup && (2.4)\\ &= \{(A, a, B, \#) \mid A \to aB \in P\} \cup && (2.5)\\ &= \{(A, a, B, (\uparrow B)) \mid A \to a \in P, B \in N\} \cup && (2.6)\\ &= \{(A, a, \boldsymbol{F}, \#) \mid A \to a \in P\}. && (2.7) \end{aligned}$$

Intuitively, the variable on the left side and the first variable on the right side of the productions are simulated by states. The second variable on the right side of the productions is pushed to the stack to store it for later. Whenever a final production is processed (Equations (2.6) and (2.7)), it is checked if another non-terminal is waiting on the stack to be processed before completing the run. The automaton model ensures that runs end with an empty stack.

Let $w = \epsilon$. Then,

$$\begin{aligned} w \in \mathcal{L}(G) &\text{ iff } S \to \epsilon \in P \\ &\text{iff } S \in F \\ &\text{iff } w = \epsilon \in \mathcal{L}(A). \end{aligned}$$

For words $w \neq \epsilon$, we give here only an exemplified version of the proof as it is in fact similar to the proof for infinite words above. For an intuition, note that an example derivation of the form

$$\delta\colon S \overset{S\to aA}{\Longrightarrow} aA \overset{A\to bBC}{\Longrightarrow} abBC \overset{B\to cDE}{\Longrightarrow} abcDEC \overset{D\to d}{\Longrightarrow} abcdEC \overset{E\to e}{\Longrightarrow} abcdeC \overset{C\to f}{\Longrightarrow} abcdef$$

corresponds to the run

$$\rho\colon (S,\epsilon) \xrightarrow{a,\#} (A,\epsilon) \xrightarrow{b,(\downarrow,C)} (B,C) \xrightarrow{c,(\downarrow,E)} (D,EC) \xrightarrow{d,(\uparrow,E)} (E,C) \xrightarrow{e,(\uparrow,C)} (C,\epsilon) \xrightarrow{f,\#} (\boldsymbol{F},\epsilon)$$

of the automaton. Let $\alpha = a\dots cAB\dots C \in T^jN^k$ be the i^{th} word in the derivation δ of G and let $\beta \in Q \times \Gamma^l$ be the i^{th} configuration in the run ρ of A. Then the terminals $a\dots c$ in α have been the labels of the transitions of ρ before the i^{th} step. Furthermore, the non-terminals $AB\dots C$ of α correspond exactly to the i^{th} configuration $(A, B\dots C)$ of ρ. This shows that derivations of G and runs of A are bijectively translated.

Note that the state $\boldsymbol{F}$ is always a sink. The construction allows the automaton to switch to state $\boldsymbol{F}$ before the input word is completely processed. But the definition of a successful run ensures that those runs are not successful. We can infer $\mathcal{L}(G) = \mathcal{L}(A)$. □

Example 2.13. Let $G = (N, \Sigma, P, S)$ be a grammar with $N = \{S, M, B\}$, $\Sigma = \{a, b\}$ and

$$\begin{aligned} S &\to \epsilon \mid aB \mid aMB \\ M &\to aB \mid aMB \\ B &\to b\,. \end{aligned}$$

Then G is in quadratic Greibach normal form and $\mathcal{L}(G) = \{a^nb^n \mid n \in \mathbb{N}\}$. By the construction above, it can be transformed into the PDA depicted in Fig. 2.2. Note that Equation (2.6) generates a rule for every final production rule and every non-terminal in the grammar. As there are no transitions that push S, M or $\boldsymbol{F}$ on the stack, we omit their pop-rules here. ▽

2.3.2 Simplify Realtime ω-Pushdown Automata

This subsection shows how to transform pushdown automata of infinite words that have already no ϵ-transitions into simple ω-pushdown automata. Similar constructions should work for the weighted case but have not been extensively treated yet.

We assume we have a general ω-pushdown automaton without ϵ-transitions. We show in four steps how to transform this automaton into a ωPDA.

Definition 2.14. An (unweighted) *general ω-pushdown automaton without ϵ-transitions* (ωGPDA) over Σ denotes a tuple $M = (Q, \Gamma, T, I, Z_0, F)$ where

- Q, Γ, I and F are defined as for ωPDA,

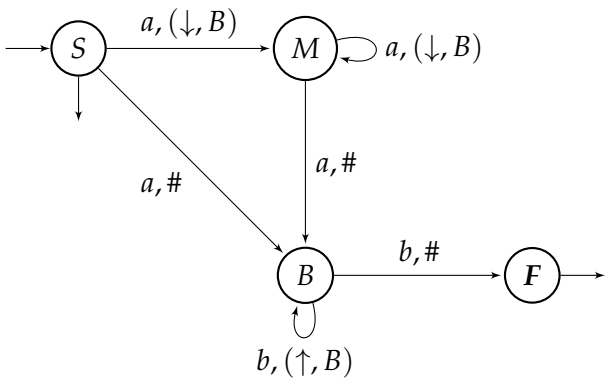


Figure 2.2: Automaton constructed in Example 2.13

- $T \subseteq Q \times \Sigma \times \Gamma \times Q \times \Gamma^*$ is a finite set of transitions and
- $Z_0 \in \Gamma$ is the initial stack symbol. ▼

A *configuration* for an ωGPDA is defined as for ωPDA. Now, for an ωGPDA $M = (Q, \Gamma, T, I, Z_0, F)$ and $t = (q, \sigma, A, q', \beta) \in T$, we let $(q, A\gamma) \vdash_M^t (q', \beta\gamma)$. The notions *label* and *state* are defined as above (see page 16).

An infinite sequence of transitions $\rho = (t_i)_{i \geq 0}$ with $t_i \in T$ is called a *run* of the ωGPDA $M = (Q, \Gamma, T, I, Z_0, F)$ on $w = \text{label}(\rho)$ iff there exists an infinite sequence of configurations $(q_i, \gamma_i)_{i \geq 0}$ such that $(q_i, \gamma_i) \vdash_M^{t_i} (q_{i+1}, \gamma_{i+1})$ for each $i \geq 0$ with $q_0 \in I$ and $\gamma_0 = Z_0$. A run ρ is called *successful* if $\text{Inf}(\text{state}(\rho)) \cap F \neq \emptyset$.

We will later use another type of automaton as an intermediate step in the transformation: An *ωGPDA which starts with an empty stack* $M = (Q, \Gamma, T, I, F)$ is defined similarly to an ωGPDA but for the run defined above, we require $\gamma_0 = \epsilon$ as for ωPDA and therefore, the transition relation is of the more general type $T \subseteq Q \times \Sigma \times (\Gamma \cup \{\epsilon\}) \times Q \times \Gamma^*$.

For an ωGPDA, the language *accepted* by M is denoted by $\mathcal{L}(M) = \{w \in \Sigma^\omega \mid \exists \text{ successful run of } M \text{ on } w\}$.

Transformation The transformation is split into four parts. Each part is described in its own paragraph.

We will start with an ωGPDA without ϵ-transitions. In Step 1, we transform it into an ωGPDA that starts with an empty stack instead of using Z_0. This automaton can therefore "pop" ϵ from the stack in its transitions.

The second step is to mark the lowest symbol on the stack. This step is optional in the unweighted case. It is added here as an option to avoid adding ambiguity in Step 4. This should make it possible to use the transformation in the weighted case, too.

The third step reduces the amount of symbols pushed onto the stack per transition. The result will be an ωGPDA that increases the stack size by at most one symbol per transition.

The last step then transforms the stack access to the three available stack commands in ωPDA, thus effectively removing the ability of the automaton to read and react to

the topmost stack symbol. Said differently, the last transformation changes the stack accesses from "pop one symbol and push some" to either "pop one", "push one" or ignore the stack.

Step 1: Starting with an empty stack

Lemma 2.15. *Let $L = \mathcal{L}(\mathcal{A})$ for an ωGPDA $\mathcal{A} = (Q, \Gamma, T, I, Z_0, F)$ over Σ.*

Then $L = \mathcal{L}(\mathcal{B})$ for an ωGPDA $\mathcal{B} = (Q', \Gamma, T', \{q_0\}, F)$ over Σ that starts with an empty stack and therefore has transitions $T' \subseteq Q' \times \Sigma \times (\Gamma \cup \{\epsilon\}) \times Q' \times \Gamma^*$. Additionally, $\mathcal{B}$ is initial state normalized, i.e., for all transitions $(p, a, \gamma, q, \gamma') \in T'$, we have $q \neq q_0$, and $\mathcal{B}$ only uses ϵ-stack-transitions, i.e., transitions $(p, a, \epsilon, q, \gamma)$, in the initial state.

Proof. Let $Q' = Q \uplus \{q_0\}$. The new set of transitions will be

$$T' = T \cup \{(q_0, a, \epsilon, q, \alpha) \mid (q_0, a, Z_0, q, \alpha) \in T, q_0 \in I, \alpha \in \Gamma^*\}.$$

Intuitively, the first transition is used to place the original initial stack symbol at the right place.

Remember that computations in $\mathcal{A}$ cannot be continued whenever the stack becomes empty. In $\mathcal{B}$, this property is fulfilled because from every original state, there exist only original transitions that pop something from the stack. In old initial states q, self-loops that empty the stack are translated to transitions from q_0 to q and for q the above again holds. □

Step 2: Marking the lowest stack symbol This step is only needed in the weighted case to make Step 4 (Lemma 2.18) possible.

For a set Θ, we define $\bar{\Theta} = \{\bar{s} \mid s \in \Theta\}$. We further set $\ddot{\Theta} = \Theta \cup \bar{\Theta}$. We will use this pattern to distinguish non-terminals in $\ddot{\Gamma}$: From now on, $A \in \Gamma$, $\bar{A} \in \bar{\Gamma}$ and for both possibilities, we write $\ddot{A} \in \ddot{\Gamma}$.

Lemma 2.16. *Let $L = \mathcal{L}(\mathcal{A})$ for an ωGPDA $\mathcal{A} = (Q, \Gamma, T, \{q_0\}, F)$ over Σ that starts with an empty stack (and therefore has transitions $T \subseteq Q \times \Sigma \times (\Gamma \cup \{\epsilon\}) \times Q \times \Gamma^*$), is initial state normalized and only uses ϵ-stack-transitions in the initial state.*

Then $L = \mathcal{L}(\mathcal{B})$ for an ωGPDA $\mathcal{B} = (Q, \ddot{\Gamma}, T', \{q_0\}, F)$ over Σ that starts with an empty stack and marks the lowest symbol A on the stack as $\bar{A}$.

Proof. The new set of transitions will be

$$\begin{aligned}
T' = &\{(q_0, a, \epsilon, q, \epsilon) && \mid (q_0, a, \epsilon, q, \epsilon) \in T\} \\
&\cup \{(q_0, a, \epsilon, q, A_1 \cdots A_{l-1} \bar{A}_l) && \mid (q_0, a, \epsilon, q, A_1 \cdots A_l) \in T, l \geq 1\} \\
&\cup \{(p, a, Z, q, \epsilon), (p, a, \bar{Z}, q, \epsilon) && \mid (p, a, Z, q, \epsilon) \in T\} \\
&\cup \{(p, a, \bar{Z}, q, A_1 \cdots A_{l-1} \bar{A}_l), (p, a, Z, q, A_1 \cdots A_l) && \mid (p, a, Z, q, A_1 \cdots A_l) \in T, l \geq 1\}.
\end{aligned}$$

We mark the lowest stack symbol when we first write it in the initial state q_0. As $\mathcal{A}$ is initial state normalized, we only do that once. In every other state, it is possible to pop the marked symbol, and in this case, the lowest pushed symbol will again be marked.

Note that $\mathcal{A}$ only uses ϵ-stack-transitions in state q_0 and therefore, empty stacks will never be refilled outside of state q_0. This means, when popping the marked symbol, we do not need to mark any other symbol afterwards. □

Step 3: Restricting the size of stack accesses

Lemma 2.17. *Let $L = \mathcal{L}(\mathcal{A})$ for an ωGPDA $\mathcal{A} = (Q, \Gamma, T, \{q_0\}, F)$ over Σ that starts with an empty stack and therefore has transitions $T \subseteq Q \times \Sigma \times (\Gamma \cup \{\epsilon\}) \times Q \times \Gamma^*$.*

Then $L = \mathcal{L}(\mathcal{B})$ for an ωGPDA $\mathcal{B} = (Q, \Gamma', T', \{q_0\}, F)$ over Σ that starts with an empty stack and every transition increases the stack size by at most one, i.e., for all transitions $(p, a, \alpha, q, \beta) \in T'$, we have $|\beta| - |\alpha| \leq 1$.

Additionally, if $\mathcal{A}$ marks the lowest symbol A on the stack as $\bar{A}$, then so does $\mathcal{B}$.

Proof. Let $n = \max\{|\alpha| \mid (p, a, Z, q, \alpha) \in T\}$ be the longest stack access in $\mathcal{A}$. Then let $\Gamma' = \Gamma^{\leq n}$ where we define $A^{\leq n}$ by $A \cup A^2 \cup \ldots \cup A^n$. The new set of transitions will be

$$\begin{aligned}
T' = \quad & \{(p, a, \epsilon, q, \epsilon) \mid (p, a, \epsilon, q, \epsilon) \in T\} \\
& \cup \{(p, a, \epsilon, q, (A_1, \ldots, A_l)) \mid (p, a, \epsilon, q, A_1 \cdots A_l) \in T, 1 \leq l \leq n\} \\
& \cup \{(p, a, (Z, X_1, \ldots, X_l), q, (X_1, \ldots, X_l)) \\
& \qquad \mid (p, a, Z, q, \epsilon) \in T, 0 \leq l \leq n-1, Z \neq \epsilon, X_i \in \Gamma\} \\
& \cup \{(p, a, (Z, X_1, \ldots, X_l), q, (A, X_1, \ldots, X_l)) \\
& \qquad \mid (p, a, Z, q, A) \in T, 0 \leq l \leq n-1, Z \neq \epsilon, X_i \in \Gamma\} \\
& \cup \{(p, a, (Z, X_1, \ldots, X_l), q, (A_1, \ldots, A_l)(X_1, \ldots, X_l)) \\
& \qquad \mid (p, a, Z, q, A_1 \cdots A_l) \in T, 2 \leq l \leq n, 0 \leq l \leq n-1, Z \neq \epsilon, X_i \in \Gamma\}.
\end{aligned}$$

Note that (Z) and Z are used interchangeably. We further define that empty tuples $()$ evaluate to ϵ.

Now assume that $\mathcal{A}$ marks the lowest stack symbol A as $\bar{A}$. Define a function $f\colon \ddot{\Gamma}^{\leq n} \to \ddot{\Gamma}'$ as $f((X_1, \ldots, \overline{X_l})) = \overline{(X_1, \ldots, X_l)}$ for $1 \leq l \leq n$ and $f = id_{\Gamma^{\leq n}}$ for all other cases. Then f applied to Z and γ in all transitions $(p, a, Z, q, \gamma) \in T'$ of $\mathcal{B}$ yields the desired result of $\mathcal{B}$ also marking the lowest stack symbol. □

Step 4: Change stack access to simple commands

Lemma 2.18. *Let $L = \mathcal{L}(\mathcal{A})$ for an ωGPDA $\mathcal{A} = (Q, \ddot{\Gamma}, T, \{q_0\}, F)$ over Σ that starts with an empty stack and therefore has transitions $T \subseteq Q \times \Sigma \times (\ddot{\Gamma} \cup \{\epsilon\}) \times Q \times \ddot{\Gamma}^*$ where for all transitions $(p, a, \alpha, q, \beta) \in T$, we have $|\beta| - |\alpha| \leq 1$. Additionally, $\mathcal{A}$ marks the lowest stack symbol A as $\bar{A}$.*

Then $L = \mathcal{L}(\mathcal{B})$ for an ωPDA $\mathcal{B} = (Q', \ddot{\Gamma}, T', \{q_0'\}, F')$ over Σ.

Proof. The idea of the proof is to keep the topmost stack symbol only in local memory and not on the stack. In this way, $\mathcal{B}$ can simulate to exchange the topmost symbol.

Let $Q' = Q \times (\ddot{\Gamma} \cup \{\epsilon\})$, $q'_0 = (q_0, \epsilon)$ and $F' = F \times (\ddot{\Gamma} \cup \{\epsilon\})$. The transitions are as follows:

$$\begin{aligned}
T' = \quad & \{((p,Z),a,(q,\ddot{A}),\uparrow \ddot{A}) \mid (p,a,Z,q,\epsilon) \in T, \ddot{A} \in \ddot{\Gamma}\} \\
& \cup \{((p,Z),a,(q,\epsilon),\#) \mid (p,a,\bar{Z},q,\epsilon) \in T\} \\
& \cup \{((p,\ddot{Z}),a,(q,\ddot{A}),\#) \mid (p,a,\ddot{Z},q,\ddot{A}) \in T\} \\
& \cup \{((p,\ddot{Z}),a,(q,A_1),\downarrow \ddot{A}_2) \mid (p,a,\ddot{Z},q,A_1\ddot{A}_2) \in T\} \\
& \cup \{((p,\tau),a,(q,\tau),\#) \mid (p,a,\epsilon,q,\epsilon) \in T, \tau \in \ddot{\Gamma} \cup \{\epsilon\}\} \\
& \cup \{((p,\ddot{Z}),a,(q,A),\downarrow \ddot{Z}) \mid (p,a,\epsilon,q,A) \in T, \ddot{Z} \in \ddot{\Gamma}\} \\
& \cup \{((p,\epsilon),a,(q,\bar{A}),\#) \mid (p,a,\epsilon,q,\bar{A}) \in T\}
\end{aligned}$$

Note how the first two and the last two sets of transitions are distinguished by Z or A, respectively, being the lowest stack symbol or not.

The marked lowest stack symbol makes sure that states (q,ϵ) are not used unless the stack is really empty. In this way, we keep the original number of runs unchanged and, later on, do not change the weight.

As it does not depend on the stack for runs in the original automaton $\mathcal{A}$ to be successful, we allow all original states F to be final independently of the topmost stack symbol stored in the state. □

By consecutive application of the above four lemmas, we get the following result.

Corollary 2.19. *Let $\mathcal{A}$ be a ωGPDA. Then there exists a ωPDA $\mathcal{B}$ with $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A})$.*

Note that we have that result already by Theorem 2.9. But that theorem has much worse complexity results. It needs the transformation of the ωGPDA into a grammar and then the transformation of the grammar into Greibach normal form. In the case that the ω-pushdown automaton already has no ϵ-transitions, the transformation can be done much faster and more direct.

Note also that we conducted the transformation such that the same constructions can probably be adapted to weighted automata (cf. Section 5.2). We can simply transfer the original weights to the new transitions in every construction. As all runs are translated one-to-one, the weights should be the same in the new automata. We can also state it differently: The construction does not add new ambiguity to any of the automata. In this way, a weighted automaton does not have the problem that some weight of one of its runs counts multiple times to the final weight if this was not the case in the original automaton.

2.4 Matching ω-Logic

The goal of this section is to find a Büchi-type logical formalism that is expressively equivalent to ωPDA. This extends the work of Lautemann, Schwentick and Thérien (1994) who defined a logic for context-free languages of finite words. They use first-order logic and equivalently monadic second-order logic (MSO) together with one second-order variable that has to define a matching. Their proof uses context-free grammars in a symmetric version of the Greibach normal form. Here we use a translation from automata and therefore, we will use the monadic second-order approach.

This section is guided by the procedure of Droste and Perevoshchikov (2015a). Their main result is a logical characterization of timed pushdown languages for finite words.

Let $w \in \Sigma^\omega$ be an ω-word. The set of all positions of w is $\mathbb{N}$.

Definition 2.20 (Lautemann, Schwentick and Thérien, 1994)**.** A binary relation $M \subseteq \mathbb{N} \times \mathbb{N}$ is a *matching* if

- M is compatible with $<$, i.e., $(i,j) \in M$ implies $i<j$,
- each element i belongs to at most one pair in M,
- M is non-crossing, i.e., $(i,j) \in M$ and $(k,l) \in M$ with $i<k<j$ implies $i<l<j$.

Let $\mathrm{Match}(\mathbb{N})$ denote the set of all matchings in $\mathbb{N} \times \mathbb{N}$. ▼

Let V_1, V_2 denote countable and pairwise disjoint sets of first-order and second-order variables. We fix a *matching variable* $\mu \notin V_1 \cup V_2$. Let $\mathcal{V} = V_1 \cup V_2 \cup \{\mu\}$.

We will define the logic in two steps. In the first layer of the logic, $\omega\mathrm{MSO}(\Sigma)$, the matching variable μ is unbounded. The second layer, $\omega\mathrm{ML}(\Sigma)$, existentially bounds this variable. We will show in Section 2.6 that $\omega\mathrm{ML}(\Sigma)$ is expressively equivalent to ωPDA. As an intermediate step in the corresponding proof, we will use the fact from Section 2.5 that $\omega\mathrm{MSO}(\Sigma)$ is expressively equivalent to visibly pushdown ω-automata.

Definition 2.21. Let Σ be an alphabet. The set $\omega\mathrm{MSO}(\Sigma)$ of *matching ω-MSO formulas* over Σ is defined by the extended Backus-Naur form (EBNF)

$$\varphi ::= P_a(x) \mid x \le y \mid x \in X \mid \mu(x,y) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\, \varphi \mid \exists X.\, \varphi$$

where $a \in \Sigma$, $x, y \in V_1$ and $X \in V_2$. ▼

Positions in the word will later be assigned to variables in φ. Here, $P_a(x)$ is a unary predicate indicating that the x^{th} letter of the word is a. Furthermore, $\mu(x,y)$ says that x and y will be matched.

A *$\mathcal{V}$-assignment* is a mapping $\sigma\colon \mathcal{V} \to \mathbb{N} \cup 2^{\mathbb{N}} \cup \mathrm{Match}(\mathbb{N})$ such that $\sigma(V_1) \subseteq \mathbb{N}$, $\sigma(V_2) \subseteq 2^{\mathbb{N}}$ and $\sigma(\mu) \in \mathrm{Match}(\mathbb{N})$.

Let σ be a $\mathcal{V}$-assignment. For $x \in V_1$ and $j \in \mathbb{N}$, the update $\sigma[x/j]$ is the $\mathcal{V}$-assignment σ' with $\sigma'(x) = j$ and $\sigma'(y) = \sigma(y)$ for all $y \in \mathcal{V} \setminus \{x\}$. The update $\sigma[X/J]$ for $X \in V_2$ and $J \subseteq \mathbb{N}$ and the update $\sigma[\mu/M]$ for $M \in \mathrm{Match}(\mathbb{N})$ are defined similarly.

Let $\varphi \in \omega\mathrm{MSO}(\Sigma)$. Furthermore, let $w = a_0 a_1 \ldots \in \Sigma^\omega$ and σ be a $\mathcal{V}$-assignment. We define $(w,\sigma) \models \varphi$ inductively over the structure of φ as shown in Table 2.1, where

$$
\begin{aligned}
(w,\sigma) &\models P_a(x) && \text{iff } a_{\sigma(x)} = a \\
(w,\sigma) &\models x \leq y && \text{iff } \sigma(x) \leq \sigma(y) \\
(w,\sigma) &\models x \in X && \text{iff } \sigma(x) \in \sigma(X) \\
(w,\sigma) &\models \mu(x,y) && \text{iff } (\sigma(x),\sigma(y)) \in \sigma(\mu) \\
(w,\sigma) &\models \neg\varphi && \text{iff } (w,\sigma) \not\models \varphi \\
(w,\sigma) &\models \varphi \vee \psi && \text{iff } (w,\sigma) \models \varphi \text{ or } (w,\sigma) \models \psi \\
(w,\sigma) &\models \exists x.\, \varphi && \text{iff } \exists j \in \mathbb{N}.\, (w,\sigma[x/j]) \models \varphi \\
(w,\sigma) &\models \exists X.\, \varphi && \text{iff } \exists J \subseteq \mathbb{N}.\, (w,\sigma[X/J]) \models \varphi
\end{aligned}
$$

Table 2.1: The semantics of $\omega\mathrm{MSO}(\Sigma)$ formulas

$a \in \Sigma$, $x, y \in V_1$ and $X \in V_2$. The logical counterparts $\wedge$, $\rightarrow$, $\forall x.\, \phi$ and $\forall X.\, \phi$ can be gained in the usual way from negation and the existing operators.

We now define $\mathrm{MATCHING}(\mu) \in \omega\mathrm{MSO}(\Sigma)$ which ensures that μ is matching.

Definition 2.22. Let

$$
\begin{aligned}
\mathrm{MATCHING}(\mu) = {} & \forall x \forall y.\, (\mu(x,y) \rightarrow x < y) \wedge \\
& \forall x \forall y \forall k.\, \big((\mu(x,y) \wedge k \neq x \wedge k \neq y) \rightarrow \neg\mu(x,k) \wedge \neg\mu(k,x) \wedge \neg\mu(y,k) \wedge \neg\mu(k,y)\big) \\
& \wedge \forall x \forall y \forall k \forall l.\, \big((\mu(x,y) \wedge \mu(k,l) \wedge x < k < y) \rightarrow x < l < y\big),
\end{aligned}
$$

where $x \neq y$, $x < y$ and $i < j < k$ have the usual translation. ▼

Definition 2.23. We let $\omega\mathrm{ML}(\Sigma)$, the set of formulas of *matching ω-logic* over Σ, be the set of all formulas ψ of the form

$$\psi = \exists \mu.\, (\varphi \wedge \mathrm{MATCHING}(\mu)),$$

for short $\psi = \exists^{\mathrm{match}} \mu.\, \varphi$, where $\varphi \in \omega\mathrm{MSO}(\Sigma)$. ▼

Let $w \in \Sigma^\omega$ and σ be a $\mathcal{V}$-assignment. Then, $(w,\sigma) \models \psi$ if there exists a matching $M \subseteq \mathbb{N}^2$ such that $(w, \sigma[\mu/M]) \models \varphi$.

Let $\psi \in \omega\mathrm{ML}(\Sigma)$. We denote by $\mathrm{Free}(\psi) \subseteq \mathcal{V}$ the set of *free variables* of ψ. A formula ψ with $\mathrm{Free}(\psi) = \emptyset$ is called a *sentence*. For a sentence ψ, the validity of $(w,\sigma) \models \psi$ does not depend on σ. Therefore, σ will be omitted and we only write $w \models \psi$. We denote by $\mathcal{L}(\psi) = \{w \in \Sigma^\omega \mid w \models \psi\}$ the language *defined* by ψ. A language $L \subseteq \Sigma^\omega$ is *ωML-definable* if there exists a sentence $\psi \in \omega\mathrm{ML}(\Sigma)$ such that $\mathcal{L}(\psi) = L$.

Example 2.24. Let

$$
\begin{aligned}
\varphi = {} & \forall x \exists y.\, (\mu(x,y) \vee \mu(y,x)) \wedge [(P_a(x) \wedge P_a(y)) \vee (P_b(x) \wedge P_b(y))] \\
& \wedge \neg\exists x \exists y \exists z_1 \exists z_2 \exists z_3 \exists z_4.\, x < z_1 < z_2 < z_3 < z_4 < y \wedge \mu(x,y) \wedge \mu(z_1,z_2) \wedge \mu(z_3,z_4)
\end{aligned}
$$

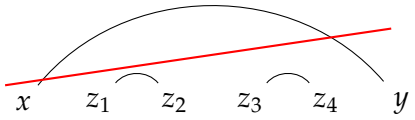


Figure 2.3: Forbidden pattern in Example 2.24

and consider the formula $\psi = \exists^{\text{match}}\mu.\, \varphi$.

The first line of the formula states that every letter is matched and that matched pairs always have the same letter (in this example, we assume $\Sigma = \{a, b\}$.). The second line forbids two *consecutive* matchings between two matched positions like in Figure 2.3.

Now, the language defined by ψ is an infinite sequence of palindromes:

$$\mathcal{L}(\psi) = L^\omega \qquad \text{for} \qquad L = \{ww^R \mid w \in \Sigma^*\}$$

where for $w = a_1 a_2 \dots a_n$, we define $w^R = a_n a_{n-1} \dots a_1$. ▽

The following will be the second main result.

Theorem 2.25. *Let Σ be an alphabet and $L \subseteq \Sigma^\omega$ an ω-language. Then L is ωML-definable if and only if L is ωPDA-recognizable.*

After some preparations, this theorem will be proved in Section 2.6.

2.5 Visibly Pushdown ω-Languages

It turns out that the ωMSO(Σ) formulas correspond exactly to the MSO-logic defined for visibly pushdown ω-languages (Alur and Madhusudan, 2004). In fact, without considering the existential quantification over the matching relation $\exists^{\text{match}}\mu$, the matching must explicitly be encoded in the words; the result is a nested word. For the convenience of the reader, we recall nested words and visibly pushdown languages (Alur and Madhusudan, 2004, 2009) in this section.

A *nested alphabet* is a triple $\tilde{\Sigma} = (\Sigma^\downarrow, \Sigma^\#, \Sigma^\uparrow)$ with $\Sigma^\downarrow$, $\Sigma^\#$ and $\Sigma^\uparrow$ being pairwise disjoint sets of *push*, *internal* and *pop* letters, respectively. Let $\hat{\Sigma} = \Sigma^\downarrow \cup \Sigma^\# \cup \Sigma^\uparrow$.

Definition 2.26. A *visibly pushdown ω-automaton* (ωVPA) over the nested alphabet $\tilde{\Sigma}$ is a 6-tuple $M = (Q, \Gamma, T, I, F)$ where

- Q is a finite set of states,
- Γ is a finite stack alphabet,
- $T = T^\downarrow \cup T^\# \cup T^\uparrow$ is a set of transitions, with
  - $T^\downarrow \subseteq Q \times \Sigma^\downarrow \times Q \times (\{\downarrow\} \times \Gamma)$,
  - $T^\# \subseteq Q \times \Sigma^\# \times Q \times \{\#\}$,
  - $T^\uparrow \subseteq Q \times \Sigma^\uparrow \times Q \times (\{\uparrow\} \times \Gamma)$,
- I is a set of initial states and
- F is a set of (Büchi accepting) final states. ▼

The following definitions are mostly similar to the ones for ωPDAs. The only difference is that the definition for T above restricts push transitions to push letters, pop transitions to pop letters and internal transitions to internal letters.

A *configuration* of an ωVPA $M = (Q, \Gamma, T, I, F)$ is a pair (q, γ), where $q \in Q$ and $\gamma \in \Gamma^*$. We define the transition relation as follows. Let $\gamma \in \Gamma^*$. For $t = (q, \sigma, q', (\downarrow, A)) \in T^{\downarrow}$, we write $(q, \gamma) \vdash_M^t (q', A\gamma)$. For $t = (q, \sigma, q', \#) \in T^{\#}$, we write $(q, \gamma) \vdash_M^t (q', \gamma)$. Finally, for $t = (q, \sigma, q', (\uparrow, A)) \in T^{\uparrow}$, we write $(q, A\gamma) \vdash_M^t (q', \gamma)$.

We denote by $\text{state}(q, \sigma, q', s) = q$ the state and by $\text{label}(q, \sigma, q', s) = \sigma$ the *label* of a transition. Both are extended to infinite sequence of transitions by letting $\text{state}((t_i)_{i\geq 0}) = (\text{state}(t_i))_{i\geq 0} \in Q^\omega$ for the infinite sequence of states and similarly $\text{label}((t_i)_{i\geq 0}) = (\text{label}(t_i))_{i\geq 0} \in \hat{\Sigma}^\omega$ for the infinite word constructed from the labels of the transitions.

We call an infinite sequence of transitions $\rho = (t_i)_{i\geq 0}$ with $t_i \in T$ a *run* of M on $w = \text{label}(\rho) \in \hat{\Sigma}^\omega$ iff there exists an infinite sequence of configurations $(q_i, \gamma_i)_{i\geq 0}$ with $q_0 \in I$ and $\gamma_0 = \epsilon$ such that $(q_i, \gamma_i) \vdash_M^{t_i} (q_{i+1}, \gamma_{i+1})$ for each $i \geq 0$.

A run ρ is called *successful* if $\text{Inf}(\text{state}(\rho)) \cap F \neq \emptyset$.

Definition 2.27. For an ωVPA $M = (Q, \Gamma, T, I, F)$ over $\tilde{\Sigma}$, the language *accepted* by M is denoted by $\mathcal{L}(M) = \{w \in \hat{\Sigma}^\omega \mid \exists \text{ successful run of } M \text{ on } w\}$. A language $L \subseteq \hat{\Sigma}^\omega$ is *ωVPA-recognizable with respect to* $\tilde{\Sigma}$ if there exists an ωVPA $M = (Q, \Gamma, T, I, F)$ over $\tilde{\Sigma}$ with $\mathcal{L}(M) = L$. ▼

Note that the ωVPA-recognizable languages with respect to $\tilde{\Sigma}$ form a proper subclass of the ωPDA-recognizable languages over $\hat{\Sigma}$.

Also note that the definition here differs from the definition of Alur and Madhusudan (2004) by the way how we handle the empty stack. Alur and Madhusudan allow their automata to check if the stack is empty by reading the special symbol $\perp$. The ωVPA does not allow this check directly. But the ωVPA can duplicate all its states and stack symbols to allow us to distinguish between states with empty stack and those with non-empty stack. Whenever pushing a symbol onto the empty stack, this new stack symbol has to contain the extra information that upon popping that symbol, the automaton has to change to an empty-stack state afterwards. Compare this to Lemma 2.16.

Let $f : \hat{\Sigma} \to \hat{\Sigma}'$ be a mapping. It *respects nesting* if $f(\Sigma^{\downarrow}) \subseteq \Sigma'^{\downarrow}$, $f(\Sigma^{\#}) \subseteq \Sigma'^{\#}$ and $f(\Sigma^{\uparrow}) \subseteq \Sigma'^{\uparrow}$. The mapping will be extended to words in the natural way.

Theorem 2.28 (Alur and Madhusudan, 2004)**.** *Let $L_1 \subseteq \hat{\Sigma}^\omega$ and $L_2 \subseteq \hat{\Sigma}^\omega$ be ωVPA-recognizable with respect to $\tilde{\Sigma}$. Then $L_1 \cup L_2$, $L_1 \cap L_2$, $\hat{\Sigma}^\omega \setminus L_1$ are ωVPA-recognizable with respect to $\tilde{\Sigma}$. If $f : \hat{\Sigma} \to \hat{\Sigma}'$ is a mapping that respects nesting, then $f(L_1)$ is ωVPA-recognizable with respect to $\tilde{\Sigma}'$.*

We now discuss how the logic ωMSO has the same expressive power as ωVPAs. Note that all $\omega\text{MSO}(\Sigma)$ formulas may contain the free variable μ. In the following we will not differentiate between a word $w \in \hat{\Sigma}^\omega$ and the pair $(\pi(w), \sigma)$ where σ is

a $\{\mu\}$-assignment mapping μ to the matching relation encoded in w. We extend the semantics definitions as follows. Let $\varphi \in \omega\mathrm{MSO}(\Sigma)$ and $\mathrm{Free}(\varphi) \subseteq \{\mu\}$, then we define

$$\mathcal{L}_{\mathrm{nw}}(\varphi) = \{(w, \sigma(\mu)) \mid w \in \Sigma^\omega,\ \sigma \text{ is a } \{\mu\}\text{-assignment with } (w,\sigma) \models \varphi\}\,.$$

Note that pairs (w, ν) with $\nu \in \mathrm{Match}(\mathbb{N})$ are called nested words by Alur and Madhusudan (2009). Now, we describe how nested words are encoded into a nested alphabet, which we call here *tagged alphabet*, $\mathrm{tag}(\Sigma) = (\{a^\downarrow \mid a \in \Sigma\}, \{a^\# \mid a \in \Sigma\}, \{a^\uparrow \mid a \in \Sigma\})$ where the *tagged* letters $a^\downarrow$, $a^\#$ and $a^\uparrow$ are not occurring in Σ. For that, consider a word $w = a_0 a_1 \cdots$ and a $\{\mu\}$-assignment σ. The encoding for $(w, \sigma(\mu))$ is the *tagged* word $\tilde{w} = \tilde{a}_0 \tilde{a}_1 \cdots$ where $\tilde{a}_i = a_i^\downarrow$ if there exists a position y with $\mu(i, y)$, and $\tilde{a}_i = a_i^\uparrow$ if there exists a position x with $\mu(x, i)$, and $\tilde{a}_i = a_i^\#$ otherwise. This encoding will be extended to sets of nested words like $\mathcal{L}_{\mathrm{nw}}(\varphi)$ in the natural way. The underlying alphabet will be called $\hat{\Sigma}_{\mathrm{tag}} = \{\sigma^\downarrow \mid \sigma \in \Sigma\} \cup \{\sigma^\# \mid \sigma \in \Sigma\} \cup \{\sigma^\uparrow \mid \sigma \in \Sigma\}$.

Theorem 2.29 (Alur and Madhusudan, 2004)**.** *Let $L \subseteq \hat{\Sigma}_{tag}^\omega$. Then, L is ωVPA-recognizable with respect to $tag(\Sigma)$ iff there is an $\omega MSO(\Sigma)$-formula φ with $Free(\varphi) \subseteq \{\mu\}$ and $\mathcal{L}_{nw}(\varphi) = L$.*

The mapping $\pi\colon \hat{\Sigma}_{\mathrm{tag}} \to \Sigma$ removes the "tag" of a tagged letter. Thus, π maps $a^\downarrow$, $a^\#$ and $a^\uparrow$ to a. This can be extended to words by letting $\pi(a_0 a_1 \cdots) = \pi(a_0)\pi(a_1)\cdots$ and to languages $L \subseteq \hat{\Sigma}_{\mathrm{tag}}^\omega$ by setting $\pi(L) = \{\pi(w) \mid w \in L\}$.

Lemma 2.30. *Let $L \subseteq \hat{\Sigma}_{tag}^\omega$ be ωVPA-recognizable with respect to $tag(\Sigma)$. Then $\pi(L) \subseteq \Sigma^\omega$ is ωPDA-recognizable.*

This has been proved by Blass and Gurevich (2006) for finite nested words. Here, we present their proof adopted similarly to infinite words.

Proof. Let $A = (Q, \Gamma, T, I, F)$ be an ωVPA over $\mathrm{tag}(\Sigma)$ with $\mathcal{L}(A) = L$. We construct an ωPDA $B = (Q, \Gamma, T', I, F)$ over Σ such that $\mathcal{L}(B) = \pi(L)$. Let

$$T' = \{(q, \pi(a), p, s) \mid (q, a, p, s) \in T\}\,.$$

Now,

$$\begin{aligned}
\mathcal{L}(B) &= \{w \in \Sigma^\omega \mid \exists \text{ successful run of } B \text{ on } w\} \\
&= \{w \in \Sigma^\omega \mid \exists \text{ successful run } \rho \text{ of } B \text{ on } w = w_0 w_1 \cdots \text{ and} \\
&\qquad\qquad \rho\colon (q_0, \gamma_0) \xrightarrow{w_0} (q_1, \gamma_1) \xrightarrow{w_1} \cdots\} \\
&= \{w \in \Sigma^\omega \mid \exists \text{ run } \rho \text{ of } B \text{ on } w \text{ with } \mathrm{Inf}(\mathrm{state}(\rho)) \cap F \neq \emptyset \text{ and} \\
&\qquad\qquad \rho\colon (q_0, \gamma_0) \xrightarrow{w_0} (q_1, \gamma_1) \xrightarrow{w_1} \cdots \text{ and} \\
&\qquad\qquad w = w_0 w_1 \cdots = \pi(v_0)\pi(v_1)\cdots = \pi(v)\}
\end{aligned}$$

$$
\begin{aligned}
&= \{w \in \Sigma^\omega \mid w = \pi(v) \text{ and } \exists \text{ run } \rho' \text{ of } A \text{ on } v = v_0 v_1 \cdots \text{ with} \\
&\qquad \operatorname{Inf}(\operatorname{state}(\rho')) \cap F \neq \emptyset \text{ and } \rho' \colon (q_0, \gamma_0) \xrightarrow{v_0} (q_1, \gamma_1) \xrightarrow{v_1} \cdots \} \\
&= \{\pi(v) \mid v \in \hat{\Sigma}^\omega_{\text{tag}} \text{ and } \exists \text{ successful run } \rho' \text{ of } A \text{ on } v = v_0 v_1 \cdots \text{ with} \\
&\qquad \rho' \colon (q_0, \gamma_0) \xrightarrow{v_0} (q_1, \gamma_1) \xrightarrow{v_1} \cdots \} \\
&= \pi\{v \in \hat{\Sigma}^\omega_{\text{tag}} \mid \exists \text{ successful run of } A \text{ on } v\} \\
&= \pi(\mathcal{L}(A)) = \pi(L)\,. \qquad \square
\end{aligned}
$$

2.6 Equivalence of Logic and Context-Free ω-Languages

This section proves the expressive equivalence of the full logic and simple ω-pushdown automata. Together with our results on simple ω-pushdown automata, this shows the expressive equivalence of the logic and ω-context-free languages.

Let $a = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n$ be a tuple. Then we define for $1 \leq i \leq n$ the i^{th} *projection* of a by $\operatorname{pr}_i(a) = a_i$.

Lemma 2.31. *Let Σ be an alphabet and $L \subseteq \Sigma^\omega$ be ωPDA-recognizable. Then L is ωML-definable.*

Proof. By assumption, there exists an ωPDA $\mathcal{A} = (Q, \Sigma, \Gamma, T, I, F)$ with $\mathcal{L}(\mathcal{A}) = L$. Let $T = \{t_1, \ldots, t_m\}$ be an enumeration of the transitions. We define an ωML-sentence ψ such that $\mathcal{L}(\psi) = L$ as follows. We hereby proceed similarly to the lines of Droste and Gastin (2007).

First, we will need an auxiliary formula

$$\operatorname{next}(x, y) = x < y \wedge \neg(\exists z.\, x < z \wedge z < y)\,.$$

We define the set of second-order variables $\mathcal{V} = \{X_t \mid t \in T\}$. The following three $\omega\text{MSO}(\Sigma)$ formulas ψ_{part}, ψ_{comp} and ψ_{final} will be used. We let

$$\psi_{\text{part}} = \forall x. \bigvee_{t \in T} \Big(x \in X_t \wedge \bigwedge_{t' \in T\colon\, t \neq t'} x \notin X_{t'}\Big)\,.$$

Then ψ_{part} ensures that the variables X_t form a partitioning of the positions x. Now let

$$\psi_{\text{comp}} = \forall x. \Big(\varphi_{\text{first}}(x) \wedge \bigwedge_{t \in T} \big(x \in X_t \to (\varphi_1(x,t) \wedge \varphi_2(x,t) \wedge \varphi_3(x,t))\big) \wedge \varphi_4(x)\Big)\,,$$

and let

$$\varphi_{\text{first}}(x) = \forall y.\, (x \leq y) \to \bigvee_{t \in T\colon \mathrm{pr}_1(t) \in I} x \in X_t\,,$$
$$\varphi_1(x, (q, a, q', s)) = P_a(x)\,,$$
$$\varphi_2(x, (q, a, q', s)) = \forall y.\, \big(next(x, y) \to \bigvee_{t' \in T\colon \mathrm{pr}_1(t') = q'} y \in X_{t'}\big)\,,$$
$$\varphi_3(x, (q, a, q', (\uparrow, A))) = \exists y.\, \Big(\bigvee_{t \in T\colon \mathrm{pr}_4(t) = (\downarrow, A)} y \in X_t \wedge \mu(y, x)\Big)\,,$$
$$\varphi_4(x) = \forall y.\, \mu(x, y) \to \bigvee_{\substack{t, t' \in T\colon \mathrm{pr}_4(t) = (\downarrow, A),\\ \mathrm{pr}_4(t') = (\uparrow, A'),\\ A = A'}} (x \in X_t \wedge y \in X'_t)\,.$$

Then ψ_{comp} ensures that there exists a run of the automaton. Note how we simulate the stack of the automaton: The formula φ_3 expresses a sufficient condition and the formula φ_4 expresses a necessary condition for a pair (x, y) to be matched.

The necessary condition φ_4 restricts the amount of possible instances of the matching variable μ. Without this restriction, the constructed ωML-formula would allow two positions to be matched even if they were not connected by a push and pop transition in the original automaton. While we use it here for the sake of completeness, φ_4 is indispensable in the weighted case (see Chapter 5) because an automaton with, e.g., only one possible run must be translated into a formula with only one possible instantiation of the matching variable μ.

Finally, ψ_{final} controls the acceptance condition that final states have to occur infinitely often in a successful run:

$$\psi_{\text{final}} = \forall x \exists y.\, \Big(x < y \wedge \bigvee_{t \in T\colon \mathrm{pr}_1(t) \in F} y \in X_t\Big)\,.$$

Let $\psi_{\text{successful}} \in \omega\text{MSO}(\Sigma)$ be the following boolean formula that is true only for successful runs of M:

$$\psi_{\text{successful}} = \psi_{\text{part}} \wedge \psi_{\text{comp}} \wedge \psi_{\text{final}} \tag{2.8}$$

Now, let $\psi \in \omega\text{ML}(\Sigma)$ be the sentence defined as

$$\psi = \exists^{\text{match}} \mu.\, \psi_{\text{successful}}\,.$$

Then,

$$\begin{aligned}\mathcal{L}(\psi) &= \{w \in \Sigma^\omega \mid \exists \text{ matching } M \text{ s.t. } (w, \varnothing[\mu/M]) \models \psi_{\text{part}} \wedge \psi_{\text{comp}} \wedge \psi_{\text{final}}\}\\ &= \{w \in \Sigma^\omega \mid \exists \text{ successful run of } \mathcal{A} \text{ on } w\}\\ &= \mathcal{L}(\mathcal{A}) = L\,.\end{aligned}$$

□

The other direction uses the corresponding results for visibly pushdown languages.

Lemma 2.32. *Let Σ be an alphabet and $L \subseteq \Sigma^\omega$ be ωML-definable. Then L is ωPDA-recognizable.*

Proof. Let $\psi = \exists^{\text{match}} \mu.\varphi \in \omega\text{ML}(\Sigma)$ be a sentence with $\mathcal{L}(\psi) = L$.

We know $\varphi \in \omega\text{MSO}(\Sigma)$ with $\text{Free}(\varphi) \subseteq \{\mu\}$. Let $L' = \mathcal{L}_{\text{nw}}(\varphi)$. By Theorem 2.29, $L' \subseteq \hat{\Sigma}^\omega$ is ωVPA-recognizable with respect to $\tilde{\Sigma} = (\{\sigma^\downarrow \mid \sigma \in \Sigma\}, \{\sigma^\# \mid \sigma \in \Sigma\}, \{\sigma^\uparrow \mid \sigma \in \Sigma\})$.

The remaining existential quantification over the matching relation μ is exactly the projection π from ωVPA-recognizable languages to ωPDA-recognizable languages. By Lemma 2.30, $L = \pi(L')$ is ωPDA-recognizable. □

Proof of Theorem 2.25. This theorem is immediate by Lemmas 2.31 and 2.32. □

Concluding, we can summarize:

Corollary 2.33. *Let $L \subseteq \Sigma^\omega$. The following are equivalent:*

(*i*) *L is ωPDA-recognizable,*
(*ii*) *L is an ω-context-free language,*
(*iii*) *L is ωML-definable.*

CHAPTER 3

Weighted Greibach Normal Form

The goal of this chapter is the Greibach normal form of weighted context-free languages of infinite words. As by Kuich and Salomaa (1986) and Ésik and Kuich (2007a), the weighted context-free languages of finite and infinite words are described by solutions of ω-algebraic systems and mixed ω-algebraic systems of equations. In the main result of this chapter, we show that these systems can be transformed into a Greibach normal form.

In the literature, Greibach normal forms, central for context-free languages of finite words, have been established for ω-context-free languages (of infinite words) (Cohen and Gold, 1977) and also for algebraic systems of equations for series over finite words (Kuich and Salomaa, 1986; Ésik and Kuich, 2007a); this latter result is employed in our proof. Hence, here we extend these classical results to a weighted version for infinite words.

The Preliminaries given in this chapter describe also the basics needed for the next chapter, Chapter 4. Then, in Section 3.2, we characterize ω-algebraic series by a series of equivalent statements.

In Section 3.3, we define mixed ω-algebraic systems and their canonical solutions. The main result of section states that each ω-algebraic series is a component of a canonical solution of a mixed ω-algebraic system in Greibach normal form.

In Section 3.4 we specialize the main result of Section 3.3: now each ω-algebraic series is a component of a canonical solution of an ω-algebraic system in Greibach normal form.

This chapter is based on Droste, Dziadek and Kuich (2019b, 2020c).

3.1 Preliminaries

For the convenience of the reader, we recall definitions and results by Ésik and Kuich (2007a).

A monoid $\langle S, +, 0\rangle$ is called *complete* if it is equipped with sum operations $\sum_I$ for all families $(a_i \mid i \in I)$ of elements of S, where I is an arbitrary index set, such that the following conditions are satisfied (see Conway, 1971; Eilenberg, 1974; Kuich, 1997):

$$\text{(i)} \quad \sum_{i\in\emptyset} a_i = 0, \qquad \sum_{i\in\{j\}} a_i = a_j, \qquad \sum_{i\in\{j,k\}} a_i = a_j + a_k \text{ for } j \neq k\,,$$

$$\text{(ii)} \quad \sum_{j\in J}\Big(\sum_{i\in I_j} a_i\Big) = \sum_{i\in I} a_i\,, \text{ if } \bigcup_{j\in J} I_j = I \text{ and } I_j \cap I_{j'} = \emptyset \text{ for } j \neq j'\,.$$

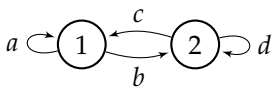


Figure 3.1: Depiction of matrix $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ as automaton

Furthermore, a semiring $\langle S, +, \cdot, 0, 1 \rangle$ is called *complete* if $\langle S, +, 0 \rangle$ is a complete monoid and if we additionally have

$$\text{(iii)} \quad \sum_{i \in I} (c \cdot a_i) = c \cdot \big(\sum_{i \in I} a_i\big), \qquad \sum_{i \in I} (a_i \cdot c) = \big(\sum_{i \in I} a_i\big) \cdot c\,.$$

This means that a semiring S is complete if it has "infinite sums" (i) that are an extension of the finite sums, (ii) that are associative and commutative and (iii) that satisfy the distributivity laws.

A semiring S equipped with an additional unary star operation $^* : S \to S$ is called a *starsemiring*. In complete semirings for each element a, the *star* a^* of a is defined by

$$a^* = \sum_{j \geq 0} a^j\,.$$

Hence, each complete semiring is a starsemiring, called a *complete starsemiring*.

Starsemirings allow us to generalize the star operation to matrices. Let $M \in S^{n \times n}$, then we define $M^* \in S^{n \times n}$ inductively as in Ésik, Kuich Ésik and Kuich (2007a), pp. 14–15 as follows. For $n = 1$ and $M = (a)$, for $a \in S$, we let $M^* = (a^*)$. Now, for $n > 1$, we partition M into submatrices, called *blocks*,

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\,, \tag{3.1}$$

with $a \in S^{1 \times 1}$, $b \in S^{1 \times (n-1)}$, $c \in S^{(n-1) \times 1}$, $d \in S^{(n-1) \times (n-1)}$, and we define

$$M^* = \begin{pmatrix} (a + bd^*c)^* & (a + bd^*c)^* bd^* \\ (d + ca^*b)^* ca^* & (d + ca^*b)^* \end{pmatrix}\,. \tag{3.2}$$

Whenever we use a matrix M as defined in (3.1), the corresponding automaton can be illustrated as in Figure 3.1.

A semiring is called *continuous* if it is ordered, each directed subset has a least upper bound and addition and multiplication preserve the least upper bound of directed sets. Any continuous semiring is complete. See Ésik and Kuich (2007a) for background.

Suppose that S is a semiring and V is a commutative monoid written additively. We call V a (left) S-semimodule if V is equipped with a (left) action

$$\begin{aligned} S \times V &\to V \\ (s, v) &\mapsto sv \end{aligned}$$

subject to the following rules:

$$s(s'v) = (ss')v\,, \quad (s+s')v = sv + s'v\,, \quad s(v+v') = sv + sv'\,,$$
$$1v = v\,, \quad 0v = 0\,, \quad s0 = 0\,,$$

for all $s, s' \in S$ and $v, v' \in V$. If V is an S-semimodule, we call (S, V) a *semiring-semimodule pair*.

Suppose that (S, V) is a semiring-semimodule pair such that S is a starsemiring and S and V are equipped with an omega operation $^\omega : S \to V$. Then we call (S, V) a *starsemiring-omegasemimodule pair*.

Ésik and Kuich (2007b) define a *complete semiring-semimodule pair* to be a semiring-semimodule pair (S, V) such that S is a complete semiring and V is a complete monoid with

$$s\Big(\sum_{i\in I} v_i\Big) = \sum_{i\in I} sv_i \qquad \text{and} \qquad \Big(\sum_{i\in I} s_i\Big)v = \sum_{i\in I} s_i v\,,$$

for all $s \in S$, $v \in V$, and for all families $(s_i)_{i\in I}$ over S and $(v_i)_{i\in I}$ over V; moreover, it is required that an *infinite product operation*

$$S^\omega \ni (s_1, s_2, \ldots) \mapsto \prod_{j\geq 1} s_j \in V$$

is given mapping infinite sequences over S to V subject to the following three conditions:

$$\text{(i)} \quad \prod_{i\geq 1} s_i = \prod_{i\geq 1} (s_{n_{i-1}+1} \cdot \cdots \cdot s_{n_i})\,,$$
$$\text{(ii)} \quad s_1 \cdot \prod_{i\geq 1} s_{i+1} = \prod_{i\geq 1} s_i\,,$$
$$\text{(iii)} \quad \prod_{j\geq 1} \sum_{i_j \in I_j} s_{i_j} = \sum_{(i_1, i_2, \ldots)\in I_1\times I_2\times \ldots} \prod_{j\geq 1} s_{i_j}\,,$$

where in the first equation $0 = n_0 \leq n_1 \leq n_2 \leq \ldots$ and $I_1, I_2, \ldots$ are arbitrary index sets. This means that the left action of the semimodule is distributive and it is required that it has "infinite products" mapping infinite sequences over S to V such that the product (i) can be partitioned (an infinite form of associativity), (ii) can be extended from the left and (iii) satisfies an infinite distributivity law.

Suppose that (S, V) is complete. Then we define

$$s^* = \sum_{i\geq 0} s^i \qquad \text{and} \qquad s^\omega = \prod_{i\geq 1} s\,,$$

for all $s \in S$. This turns (S, V) into a starsemiring-omegasemimodule pair. Observe that, if (S, V) is a complete semiring-semimodule pair, then $0^\omega = 0$.

A *star-omega semiring* is a semiring S equipped with unary operations * and $^\omega : S \to S$. A star-omega semiring S is called *complete* if (S, S) is a complete semiring-semimodule pair, i.e., if S is complete and is equipped with an infinite product operation that satisfies the three conditions stated above. A complete star-omega semiring S is called *continuous* if the semiring S is continuous.

Example 3.1. Formal languages are covered by our model. Let $\langle \mathbb{B}, +, \cdot, 0, 1 \rangle$ be the Boolean semiring. Then let $0^* = 1^* = 1$ and take infima as infinite products. This makes $\mathbb{B}$ a continuous star-omega and commutative semiring. It then follows that $\mathbb{B}\langle\langle \Sigma^* \rangle\rangle \times \mathbb{B}\langle\langle \Sigma^\omega \rangle\rangle$ is isomorphic to formal languages of finite and infinite words with the usual operations.

The semiring $\langle \mathbb{N}^\infty, +, \cdot, 0, 1 \rangle$ with $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ and the natural infinite product operation of numbers is a continuous star-omega and commutative semiring.

The tropical semiring $\langle \mathbb{N}^\infty, \min, +, \infty, 0 \rangle$ with the usual infinite sum operation as infinite product is a commutative semiring and a continuous star-omega semiring.

Analogously, the arctic semiring $\langle \bar{\mathbb{N}}, \max, +, -\infty, 0 \rangle$ with $\bar{\mathbb{N}} = \mathbb{N} \cup \{-\infty, \infty\}$ and the infinite sum operation as infinite product is a commutative semiring and a continuous star-omega semiring. ▽

A *Conway semiring* (see Conway, 1971; Bloom and Ésik, 1993) is a starsemiring S satisfying the *sum star identity*

$$(a + b)^* = a^*(ba^*)^*$$

and the *product star identity*

$$(ab)^* = 1 + a(ba)^*b$$

for all $a, b \in S$. Observe that by Ésik and Kuich (2007a, Theorem 1.2.24), each complete starsemiring is a Conway semiring.

Note that from the identities in Conway semirings, it follows

$$\begin{aligned} a^* &= 1 + aa^* = 1 + a^*a\,, \\ a(ba)^* &= (ab)^*a\,, \end{aligned} \tag{3.3}$$

for all $a, b \in S$.

If S is a Conway semiring then so is $S^{n\times n}$. Let $M \in S^{n\times n}$. Assume that $n > 1$ and write M as in (3.1). Applying the identities of Conway semirings, we get an equivalent definition (cf. Conway, 1971, pp. 27–28) to (3.2):

$$M^* = \begin{pmatrix} (a + bd^*c)^* & a^*b(d + ca^*b)^* \\ d^*c(a + bd^*c)^* & (d + ca^*b)^* \end{pmatrix}. \tag{3.4}$$

Following Bloom and Ésik (1993), we call a starsemiring-omegasemimodule pair (S, V) a *Conway semiring-semimodule pair* if S is a Conway semiring and if the omega operation satisfies the *sum omega identity* and the *product omega identity*:

$$(a + b)^\omega = (a^*b)^\omega + (a^*b)^*a^\omega \qquad \text{and} \qquad (ab)^\omega = a(ba)^\omega\,,$$

for all $a, b \in S$. By Ésik, Kuich Ésik and Kuich (2007b) each complete semiring-semimodule pair is a Conway semiring-semimodule pair.

Observe that the *omega fixed-point equation* holds, i.e.

$$aa^\omega = a^\omega\,,$$

for all $a \in S$.

Consider a starsemiring-omegasemimodule pair (S, V). Following Bloom and Ésik (1993), we define a matrix operation $^\omega \colon S^{n\times n} \to V^{n\times 1}$ on a starsemiring-omegasemimodule pair (S, V) as follows. If $n = 0$, M^ω is the unique element of V^0, and if $n = 1$, so that $M = (a)$, for some $a \in S$, $M^\omega = (a^\omega)$. Assume now that $n > 1$ and write M as in (3.1). Then

$$M^\omega = \begin{pmatrix} (a + bd^*c)^\omega + (a + bd^*c)^* bd^\omega \\ (d + ca^*b)^\omega + (d + ca^*b)^* ca^\omega \end{pmatrix}.$$

Additionally, the *matrix star identity* is valid for Conway semirings and states that the star of a matrix is independent of the partitioning of the matrix. The *matrix omega identity* is valid for Conway semiring-semimodule pairs and states that the operation $^\omega$ is independent of the partitioning of the matrix, i.e., the blocks of (3.1) can have arbitrary sizes: $a \in S^{n_1\times n_1}$, $b \in S^{n_1\times n_2}$, $c \in S^{n_2\times n_1}$, $d \in S^{n_2\times n_2}$ for $n_1 + n_2 = n$. If (S, V) is a Conway semiring-semimodule pair, then so is $(S^{n\times n}, V^n)$. See also Ésik and Kuich (2007a, p. 106).

Following Ésik and Kuich (2005b), we define matrix operations $^{\omega,t} \colon S^{n\times n} \to V^{n\times 1}$ for $0 \leq t \leq n$ as follows. Assume that $M \in S^{n\times n}$ is decomposed into blocks a, b, c, d as in (3.1), but with a of dimension $t \times t$ and d of dimension $(n - t) \times (n - t)$. Then

$$M^{\omega,t} = \begin{pmatrix} (a + bd^*c)^\omega \\ d^*c(a + bd^*c)^\omega \end{pmatrix}. \tag{3.5}$$

Observe that $M^{\omega,0} = 0$ and $M^{\omega,n} = M^\omega$. Intuitively, M can be interpreted as an adjacency matrix and $M^{\omega,t}$ are infinite paths where the first t states are repeated states, i.e., states that are Büchi-accepting.

The next theorem states that, in case of a Conway semiring, $M^{\omega,t}$, for $0 \leq t \leq n$, can be computed also in a way different from its definition and, with certain limits, is independent of the partitioning of the matrix M.

Theorem 3.2. *Let S be a Conway semiring and $0 \leq t \leq k \leq n$. Assume $M \in S^{n\times n}$ is decomposed into blocks*

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

with block a being of dimension $k \times k$ and block d of dimension $(n - k) \times (n - k)$.

Then we have,

$$M^{\omega,t} = \begin{pmatrix} (a + bd^*c)^{\omega,t} \\ d^*c(a + bd^*c)^{\omega,t} \end{pmatrix}. \tag{3.6}$$

Proof. The proof resembles the proof of the matrix omega identity (cf. Ésik and Kuich, 2007a, Theorem 5.3.13). Assume $M \in S^{n\times n}$ is decomposed into nine blocks

$$M = \begin{pmatrix} f & g & h \\ i & a & b \\ j & c & d \end{pmatrix}$$

with dimensions $f \in S^{t\times t}$, $a \in S^{(k-t)\times(k-t)}$ and $d \in S^{(n-k)\times(n-k)}$. Consider the following two partitionings:

$$M = \left(\begin{array}{c|cc} f & g & h \\ \hline i & a & b \\ j & c & d \end{array}\right) \qquad M' = \left(\begin{array}{cc|c} f & g & h \\ i & a & b \\ \hline j & c & d \end{array}\right)$$

Now we need to show that $M^{\omega,t}$, calculated as in (3.5)

$$M^{\omega,t} = \begin{pmatrix} \alpha \\ \begin{pmatrix} a & b \\ c & d \end{pmatrix}^* \begin{pmatrix} i \\ j \end{pmatrix} \alpha \end{pmatrix},$$

where

$$\alpha = \left(f + \begin{pmatrix} g & h \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}^* \begin{pmatrix} i \\ j \end{pmatrix}\right)^\omega$$

is equal to $M'^{\omega,t}$, calculated as in (3.6)

$$M'^{\omega,t} = \begin{pmatrix} \mu \\ d^* \begin{pmatrix} j & c \end{pmatrix} \mu \end{pmatrix},$$

where

$$\mu = \left(\begin{pmatrix} f & g \\ i & a \end{pmatrix} + \begin{pmatrix} h \\ b \end{pmatrix} d^* \begin{pmatrix} j & c \end{pmatrix}\right)^{\omega,t}.$$

In the case $t = k$, we have

$$M = M' = \begin{pmatrix} f & h \\ j & d \end{pmatrix}.$$

It follows that

$$\begin{aligned} \alpha &= (f + hd^*j)^\omega \\ &= (f + hd^*j)^{\omega,t} = \mu, \end{aligned}$$

where the second equality is due to t being the full dimension of $f + hd^*j$. The second components of $M^{\omega,t}$ and $M'^{\omega,t}$ then both reduce to $d^*j(f + hd^*j)^\omega$.

If $k = n$, we have

$$M = M' = \begin{pmatrix} f & g \\ i & a \end{pmatrix}.$$

Now, the second component of $M'^{\omega,t}$ and the second summand of μ have dimension 0 and thus

$$M'^{\omega,t} = \begin{pmatrix} f & g \\ i & a \end{pmatrix}^{\omega,t} = M^{\omega,t}$$

Hence, in the following, we can assume $t < k < n$.

First, we compute $M^{\omega,t}$. We denote the blocks of $M^{\omega,t}$ by $(M^{\omega,t})_i$ for $1 \leq i \leq 3$. Then we have

$$\begin{aligned}
(M^{\omega,t})_1 = \alpha &= \left(f + \begin{pmatrix} g & h \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix}^* \begin{pmatrix} i \\ j \end{pmatrix}\right)^\omega \\
&= \left(f + \begin{pmatrix} g & h \end{pmatrix} \begin{pmatrix} (a+bd^*c)^* & a^*b(d+ca^*b)^* \\ d^*c(a+bd^*c)^* & (d+ca^*b)^* \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix}\right)^\omega \\
&= \left(f + \begin{pmatrix} g & h \end{pmatrix} \begin{pmatrix} (a+bd^*c)^*i + a^*b(d+ca^*b)^*j \\ d^*c(a+bd^*c)^*i + (d+ca^*b)^*j \end{pmatrix}\right)^\omega \\
&= \big(f + g(a+bd^*c)^*i + ga^*b(d+ca^*b)^*j \\
&\qquad + hd^*c(a+bd^*c)^*i + h(d+ca^*b)^*j\big)^\omega .
\end{aligned}$$

Here, we used the star of a matrix in the form shown in (3.4). We will now compute the other two blocks by using the star of a matrix as in (3.2):

$$\begin{aligned}
\begin{pmatrix} (M^{\omega,t})_2 \\ (M^{\omega,t})_3 \end{pmatrix} &= \begin{pmatrix} a & b \\ c & d \end{pmatrix}^* \begin{pmatrix} i \\ j \end{pmatrix} \alpha \\
&= \begin{pmatrix} (a+bd^*c)^* & (a+bd^*c)^*bd^* \\ (d+ca^*b)^*ca^* & (d+ca^*b)^* \end{pmatrix} \begin{pmatrix} i \\ j \end{pmatrix} \alpha \\
&= \begin{pmatrix} (a+bd^*c)^*i + (a+bd^*c)^*bd^*j \\ (d+ca^*b)^*ca^*i + (d+ca^*b)^*j \end{pmatrix} \alpha \\
&= \begin{pmatrix} \big((a+bd^*c)^*i + (a+bd^*c)^*bd^*j\big)\alpha \\ \big((d+ca^*b)^*ca^*i + (d+ca^*b)^*j\big)\alpha \end{pmatrix}
\end{aligned}$$

Now, we compute $M'^{\omega,t}$. We denote the blocks of $M'^{\omega,t}$ by $(M'^{\omega,t})_i$ for $1 \leq i \leq 3$. Then we have

$$\begin{aligned}
\begin{pmatrix} (M'^{\omega,t})_1 \\ (M'^{\omega,t})_2 \end{pmatrix} = \mu &= \left(\begin{pmatrix} f & g \\ i & a \end{pmatrix} + \begin{pmatrix} h \\ b \end{pmatrix} d^* \begin{pmatrix} j & c \end{pmatrix}\right)^{\omega,t} \\
&= \left(\begin{pmatrix} f & g \\ i & a \end{pmatrix} + \begin{pmatrix} hd^*j & hd^*c \\ bd^*j & bd^*c \end{pmatrix}\right)^{\omega,t} \\
&= \begin{pmatrix} f+hd^*j & g+hd^*c \\ i+bd^*j & a+bd^*c \end{pmatrix}^{\omega,t} \\
&= \begin{pmatrix} \delta \\ (a+bd^*c)^*(i+bd^*j)\delta \end{pmatrix},
\end{aligned}$$

where

$$\delta = \big(f + hd^*j + (g+hd^*c)(a+bd^*c)^*(i+bd^*j)\big)^\omega .$$

It remains to calculate

$$\begin{aligned}
(M'^{\omega,t})_3 &= d^* \begin{pmatrix} j & c \end{pmatrix} \mu \\
&= d^*\big(j + c(a+bd^*c)^*(i+bd^*j)\big)\delta .
\end{aligned}$$

The last step is to verify the three equalities $(M^{\omega,t})_i = (M'^{\omega,t})_i$ for $1 \leq i \leq 3$. The first equality follows basically from Lemma 1.2.16 of Ésik and Kuich (2007a). We will mark the use of Lemma 1.2.16 by $\Diamond$ and obtain

$$\begin{aligned}
(M^{\omega,t})_1 &= \alpha \\
&= \big(f + g(a + bd^*c)^*i + ga^*b(d + ca^*b)^*j \\
&\qquad + hd^*c(a + bd^*c)^*i + h(d + ca^*b)^*j\big)^\omega \\
&\overset{\Diamond}{=} \big(f + hd^*j + g(a + bd^*c)^*i + g(a + bd^*c)^*bd^*j \\
&\qquad + hd^*c(a + bd^*c)^*i + hd^*c(a + bd^*c)^*bd^*j\big)^\omega \\
&= \big(f + hd^*j + g(a + bd^*c)^*(i + bd^*j) + hd^*c(a + bd^*c)^*(i + bd^*j)\big)^\omega \\
&= \big(f + hd^*j + (g + hd^*c)(a + bd^*c)^*(i + bd^*j)\big)^\omega \\
&= \delta = (M'^{\omega,t})_1
\end{aligned}$$

For the second equality, we have

$$\begin{aligned}
(M^{\omega,t})_2 &= \big((a + bd^*c)^*i + (a + bd^*c)^*bd^*j\big)\alpha \\
&= \big((a + bd^*c)^*(i + bd^*j)\big)\delta \\
&= (M'^{\omega,t})_2 .
\end{aligned}$$

Now, for the third equality, it suffices to prove

$$(d + ca^*b)^*ca^*i + (d + ca^*b)^*j = d^*\big(j + c(a + bd^*c)^*(i + bd^*j)\big) .$$

We have

$$\begin{aligned}
d^*\big(j + c(a + bd^*c)^*(i + bd^*j)\big) &= d^*j + d^*c(a + bd^*c)^*(i + bd^*j) \\
&= d^*j + d^*c(a + bd^*c)^*i + d^*c(a + bd^*c)^*bd^*j \\
&= d^*j + d^*c(a^*bd^*c)^*a^*i + d^*c(a^*bd^*c)^*a^*bd^*j \\
&= d^*j + (d^*ca^*b)^*d^*ca^*i + (d^*ca^*b)^*d^*ca^*bd^*j \\
&= (d^*ca^*b)^*d^*ca^*i + (d^*ca^*b)^*d^*ca^*bd^*j + d^*j \\
&= (d^*ca^*b)^*d^*ca^*i + \big((d^*ca^*b)^*d^*ca^*b + 1\big)d^*j \\
&= (d^*ca^*b)^*d^*ca^*i + (d^*ca^*b)^*d^*j \\
&= (d + ca^*b)^*ca^*i + (d + ca^*b)^*j .
\end{aligned}$$

Note that for this calculation, we rely heavily on commutativity of addition, distributivity and the sum star identity and the product star identity of Conway semirings together with their derived identities (3.3). This completes the proof. □

We will use quemirings as defined by Elgot (1976) and as elaborated on by Ésik and Kuich (2007a, pp. 109 ff.).

$$
\begin{aligned}
(x+y)+z &= x+(y+z)\\
x+y &= y+x\\
x+0 &= x\\
(x\cdot y)\cdot z &= x\cdot(y\cdot z)\\
x\cdot 1 &= x\\
1\cdot x &= x\\
(x+y)\cdot z &= (x\cdot z)+(y\cdot z)\\
0\cdot x &= 0\\
x\P\cdot(y+z) &= (x\P\cdot y)+(x\P\cdot z)\\
x &= x\P+(x\cdot 0)\\
x\P\cdot 0 &= 0\\
(x+y)\P &= x\P+y\P\\
(x\cdot y)\P &= x\P\cdot y\P
\end{aligned}
$$

Table 3.1: Axioms of quemirings

Consider a semiring-semimodule pair (S, V) and let $T = S \times V$. We define an addition given componentwise, i.e.,

$$(s, v) + (s', v') = (s + s', v + v'),$$

a semidirect product type multiplication (using that S acts on V), i.e.,

$$(s, v) \cdot (s', v') = (ss', v + sv'),$$

a unary operation

$$(s, v)\P = (s, 0),$$

and two constants $0 = (0, 0)$ and $1 = (1, 0)$.

Now, a *quemiring* is an algebraic structure $(T, +, \cdot, \P, 0, 1)$ with the above operations and constants satisfying the axioms in Table 3.1. Note that these axioms follow from the axioms of semiring-semimodule pairs applied to the above operations.

The constant 0 only behaves like a zero from the left. Additionally, a quemiring is not necessarily distributive from the left. However, if V is idempotent, then we indeed have

$$x \cdot (y + z) = x \cdot y + x \cdot z.$$

That makes a quemiring *quasi a semiring*.

While the operation ¶ *selects the first component*, multiplication with 0 from the right *selects the second component*, i.e.,

$$(s, v) \cdot 0 = (0, v).$$

For a quemiring T, we have that $S = T\P = \{x\P \mid x \in T\}$ is a semiring. Moreover, $V = T0 = \{x \cdot 0 \mid x \in T\}$ is a $T\P$-semimodule. Elgot (1976) showed that a quemiring T is isomorphic to a quemiring $S \times V$ determined by the semiring-semimodule pair (S, V). It follows that we can identify every element t of a quemiring T by a pair (s, v) of a semiring-semimodule pair (S, V). Note that subsequently in this thesis, we will only use quemirings $S \times V$.

Also, if (S, V) is a starsemiring-omegasemimodule pair, we define a star operation on $T = S \times V$, by

$$(s, v)^{\otimes} = (s^*, s^\omega + s^* v)\,,$$

making it a *generalized starquemiring* (see Ésik and Kuich, 2007a).

For an alphabet Σ, we call mappings r of Σ^* into S *series*. The collection of all such series r is denoted by $S\langle\langle\Sigma^*\rangle\rangle$. We call the set $\mathrm{supp}(r) = \{w \mid (r, w) \neq 0\}$ the *support* of a series r. The set of series with finite support $S\langle\Sigma^*\rangle = \{s \in S\langle\langle\Sigma^*\rangle\rangle \mid \mathrm{supp}(s) \text{ is finite}\}$ is called the set of *polynomials*. We denote by $S\langle\Sigma\rangle$, $S\langle\{\epsilon\}\rangle$ and $S\langle\Sigma \cup \{\epsilon\}\rangle$ the series with support in Σ, $\{\epsilon\}$ and $\Sigma \cup \{\epsilon\}$, respectively. Series s with $|\mathrm{supp}(s)| \leq 1$ are called *monomials*. Note that polynomials are finite sums of monomials.

Mappings of Σ^ω into S are called *ω-series* and their collection is denoted by $S\langle\langle\Sigma^\omega\rangle\rangle$. See Kuich and Salomaa (1986) and Ésik and Kuich (2007a) for more information. Examples of monomials in $S\langle\Sigma^*\rangle$ for a semiring $\langle S, +, \cdot, 0, 1\rangle$ are 0, w, sw for $s \in S$ and $w \in \Sigma^*$, defined by

$(0, w) = 0$ for all w,
$(w, w) = 1$ and $(w, w') = 0$ for $w \neq w'$,
$(sw, w) = s$ and $(sw, w') = 0$ for $w \neq w'$.

Note that $w = 1w$.

3.2 ω-Algebraic Systems

This section introduces (mixed) ω-algebraic systems and gives a characterization of ω-algebraic series. The ω-algebraic systems in some sense generalize context-free grammars of infinite words. In fact, ω-algebraic systems handle finite and infinite words at the same time. We will also use mixed ω-algebraic systems where the finite and the infinite words are divided into two parts.

In the sequel, x, y and z denote vectors of dimension n, i.e., $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$ and $z = (z_1, \ldots, z_n)$. Later, we will also use z of dimension m. It is clear by the context whether they are used as row or as column vectors. Similar conventions hold for vectors p, σ, ω and τ. Moreover, X denotes the set of variables $\{x_1, \ldots, x_n\}$ for $S\langle\langle\Sigma^*\rangle\rangle$, while $\{z_1, \ldots, z_n\}$ is the set of variables for $S\langle\langle\Sigma^\omega\rangle\rangle$. The set Y denotes the set of variables $\{y_i, \ldots, y_n\}$ for the quemiring $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$.

It makes sense to first define algebraic systems that are a generalization of context-free grammars of finite words. Let Σ be an alphabet and let S be a continuous semiring.

An *algebraic system* over $S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$ is a system of equations

$$x_i = p_i(x), \quad \text{for } p_i \in S\langle(\Sigma \cup X)^*\rangle \text{ and } 1 \leq i \leq n\,,$$

or equivalently

$$x = p(x), \quad \text{for } p \in (S\langle(\Sigma \cup X)^*\rangle)^{n\times 1}\,.$$

A *solution* of $x = p(x)$ is given by $\sigma \in (S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle)^n$ such that $\sigma = p(\sigma)$. We call a solution $(\sigma_1, \ldots, \sigma_n)$ *least solution* if

$$\sigma_i \leq \tau_i, \quad \text{for each } 1 \leq i \leq n,$$

for all solutions $(\tau_1, \ldots, \tau_n)$ of $x = p(x)$.

We can also generalize right-linear grammars in the same manner. An algebraic system $x_i = p_i(x)$ is called *linear system* if for each $1 \leq i \leq n$, we have

$$p_i = \sum_{1\leq j\leq n} M_{ij}x_j + R_i, \quad \text{where } M_{ij}, R_i \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle\,.$$

For a matrix $M \in (S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$ and a column vector $R \in (S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle)^{n\times 1}$, we can write the above system as

$$x = Mx + R\,.$$

We let $S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$, the *algebraic series*, be the collection of all components of least solutions of algebraic systems

$$x_i = p_i(x) \quad \text{where } p_i \in S\langle(\Sigma \cup X)^*\rangle \text{ for } 1 \leq i \leq n\,.$$

For the rest of this chapter, *S is a continuous, and therefore complete, star-omega semiring*. If we consider $S\langle\langle\Sigma^*\rangle\rangle$ or $S\langle\langle\Sigma^\omega\rangle\rangle$, then we assume additionally that *the underlying semiring S is commutative*. Let further Σ denote an alphabet.

By Theorem 5.5.5 of Ésik and Kuich (2007a), $(S\langle\langle\Sigma^*\rangle\rangle, S\langle\langle\Sigma^\omega\rangle\rangle)$ is a complete semiring-semimodule pair, hence a Conway semiring-semimodule pair, satisfying $\epsilon^\omega = 0$. Hence, $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ is a generalized starquemiring.

We will be working with two different generalizations of ω-context-free grammars, the ω-algebraic systems and the mixed ω-algebraic systems.

An *ω-algebraic system* over the quemiring $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ consists of an algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$

$$y = p(y), \quad \text{for } p \in (S\langle(\Sigma \cup Y)^*\rangle)^{n\times 1}\,.$$

The vector of quemiring elements $\tau \in (S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle)^n$ is a *solution* of the ω-algebraic system

$$y = p(y)\,,$$

if

$$\tau = p(\tau)\,.$$

Note that every p_i is a polynomial, i.e., a finite sum of monomials in $S\langle(\Sigma \cup Y)^*\rangle$. Let $y_i = (x_i, z_i)$, for $1 \leq i \leq n$. Now, we can apply the quemiring addition and multiplication to p.

Consider a monomial

$$t(y_1, \ldots, y_n) = s w_0 y_{i_1} w_1 \ldots w_{k-1} y_{i_k} w_k \,,$$

where $s \in S$ and $w_i \in \Sigma^*$ for $1 \leq i \leq k$. Note that from the quemiring operations, we have

$$t((x_1, z_1), \ldots, (x_n, z_n)) = (s w_0 x_{i_1} w_1 \ldots w_{k-1} x_{i_k} w_k,\ s w_0 z_{i_1} + s w_0 x_{i_1} w_1 z_{i_2} + \ldots + s w_0 x_{i_1} w_1 \cdots w_{k-2} x_{i_{k-1}} w_{k-1} z_{i_k}) \,.$$

Therefore, following Ésik and Kuich (2007a), we define

$$t_x(x_1, \ldots, x_n, z_1, \ldots, z_n) = s w_0 z_{i_1} + s w_0 x_{i_1} w_1 z_{i_2} + \ldots + s w_0 x_{i_1} w_1 \cdots w_{k-2} x_{i_{k-1}} w_{k-1} z_{i_k} \,,$$

and for a polynomial $p(y_1, \ldots, y_n) = \sum_{1 \leq j \leq m} t_j(y_1, \ldots, y_n)$, we let

$$p_x(x_1, \ldots, x_n, z_1, \ldots, z_n) = \sum_{1 \leq j \leq m} (t_j)_x(x_1, \ldots, x_n, z_1, \ldots, z_n) \,.$$

For an ω-algebraic system $y = p(y)$ over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$, we call $x = p(x)$, $z = p_x(x, z)$ the *mixed ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ induced by $y = p(y)$*.

In general, a *mixed ω-algebraic system* over the quemiring $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ consists of an algebraic system over $S\langle\langle\Sigma^*\rangle\rangle$

$$x = p(x), \quad p \in (S\langle(\Sigma \cup X)^*\rangle)^{n \times 1}$$

and a linear system over $S\langle\langle\Sigma^\omega\rangle\rangle$

$$z = \rho(x) z, \quad \rho \in (S\langle(\Sigma \cup X)^*\rangle)^{m \times m} \,.$$

The pair $(\sigma, \omega) \in (S\langle\langle\Sigma^*\rangle\rangle)^n \times (S\langle\langle\Sigma^\omega\rangle\rangle)^m$ is a *solution* of the mixed ω-algebraic system

$$x = p(x), \ \ z = \rho(x) z \,,$$

if

$$\sigma = p(\sigma), \ \ \omega = \rho(\sigma)\omega \,.$$

Observe that, by Theorem 5.5.1 of Ésik and Kuich (2007a), $\omega^{(k)} = \rho(\sigma)^{\omega,k}$ for each $1 \leq k \leq n$, is solution for the linear system

$$z = \rho(\sigma) z \,.$$

If σ is the least solution of $x = p(x)$, then $z = \rho(\sigma)z$ is an $S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$-linear system and $(\sigma, \omega^{(k)}) = (\sigma, \rho(\sigma)^{\omega,k})$, where $k \in \{0, 1, \ldots, m\}$, is called *$k^{\mathrm{th}}$-canonical solution* of

$x = p(x), z = \rho(x)z$. Observe that the k^{th} canonical solution is unique by definition. A solution (σ, ω) is called *canonical*, if there exists a k such that (σ, ω) is the k^{th}-canonical solution. The k^{th}-canonical solution of an ω-algebraic system $y = p(y)$ is defined to be the k^{th}-canonical solution of the mixed ω-algebraic system $x = p(x), z = p_x(x, z)$ induced by $y = p(y)$.

We let $S^{\text{alg}}\langle\langle\Sigma^\omega\rangle\rangle$, the *$\omega$-algebraic series*, be the collection of all components of vectors $M^{\omega,k}$, where $M \in (S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$, for $n \geq 1$ and $k \in \{1, \ldots, n\}$.

Moreover, we let ω-$\mathfrak{Rat}(S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle)$ be the ω-Kleene closure of (i.e., the generalized starquemiring generated by) $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$.

Example 3.3. Consider the following ω-algebraic system over the quemiring $\mathbb{B}\langle\langle\Sigma^*\rangle\rangle \times \mathbb{B}\langle\langle\Sigma^\omega\rangle\rangle$ for the Boolean semiring $\langle\mathbb{B}, +, \cdot, 0, 1\rangle$

$$\begin{aligned} y_1 &= y_2 y_1 + \epsilon \\ y_2 &= a y_2 b + \epsilon\,, \end{aligned}$$

where $a, b \in \Sigma$. This induces the following mixed ω-algebraic system

$$\begin{aligned} x_1 &= x_2 x_1 + \epsilon & \qquad z_1 &= z_2 + x_2 z_1 \\ x_2 &= a x_2 b + \epsilon & \qquad z_2 &= a z_2\,. \end{aligned}$$

Then for the algebraic system $x = p(x)$ over $\mathbb{B}\langle\langle\Sigma^*\rangle\rangle$, we get the least solution $\sigma_2 = \sum_{n\geq 0} a^n b^n$ and therefore $\sigma_1 = (\sum_{n\geq 0} a^n b^n)^*$. For the semimodule part, we can consider the first canonical solution where only z_1 is Büchi-accepting and the second canonical solution where both z_1 and z_2 are Büchi-accepting. The first canonical solution of the mixed ω-algebraic system $x = p(x), z = p_x(x, z)$ over $\mathbb{B}\langle\langle\Sigma^*\rangle\rangle \times \mathbb{B}\langle\langle\Sigma^\omega\rangle\rangle$ is then $(\sigma_1, \sigma_2; (\sum_{n\geq 0} a^n b^n)^\omega, 0)$. The second canonical solution would be $(\sigma_1, \sigma_2; (\sum_{n\geq 0} a^n b^n)^\omega + (\sum_{n\geq 0} a^n b^n)^* a^\omega, a^\omega)$. ▽

Example 3.4. We consider the following mixed ω-algebraic system over the quemiring $\mathbb{N}^\infty\langle\langle\Sigma^*\rangle\rangle \times \mathbb{N}^\infty\langle\langle\Sigma^\omega\rangle\rangle$ for the tropical semiring $\langle\mathbb{N}^\infty, \min, +, \infty, 0\rangle$

$$\begin{aligned} x_1 &= 1 a x_1 b + 1 a b & \qquad z_1 &= c z_1 \\ & & \qquad z_2 &= x_1 z_1 + z_1 \end{aligned}$$

where $a, b, c \in \Sigma$ and using the natural number 1.

Then for the algebraic system $x = p(x)$ over $\mathbb{N}^\infty\langle\langle\Sigma^*\rangle\rangle$, we get the least solution $\sigma = a^n b^n \mapsto n$. The first canonical solution of the mixed ω-algebraic system $x = p(x)$, $z = \rho(x)z$ over $\mathbb{N}^\infty\langle\langle\Sigma^*\rangle\rangle \times \mathbb{N}^\infty\langle\langle\Sigma^\omega\rangle\rangle$ is then $(\sigma, c^\omega \mapsto 0, a^n b^n c^\omega \mapsto n)$. Hence the series $a^n b^n c^\omega \mapsto n$ is ω-algebraic but it is clearly not recognizable by a weighted automaton without stack. ▽

Now we have the following characterization of ω-algebraic series.

Theorem 3.5. *Let S be a continuous complete star-omega semiring with the underlying semiring S being commutative and let Σ be an alphabet. Then the following statements are equivalent for $(s, v) \in S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$:*

(i) $(s, v) \in S^{alg}\langle\langle\Sigma^*\rangle\rangle \times S^{alg}\langle\langle\Sigma^\omega\rangle\rangle$,
(ii) $(s, v) \in \omega\text{-}\mathfrak{Rat}(S^{alg}\langle\langle\Sigma^*\rangle\rangle)$,
(iii) $(s, v) = \|\mathfrak{A}\|$, *where* $\mathfrak{A}$ *is a finite* $S^{alg}\langle\langle\Sigma^*\rangle\rangle$*-automaton over* $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$,
(iv) $s \in S^{alg}\langle\langle\Sigma^*\rangle\rangle$ *and* $v = \sum_{1\leq j\leq l} s_j t_j^\omega$ *for some* $l \geq 0$, *where* $s_j, t_j \in S^{alg}\langle\langle\Sigma^*\rangle\rangle$,
(v) (s, v) *is a component of a canonical solution of a mixed* ω*-algebraic system over* $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$.

Proof. The statements (ii), (iii) and (iv) are equivalent by Theorem 5.4.9 (see also Theorem 5.6.6) of Ésik and Kuich (2007a).

(iii)⇒(v): Assume that $(s, v) = \|\mathfrak{A}\|$, where $\mathfrak{A} = (n, I, M, P, k)$ is a finite $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$-automaton. Without loss of generality $\mathfrak{A}$ is normalized by Theorem 5.4.2 of Ésik and Kuich (2007a); i.e., $I = e_i$ for some i. Hence, $(s, v) = ((M^*P)_i, (M^{\omega,k})_i)$ is a component of the k^{th} canonical solution of the mixed ω-algebraic system

$$x = Mx + P, \quad z = Mz\,.$$

(v)⇒(i): Assume there exists a mixed ω-algebraic system $x = p(x), z = \rho(x)z$, with canonical solution $(\sigma, \rho(\sigma)^{\omega,k})$ such that $(s, v) = (\sigma_i, (\rho(\sigma)^{\omega,k})_j)$ for some i and j. Since the entries of σ and $\rho(\sigma)$ are in $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$, (s, v) is in $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\text{alg}}\langle\langle\Sigma^\omega\rangle\rangle$.

(i)⇒(iv): Now assume $s \in S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$ and $v = (M^{\omega,k})_i$ for some $M \in (S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$, $n \geq 1$, and $i, k \in \{1, \ldots, n\}$. By the definition of $M^{\omega,k}$, each entry of $M^{\omega,k}$ is of the form $\sum_{1\leq j\leq l} s_j t_j^\omega$ for some $l \geq 0$, where $s_j, t_j \in S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$ for $1 \leq j \leq l$. □

3.3 Greibach Normal Form for Mixed ω-Algebraic Systems

In this section we show that for any element (s, v) of $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\text{alg}}\langle\langle\Sigma^\omega\rangle\rangle$ there exists a mixed ω-algebraic system in Greibach normal form such that (s, v) is a component of a solution of this ω-algebraic system.

Similar to the definition for algebraic systems on finite words (see Chapter 4, p. 69 and cf. Greibach, 1965), a mixed ω-algebraic system

$$x = p(x), \quad z = \rho(x)z$$

is in *Greibach normal form* if

$$\begin{aligned} &\operatorname{supp}(p_i(x)) \subseteq \{\epsilon\} \cup \Sigma \cup \Sigma X \cup \Sigma X X, && \text{for all } 1 \leq i \leq n, \quad \text{and} \\ &\operatorname{supp}(\rho_{ij}(x)) \subseteq \Sigma \cup \Sigma X, && \text{for all } 1 \leq i, j \leq m\,. \end{aligned}$$

Note that in the previous chapter, Chapter 2, we called the normal form above *quadratic* Greibach normal form. As we will only consider the quadratic version from now on, we will skip the "quadratic" to save space in this and the subsequent chapters.

For the construction of the Greibach normal form we need a corollary to Theorem 3.5.

Corollary 3.6. *The following statement for* $(s, v) \in S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ *is equivalent to the statements* (i) *to* (v) *of Theorem 3.5:*
$s \in S^{alg}\langle\langle\Sigma^*\rangle\rangle$ *and* $v = \sum_{1\le j\le l} s_j t_j^\omega$ *for some* $l \ge 0$, *where* $s_j, t_j \in S^{alg}\langle\langle\Sigma^*\rangle\rangle$ *with* $(t_j, \epsilon) = 0$; *moreover* $(s_j, \epsilon) = 0$ *or* $s_j = (s_j, \epsilon)\epsilon$.

Proof. Assume $(s_j, \epsilon) \neq 0$. Then $s_j = (s_j, \epsilon)\epsilon + s_j'$ where $(s_j', \epsilon) = 0$, and $s_j t_j^\omega = (s_j, \epsilon)t_j^\omega + s_j' t_j^\omega$.

Assume $(t_j, \epsilon) \neq 0$. Then $t_j = (t_j, \epsilon)\epsilon + t_j'$, where $(t_j', \epsilon) = 0$. Since $(S\langle\langle\Sigma^*\rangle\rangle, S\langle\langle\Sigma^\omega\rangle\rangle)$ is a Conway semiring-semimodule pair satisfying $\epsilon^\omega = 0$, we obtain $t_j^\omega = ((t_j, \epsilon)^*\epsilon^* t_j')^\omega$ with $(t_j, \epsilon)^*\epsilon^* t_j' \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$, since $((t_j, \epsilon)\epsilon)^\omega = (t_j, \epsilon)^\omega\epsilon^\omega = 0$. □

We now assume that $(s, v) \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ is given in the form of Corollary 3.6 with $l = 1$. By Theorem 2.4.10 of Ésik and Kuich (2007a), there exist algebraic systems in Greibach normal form whose first component of their least solutions equals s_1, t_1.

Firstly, we deal with the case $(s_1, \epsilon) = 0$. Let

$$x_i = p_i(x) + \sum_{1\le j\le n} p_{ij}(x)x_j, \quad \text{for each } 1 \le i \le n, \tag{$*$}$$

where $\mathrm{supp}(p_i(x)) \subseteq \Sigma \cup \Sigma X$, $\mathrm{supp}(p_{ij}(x)) \subseteq \Sigma X$, be the algebraic system in Greibach normal form for s_1 and

$$x_i' = p_i'(x') + \sum_{1\le j\le m} p_{ij}'(x')x_j', \quad \text{for each } 1 \le i \le m, \tag{$**$}$$

where $\mathrm{supp}(p_i'(x')) \subseteq \Sigma \cup \Sigma X'$, $\mathrm{supp}(p_{ij}(x')) \subseteq \Sigma X'$, be the algebraic system in Greibach normal form for t_1. Let σ and σ' with $\sigma_1 = s_1$ and $\sigma_1' = t_1$ be the least solutions of $(*)$ and $(**)$, respectively.

Consider now the mixed ω-algebraic system consisting of the algebraic system $(*)$, $(**)$ over $S\langle\langle\Sigma^*\rangle\rangle$ and the linear system over $S\langle\langle\Sigma^\omega\rangle\rangle$

$$\begin{aligned} z'' &= p_1'(x')z'' + \sum_{1\le j\le m} p_{1j}'(x')z_j', \\ z_i' &= p_i'(x')z'' + \sum_{1\le j\le m} p_{ij}'(x')z_j', \quad \text{for } 1 \le i \le m\,, \\ z_i &= p_i(x)z'' + \sum_{1\le j\le n} p_{ij}(x)z_j, \quad \text{for } 1 \le i \le n\,. \end{aligned} \tag{$***$}$$

Observe that the mixed ω-algebraic system is in Greibach normal form. We then order the variables of the mixed ω-algebraic system $(*)$, $(**)$, $(***)$ as $x_1, \ldots, x_n; x_1', \ldots, x_m'; z''; z_1', \ldots, z_m'; z_1, \ldots, z_n$. After an example, we will prove that

$$(\sigma_1, \ldots, \sigma_n; \sigma_1', \ldots, \sigma_m'; \sigma_1'\sigma_1'^\omega; \sigma_1'\sigma_1'^\omega, \ldots, \sigma_m'\sigma_1'^\omega; \sigma_1\sigma_1'^\omega, \ldots, \sigma_n\sigma_1'^\omega) \tag{3.7}$$

is a solution of $(*)$, $(**)$, $(***)$. Observe that $\sigma_1'\sigma_1'^\omega = \sigma_1'^\omega$.

Example 3.7. Consider the quemiring $\mathbb{N}^\infty\langle\langle\Sigma^*\rangle\rangle \times \mathbb{N}^\infty\langle\langle\Sigma^\omega\rangle\rangle$ for the tropical semiring $\langle\mathbb{N}^\infty, \min, +, \infty, 0\rangle$. Note that subsequently, 1 stands for the natural number 1 and the neutral element of the semiring multiplication is $\mathbb{1} = 0$.

We now define algebraic systems in Greibach normal form for $s = a^n b^n \mapsto n$ and $t = ((dd)^* c) \mapsto 0$. Let

$$\begin{aligned} x_1 &= 1ax_2 + 1ax_1x_2 & x_1' &= c + dx_2'x_1' \\ x_2 &= b & x_2' &= d \end{aligned}$$

Here, x_1 is the start variable for s and x_1' is the start variable for t. In the proof, these two systems are called $(*)$ and $(**)$. Now, we construct a mixed ω-algebraic system:

$$\begin{aligned} z'' &= cz'' + dx_2'z_1' & & \\ z_1' &= cz'' + dx_2'z_1' & z_2' &= dz'' \\ z_1 &= 1ax_2z'' + 1ax_1z_2 & z_2 &= bz'' \end{aligned}$$

In the new system (corresponding to $(***)$), variable z'' is Büchi-accepting and variable z_1 acts as the start variable, i.e., we consider the fourth component (with the ordering $z'', z_1', z_2', z_1, z_2$) of the first canonical solution. The semimodule part of the solution is $st^\omega = a^n b^n ((dd)^* c)^\omega \mapsto n$. Note that the equation for z'' is needed in this example because z_1' is not allowed to be Büchi-accepting to prevent $(dd)^\omega$ as part of the canonical solution. ▽

Lemma 3.8. *The mixed ω-algebraic system $(*)$, $(**)$, $(***)$ has solution*

$$(\sigma_1, \ldots, \sigma_n; \sigma_1', \ldots, \sigma_m'; \sigma_1'\sigma_1'^\omega; \sigma_1'\sigma_1'^\omega, \ldots, \sigma_m'\sigma_1'^\omega; \sigma_1\sigma_1'^\omega, \ldots, \sigma_n\sigma_1'^\omega)$$

Proof. Again, observe that $\sigma_1'^\omega = \sigma_1'\sigma_1'^\omega$. We obtain, for the first equation,

$$\begin{aligned} p_1'(\sigma')\sigma_1'^\omega + \sum_{1\le j\le m} p_{1j}'(\sigma')\sigma_j'\sigma_1'^\omega &= \big(p_1'(\sigma') + \sum_{1\le j\le m} p_{1j}'(\sigma')\sigma_j'\big)\sigma_1'^\omega \\ &= \sigma_1'\sigma_1'^\omega\,, \end{aligned}$$

then, for $1 \le i \le m$, and the second equation,

$$\begin{aligned} p_i'(\sigma')\sigma_1'^\omega + \sum_{1\le j\le m} p_{ij}'(\sigma')\sigma_j'\sigma_1'^\omega &= \big(p_i'(\sigma') + \sum_{1\le j\le m} p_{ij}'(\sigma')\sigma_j'\big)\sigma_1'^\omega \\ &= \sigma_i'\sigma_1'^\omega\,, \end{aligned}$$

and, for $1 \le i \le n$, and the third equation,

$$\begin{aligned} p_i(\sigma)\sigma_1'^\omega + \sum_{1\le j\le n} p_{ij}(\sigma)\sigma_j\sigma_1'^\omega &= \big(p_i(\sigma) + \sum_{1\le j\le n} p_{ij}(\sigma)\sigma_j\big)\sigma_1'^\omega \\ &= \sigma_i\sigma'^\omega\,. \end{aligned}$$

□

But we need more: We will prove that this solution is the first canonical solution of $(*)$, $(**)$, $(***)$.

Lemma 3.9. *The solution (3.7) is the first canonical solution of the mixed ω-algebraic system $(*)$, $(**)$, $(***)$.*

Proof. Let

$$P'_{1m}(x') = \left(p'_{11}(x') \cdots p'_{1m}(x')\right),$$

$$P'_{m1}(x') = \begin{pmatrix} p'_1(x') \\ \vdots \\ p'_m(x') \end{pmatrix}, \qquad P'_{mm}(x') = \begin{pmatrix} p'_{11}(x') \ \ldots \ p'_{1m}(x') \\ \vdots \qquad\qquad \vdots \\ p'_{m1}(x') \ \ldots \ p'_{mm}(x') \end{pmatrix},$$

$$P_{n1}(x) = \begin{pmatrix} p_1(x) \\ \vdots \\ p_n(x) \end{pmatrix}, \qquad P_{nn}(x) = \begin{pmatrix} p_{11}(x) \ \ldots \ p_{1n}(x) \\ \vdots \qquad\qquad \vdots \\ p_{n1}(x) \ \ldots \ p_{nn}(x) \end{pmatrix},$$

$$z = \begin{pmatrix} z_1 \\ \vdots \\ z_n \end{pmatrix}, \qquad z' = \begin{pmatrix} z'_1 \\ \vdots \\ z'_m \end{pmatrix},$$

and

$$M(x, x') = \begin{pmatrix} p'_1(x') & P'_{1m}(x') & 0 \\ P'_{m1}(x') & P'_{mm}(x') & 0 \\ P_{n1}(x) & 0 & P_{nn}(x) \end{pmatrix}.$$

Then the linear system $(***)$ can be written in the form

$$\begin{pmatrix} z'' \\ z' \\ z \end{pmatrix} = M(x, x') \begin{pmatrix} z'' \\ z' \\ z \end{pmatrix}.$$

Hence, the first canonical solution of $(*)$, $(**)$, $(***)$ is $(\sigma, \sigma', M(\sigma, \sigma')^{\omega,1})$. Before we prove our lemma, we prove three identities.

The system $(*)$ can be written in the form

$$x = P_{n1}(x) + P_{nn}(x)x, \quad \text{for } x = (x_1, \ldots, x_n)^{\mathsf{T}}.$$

By the diagonal identity (see Proposition 2.2.11 of Ésik and Kuich (2007a)) the system

$$x = P_{n1}(\sigma) + P_{nn}(\sigma)x$$

has the same least solution as $(*)$. Hence,

$$\sigma = P_{nn}(\sigma)^* P_{n1}(\sigma). \tag{3.8}$$

The system $(**)$ can be written in the form

$$x' = P'_{m1}(x') + P'_{mm}(x')x', \quad \text{for } x' = (x'_1, \ldots, x'_m)^{\mathsf{T}}.$$

Again, by the diagonal identity (see Proposition 2.2.11 of Ésik and Kuich, 2007a) the system

$$x' = P'_{m1}(\sigma') + P'_{mm}(\sigma')x'$$

has the same solution. Hence

$$\sigma' = P'_{mm}(\sigma')^* P'_{m1}(\sigma')\,. \tag{3.9}$$

It follows for the first component

$$\begin{aligned}
\sigma'_1 &= \left(P'_{mm}(\sigma')^* P'_{m1}(\sigma')\right)_1 \\
&= \left(P'_{m1}(\sigma') + P'_{mm}(\sigma')^+ P'_{m1}(\sigma')\right)_1 \\
&= \left(P'_{m1}(\sigma') + P'_{mm}(\sigma') P'_{mm}(\sigma')^* P'_{m1}(\sigma')\right)_1 \\
&= p'_1(\sigma') + P'_{1m}(\sigma') P'_{mm}(\sigma')^* P'_{m1}(\sigma')\,.
\end{aligned} \tag{3.10}$$

We now compute

$$\begin{aligned}
(M^{\omega,1}(\sigma,\sigma'))_{z''} &= \left[p'_1(\sigma') + \begin{pmatrix} P'_{1m}(\sigma') & 0 \end{pmatrix} \begin{pmatrix} P'_{mm}(\sigma') & 0 \\ 0 & P_{nn}(\sigma) \end{pmatrix}^* \begin{pmatrix} P'_{m1}(\sigma') \\ P_{n1}(\sigma) \end{pmatrix}\right]^\omega \\
&= \left[p'_1(\sigma') + \begin{pmatrix} P'_{1m}(\sigma') & 0 \end{pmatrix} \begin{pmatrix} P'_{mm}(\sigma')^* & 0 \\ 0 & P_{nn}(\sigma)^* \end{pmatrix} \begin{pmatrix} P'_{m1}(\sigma') \\ P_{n1}(\sigma) \end{pmatrix}\right]^\omega \\
&= \left[p'_1(\sigma') + P'_{1m}(\sigma') P'_{mm}(\sigma')^* P'_{m1}(\sigma')\right]^\omega \\
&= \sigma_1'^\omega\,.
\end{aligned}$$

The last equality is by (3.10).

When starting with another variable z_i or z'_j for $1 \le i \le n$ and $1 \le j \le m$, we get

$$\begin{aligned}
(M^{\omega,1}(\sigma,\sigma'))_{(z',z)} &= \begin{pmatrix} P'_{mm}(\sigma') & 0 \\ 0 & P_{nn}(\sigma) \end{pmatrix}^* \begin{pmatrix} P'_{m1}(\sigma') \\ P_{n1}(\sigma) \end{pmatrix} (M^{\omega,1}(\sigma,\sigma'))_{z''} \\
&= \begin{pmatrix} P'_{mm}(\sigma')^* & 0 \\ 0 & P_{nn}(\sigma)^* \end{pmatrix} \begin{pmatrix} P'_{m1}(\sigma') \\ P_{n1}(\sigma) \end{pmatrix} \sigma_1'^\omega \\
&= \begin{pmatrix} P'_{mm}(\sigma')^* P'_{m1}(\sigma') \\ P_{nn}(\sigma)^* P_{n1}(\sigma) \end{pmatrix} \sigma_1'^\omega
\end{aligned}$$

Thus, by (3.9), we have, for $1 \le i \le m$,

$$(M^{\omega,1}(\sigma,\sigma'))_{z'_i} = \left[P'_{mm}(\sigma')^* P'_{m1}(\sigma')\right]_i \sigma_1'^\omega = \sigma'_i \sigma_1'^\omega\,,$$

and, by (3.8), we have, for $1 \le i \le n$,

$$(M^{\omega,1}(\sigma,\sigma'))_{z_i} = \left[P_{nn}(\sigma)^* P_{n1}(\sigma)\right]_i \sigma_1'^\omega = \sigma_i \sigma_1'^\omega\,.$$

This completes the proof. □

Secondly, we deal with the case $s_1 = (s_1, \epsilon)\epsilon$. Consider now the mixed ω-algebraic system consisting of $(**)$ and the linear system over $S\langle\langle\Sigma^\omega\rangle\rangle$

$$\begin{aligned} z'' &= p'_1(x')z'' + \sum_{1\le j\le m} p'_{1j}(x')z'_j\,, \\ z'_i &= p'_i(x')z'' + \sum_{1\le j\le m} p'_{ij}(x')z'_j,\ 1 \le i \le m\,, \\ z_1 &= (s_1,\epsilon)p'_1(x')z'' + (s_1,\epsilon)\sum_{1\le j\le m} p'_{1j}(x')z'_j\,. \end{aligned} \tag{$****$}$$

Lemma 3.10. *The mixed ω-algebraic system $(**)$, $(****)$ has solution*

$$(\sigma'_1, \ldots, \sigma'_m; \sigma'_1\sigma_1'^{\omega}; \sigma'_1\sigma_1'^{\omega}, \ldots, \sigma'_m\sigma_1'^{\omega}; (s_1,\epsilon)\sigma_1'^{\omega})\,. \tag{3.11}$$

Proof. As in the proof of Lemma 3.8, we obtain that $(\sigma'_1\sigma_1'^{\omega}; \sigma'_1\sigma_1'^{\omega}, \ldots, \sigma'_m\sigma_1'^{\omega})$ is solution of the z''- and z'_i-equations, $1 \le i \le m$. For the right side of the z_1-equation, we obtain

$$(s_1,\epsilon)\Big(p'_1(\sigma')\sigma_1'^{\omega} + \sum_{1\le j\le m} p'_{1j}(\sigma')\sigma'_j\sigma_1'^{\omega}\Big) = (s_1,\epsilon)\sigma'_1\sigma_1'^{\omega} = (s_1,\epsilon)\sigma_1'^{\omega}\,. \qquad \square$$

But again we need more: We will prove that this solution is the first canonical solution of $(**)$, $(****)$.

Lemma 3.11. *The solution (3.11) is the first canonical solution of the mixed algebraic system $(**)$, $(****)$.*

Proof. Let

$$M_\epsilon(x') = \begin{pmatrix} p'_1(x') & P'_{1m}(x') & 0 \\ P'_{m1}(x') & P'_{mm}(x') & 0 \\ (s_1,\epsilon)p'_1(x') & (s_1,\epsilon)P'_{1m}(x') & 0 \end{pmatrix}.$$

Then the linear system $(****)$ can be written in the form

$$\begin{pmatrix} z'' \\ z' \\ z_1 \end{pmatrix} = M_\epsilon(x') \begin{pmatrix} z'' \\ z' \\ z_1 \end{pmatrix}.$$

Hence, the first canonical solution of $(**)$, $(****)$ is $(\sigma', M_\epsilon(\sigma')^{\omega,1})$. We now compute

$$\begin{aligned} &(M_\epsilon^{\omega,1}(\sigma'))_{z''} \\ &= \left[p'_1(\sigma') + \begin{pmatrix} P'_{1m}(\sigma') & 0 \end{pmatrix} \begin{pmatrix} P'_{mm}(\sigma') & 0 \\ (s_1,\epsilon)P'_{1m}(\sigma') & 0 \end{pmatrix}^* \begin{pmatrix} P'_{m1}(\sigma') \\ (s_1,\epsilon)p'_1(\sigma') \end{pmatrix}\right]^\omega \\ &= \left[p'_1(\sigma') + \begin{pmatrix} P'_{1m}(\sigma') & 0 \end{pmatrix} \begin{pmatrix} P'_{mm}(\sigma')^* & 0 \\ (s_1,\epsilon)P'_{1m}(\sigma')P'_{mm}(\sigma')^* & 1 \end{pmatrix} \begin{pmatrix} P'_{m1}(\sigma') \\ (s_1,\epsilon)p'_1(\sigma') \end{pmatrix}\right]^\omega \\ &= \left[p'_1(\sigma') + P'_{1m}(\sigma')P'_{mm}(\sigma')^*P'_{m1}(\sigma')\right]^\omega \\ &= \sigma_1'^{\omega}\,. \end{aligned}$$

The last equality is by (3.10).

When starting with another variable z_i' or z_1 for $1 \leq i \leq m$, we get

$$\begin{aligned}(M_\epsilon^{\omega,1}(\sigma'))_{(z',z_1)} &= \begin{pmatrix} P'_{mm}(\sigma') & 0 \\ (s_1,\epsilon)P'_{1m}(\sigma') & 0 \end{pmatrix}^* \begin{pmatrix} P'_{m1}(\sigma') \\ (s_1,\epsilon)p'_1(\sigma') \end{pmatrix} (M_\epsilon^{\omega,1}(\sigma'))_{z''} \\ &= \begin{pmatrix} P'_{mm}(\sigma')^* & 0 \\ (s_1,\epsilon)P'_{1m}(\sigma')P'_{mm}(\sigma')^* & 1 \end{pmatrix} \begin{pmatrix} P'_{m1}(\sigma') \\ (s_1,\epsilon)p'_1(\sigma') \end{pmatrix} \sigma_1'^{\omega} \\ &= \begin{pmatrix} P'_{mm}(\sigma')^* P'_{m1}(\sigma') \\ (s_1,\epsilon)P'_{1m}(\sigma')P'_{mm}(\sigma')^* P'_{m1}(\sigma') + (s_1,\epsilon)p'_1(\sigma') \end{pmatrix} \sigma_1'^{\omega}\end{aligned}$$

Thus, by (3.9), we have, for $1 \leq i \leq m$,

$$(M_\epsilon^{\omega,1}(\sigma'))_{z_i'} = \left[P'_{mm}(\sigma')^* P'_{m1}(\sigma')\right]_i \sigma_1'^{\omega} = \sigma_i' \sigma_1'^{\omega},$$

and, by (3.10), we have

$$\begin{aligned}(M_\epsilon^{\omega,1}(\sigma'))_{z_1} &= \big((s_1,\epsilon)P'_{1m}(\sigma')P'_{mm}(\sigma')^* P'_{m1}(\sigma') + (s_1,\epsilon)p'_1(\sigma')\big)\sigma_1'^{\omega} \\ &= (s_1,\epsilon)\big(P'_{1m}(\sigma')P'_{mm}(\sigma')^* P'_{m1}(\sigma') + p'_1(\sigma')\big)\sigma_1'^{\omega} \\ &= (s_1,\epsilon)\sigma_1'\sigma_1'^{\omega} = (s_1,\epsilon)\sigma_1'^{\omega}.\end{aligned}$$

□

We now consider general sums of series of the above form. The next lemma shows how to construct a mixed ω-algebraic system whose canonical solution is the sum of the canonical solutions of multiple mixed ω-algebraic systems as given in Lemmas 3.9 and 3.11.

Lemma 3.12. *Let $(s,v) \in S^{alg}\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ be given in the form of Corollary 3.6. Then there exists a mixed ω-algebraic system in Greibach normal form such that v is a component of its l^{th} canonical solution.*

Proof. Let $v = \sum_{1\leq i\leq l} s_i t_i^\omega$ as in the statement of Corollary 3.6 and let $l \geq 1$. By Lemmas 3.9 and 3.11, for $1 \leq i \leq l$, there exist mixed ω-algebraic systems

$$\begin{aligned} x_i &= p_i(x_i), \\ \begin{pmatrix} z_i \\ \bar{z}_i \end{pmatrix} &= M_i(x_i) \begin{pmatrix} z_i \\ \bar{z}_i \end{pmatrix}, \end{aligned} \qquad (\sharp)$$

in Greibach normal form with

$$M_i(x_i) = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix},$$

where

$$\begin{aligned} a_i &\in (S\langle(\Sigma \cup X)^*\rangle)^{1\times 1}, \\ b_i &\in (S\langle(\Sigma \cup X)^*\rangle)^{1\times (n_i-1)}, \\ c_i &\in (S\langle(\Sigma \cup X)^*\rangle)^{(n_i-1)\times 1}, \\ d_i &\in (S\langle(\Sigma \cup X)^*\rangle)^{(n_i-1)\times(n_i-1)}, \end{aligned}$$

such that $s_i t_i^\omega$ is a component of the first canonical solution of the i^{th} system. We will assume without loss of generality that $s_i t_i^\omega$ is the first component of variable $\bar{z}_i$, i.e.,

$$s_i t_i^\omega = \left[(M_i^{\omega,1})_{\bar{z}_i} \right]_1 = \left[(d_i^* c_i)(a_i + b_i d_i^* c_i)^\omega \right]_1 . \tag{3.12}$$

Similarly to the case of summation in Theorem 5.4.4 of Ésik and Kuich (2007a), we consider now the mixed ω-algebraic system consisting of the algebraic systems ($\sharp$) over $S\langle\langle \Sigma^* \rangle\rangle$ and the linear system over $S\langle\langle \Sigma^\omega \rangle\rangle$

$$\hat{z} = M\hat{z} , \tag{$\sharp\sharp$}$$

with

$$M = \begin{pmatrix} \begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_l \end{pmatrix} & \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_l \end{pmatrix} & 0 \\ \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} & \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix} & 0 \\ (c_1 \cdots c_l) & (d_1 \cdots d_l) & 0 \end{pmatrix}, \quad \hat{z} = \begin{pmatrix} z_1 \\ \vdots \\ z_l \\ \bar{z}_1 \\ \vdots \\ \bar{z}_l \\ z' \end{pmatrix} .$$

Note that this system $\sharp\sharp$ is still in Greibach normal form.

We order the variables of the mixed ω-algebraic system ($\sharp$), ($\sharp\sharp$) as $z_1, \ldots, z_l; \bar{z}_1, \ldots, \bar{z}_l; z'$. We now compute the l^{th} canonical solution, starting with variable $z = (z_1, \ldots, z_l)^{\mathsf{T}}$. Then

$$\begin{aligned}
&(M^{\omega,l})_z \\
&= \left[\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_l \end{pmatrix} + \begin{pmatrix} \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_l \end{pmatrix} & 0 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix} & 0 \\ (d_1 \cdots d_l) & 0 \end{pmatrix}^* \begin{pmatrix} \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \\ (c_1 \cdots c_l) \end{pmatrix} \right]^\omega \\
&= \left[\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_l \end{pmatrix} + \begin{pmatrix} \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_l \end{pmatrix} & 0 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* & 0 \\ (d_1 \cdots d_l) \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* & 1 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \\ (c_1 \cdots c_l) \end{pmatrix} \right]^\omega \\
&= \left[\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_l \end{pmatrix} + \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_l \end{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \right]^\omega \\
&= \begin{pmatrix} (a_1 + b_1 d_1^* c_1)^\omega \\ \vdots \\ (a_l + b_l d_l^* c_l)^\omega \end{pmatrix} .
\end{aligned}$$

When starting with the new variable z', we get a sum of the original solutions:

$$
\begin{aligned}
(M^{\omega,l})_{z'} &= \left[\begin{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix} & 0 \\ (d_1 \cdots d_l) & 0 \end{pmatrix}^* \begin{pmatrix} \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \\ (c_1 \cdots c_l) \end{pmatrix} (M^{\omega,l})_z \right]_{l+1} \\
&= \left[\begin{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* & 0 \\ (d_1 \cdots d_l) \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* & 1 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \\ (c_1 \cdots c_l) \end{pmatrix} (M^{\omega,l})_z \right]_{l+1} \\
&= \left[\begin{pmatrix} \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_l \end{pmatrix}^* \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} \\ (d_1 d_1^* \cdots d_l d_l^*) \begin{pmatrix} c_1 & & \\ & \ddots & \\ & & c_l \end{pmatrix} + (c_1 \cdots c_l) \end{pmatrix} (M^{\omega,l})_z \right]_{l+1} \\
&= \left[\begin{pmatrix} \begin{pmatrix} d_1^* c_1 & & \\ & \ddots & \\ & & d_l^* c_l \end{pmatrix} \\ (d_1 d_1^* c_1 + c_1 \cdots d_l d_l^* c_l + c_l) \end{pmatrix} \begin{pmatrix} (a_1 + b_1 d_1^* c_1)^\omega \\ \vdots \\ (a_l + b_l d_l^* c_l)^\omega \end{pmatrix} \right]_{l+1} \\
&= \left[\begin{pmatrix} d_1^* c_1 (a_1 + b_1 d_1^* c_1)^\omega \\ \vdots \\ d_l^* c_l (a_l + b_l d_l^* c_l)^\omega \\ \sum_{1 \le i \le l} (d_i d_i^* c_i + c_i)(a_i + b_i d_i^* c_i)^\omega \end{pmatrix} \right]_{l+1} \\
&= \sum_{1 \le i \le l} (d_i d_i^* c_i + c_i)(a_i + b_i d_i^* c_i)^\omega \\
&= \sum_{1 \le i \le l} (d_i^* c_i)(a_i + b_i d_i^* c_i)^\omega
\end{aligned}
$$

Thus, the first component is (by identity (3.12))

$$
\begin{aligned}
\left[(M^{\omega,l})_{z'} \right]_1 &= \left[\sum_{1 \le i \le l} (d_i^* c_i)(a_i + b_i d_i^* c_i)^\omega \right]_1 \\
&= \sum_{1 \le i \le l} \left[(d_i^* c_i)(a_i + b_i d_i^* c_i)^\omega \right]_1 = \sum_{1 \le i \le l} s_i t_i^\omega = v \,. \qquad \square
\end{aligned}
$$

We can now conclude the following theorem.

Theorem 3.13. *The following statement for $(s, v) \in S\langle\langle \Sigma^* \rangle\rangle \times S\langle\langle \Sigma^\omega \rangle\rangle$ is equivalent to the statements of Theorem 3.5:*

(s, v) is component of a canonical solution of a mixed ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ in Greibach normal form.

Proof. The above statement trivially implies statement (v) of Theorem 3.5. By Corollary 3.6 and Lemma 3.12, the statements of Theorem 3.5 imply the above statement. □

3.4 Greibach Normal Form for ω-Algebraic Systems

For the following chapter, we need the Greibach normal form not only for mixed ω-algebraic systems but also for ω-algebraic systems. So we show in this section a specialization of Theorem 3.13 for ω-algebraic systems.

Similar to the definition for mixed ω-algebraic systems, an ω-algebraic system

$$y = p(y)$$

where $\{y_1, \ldots, y_n\}$ is a set of variables for the quemiring $S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\mathrm{alg}}\langle\langle\Sigma^\omega\rangle\rangle$, is in *Greibach normal form* if

$$\mathrm{supp}(p_i(y)) \subseteq \{\epsilon\} \cup \Sigma \cup \Sigma Y \cup \Sigma Y Y, \qquad \text{for all } 1 \leq i \leq n\,.$$

The main result of this chapter is the following.

Theorem 3.14. *The following statement for $(s, v) \in S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ is equivalent to the statements of Theorem 3.5:*
(s, v) is component of a canonical solution of an ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ in Greibach normal form.

Proof. By Theorem 3.13, we can assume that (s, v) is component of the t^{th} canonical solution of a mixed ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ in Greibach normal form for a $t \in \mathbb{N}$. Let the mixed ω-algebraic system be given in the following form:

$$x_i = p_i + \sum_{1 \leq j \leq n} (p_{ij}x + q_{ij})x_j, \quad \text{for } 1 \leq i \leq n\,, \tag{$*$}$$

$$z_i = \sum_{1 \leq j \leq m} (p'_{ij}x + q'_{ij})z_j, \qquad \text{for } 1 \leq i \leq m\,, \tag{$**$}$$

where

$$p_{ij} \in S\langle\Sigma\rangle^{1\times n}, \qquad \text{for } 1 \leq i, j \leq n\,,$$
$$p'_{ij} \in S\langle\Sigma\rangle^{1\times n}, \qquad \text{for } 1 \leq i, j \leq m\,,$$

and

$$\mathrm{supp}(p_i) \subseteq \{\epsilon\} \cup \Sigma, \quad \mathrm{supp}(p_{ij}x) \subseteq \Sigma X, \quad \mathrm{supp}(q_{ij}) \subseteq \Sigma\,,$$
$$\mathrm{supp}(p'_{ij}x) \subseteq \Sigma X, \quad \mathrm{supp}(q'_{ij}) \subseteq \Sigma\,.$$

Note that

$$p_{ij}x = \sum_{1\leq k\leq n} (p_{ij})_k x_k\,;$$

we decided for this notation because of brevity, important especially in matrices.

For the remainder of the proof, consider integers k and l to be fixed such that the t^{th} canonical solution of $(\divideontimes)$, $(\divideontimes\divideontimes)$ is (σ, ω) with $\sigma_k = s$ and $\omega_l = v$.

We will later need a simple implication: We can write the linear system $(\divideontimes\divideontimes)$ as

$$z = P'_{mm}(x)z\,,$$

where

$$P'_{mm}(x) = \begin{pmatrix} p'_{11}x + q'_{11} & \cdots & p'_{1m}x + q'_{1m} \\ \vdots & \ddots & \vdots \\ p'_{m1}x + q'_{m1} & \cdots & p'_{mm}x + q'_{mm} \end{pmatrix}.$$

Note that $t \leq m$. It follows that

$$\omega = P'_{mm}(\sigma)^{\omega,t}. \tag{3.13}$$

Now, we construct from $(\divideontimes)$, $(\divideontimes\divideontimes)$ an ω-algebraic system $(\divideontimes\divideontimes\divideontimes)$ where the variables x are substituted by $\bar{y}$ and z by $\hat{y}$. Additionally, we add a new equation and a new variable $\dot{y}$ to combine the k^{th} component of the semiring part and the l^{th} component of the semimodule part:

$$\begin{aligned} \hat{y}_i &= \sum_{1\leq j\leq m} (p'_{ij}\bar{y} + q'_{ij})\hat{y}_j, \qquad \text{for } 1 \leq i \leq m\,, \\ \bar{y}_i &= p_i + \sum_{1\leq j\leq n} (p_{ij}\bar{y} + q_{ij})\bar{y}_j, \qquad \text{for } 1 \leq i \leq n\,, \\ \dot{y} &= p_k + \sum_{1\leq j\leq n} (p_{kj}\bar{y} + q_{kj})\bar{y}_j + \sum_{1\leq j\leq m} (p'_{lj}\bar{y} + q'_{lj})\hat{y}_j\,. \end{aligned} \tag{$\divideontimes\divideontimes\divideontimes$}$$

Note that $(\divideontimes\divideontimes\divideontimes)$ is in Greibach normal form. Moreover, note that we order the equations such that the first equations are those corresponding to the old equations of variables z_i. This ensures that the t^{th} canonical solution still considers the correct variables as Büchi-accepting.

Claim: The $(m+n+1)^{\text{th}}$ component of the t^{th} canonical solution of $(\divideontimes\divideontimes\divideontimes)$ is $(\sigma_k, \omega_l) = (s, v)$.

We now compute this solution. The t^{th} canonical solution of the ω-algebraic system $(\divideontimes\divideontimes\divideontimes)$ is defined to be the t^{th} canonical solution of the mixed ω-algebraic system induced by $(\divideontimes\divideontimes\divideontimes)$. The corresponding induced mixed ω-algebraic system is given by the algebraic system over $S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle$

$$\begin{aligned} \hat{x}_i &= \sum_{1\leq j\leq m} (p'_{ij}\bar{x} + q'_{ij})\hat{x}_j, \qquad \text{for } 1 \leq i \leq m\,, \\ \bar{x}_i &= p_i + \sum_{1\leq j\leq n} (p_{ij}\bar{x} + q_{ij})\bar{x}_j, \qquad \text{for } 1 \leq i \leq n\,, \\ \dot{x} &= p_k + \sum_{1\leq j\leq n} (p_{kj}\bar{x} + q_{kj})\bar{x}_j + \sum_{1\leq j\leq m} (p'_{lj}\bar{x} + q'_{lj})\hat{x}_j\,, \end{aligned} \tag{\#}$$

and the linear system over $S^{\mathrm{alg}}\langle\langle\Sigma^\omega\rangle\rangle$

$$\begin{aligned}
\hat{z}_i &= \sum_{1\le j\le m} (p'_{ij}\bar{x} + q'_{ij})\hat{z}_j + p'_{ij}\bar{z}, \qquad \text{for } 1 \le i \le m\,,\\
\bar{z}_i &= \sum_{1\le j\le n} (p_{ij}\bar{x} + q_{ij})\bar{z}_j + p_{ij}\bar{z}, \qquad \text{for } 1 \le i \le n\,,\\
\dot{z} &= \sum_{1\le j\le n} (p_{kj}\bar{x} + q_{kj})\bar{z}_j + p_{kj}\bar{z} + \sum_{1\le j\le m} (p'_{lj}\bar{x} + q'_{lj})\hat{z}_j + p'_{lj}\bar{z}\,.
\end{aligned} \tag{\#\#}$$

Claim: $(0,\dots,0;\sigma;\sigma_k)$ is the least solution of (#).

First, we prove that it is a solution by plugging it into the right sides of the equations. We have for the first m equations, and for $1 \le i \le m$,

$$\sum_{1\le j\le m} (p'_{ij}\sigma + q'_{ij})0 = 0\,.$$

Then for the second set of equations and $1 \le i \le n$,

$$p_i + \sum_{1\le j\le n} (p_{ij}\sigma + q_{ij})\sigma_j = \sigma_i\,;$$

because σ is a solution of $(\divideontimes)$. Finally, we obtain by the same reason, for the last equation,

$$\begin{aligned}
p_k + \sum_{1\le j\le n} (p_{kj}\sigma + q_{kj})\sigma_j + \sum_{1\le j\le m} (p'_{lj}\sigma + q'_{lj})0_j &= p_k + \sum_{1\le j\le n} (p_{kj}\sigma + q_{kj})\sigma_j + 0\\
&= \sigma_k\,.
\end{aligned}$$

The algebraic system (#) is strict and therefore has a unique solution. This means that $(0,\dots,0;\sigma;\sigma_k)$ is also the least solution. This proves the claim.

Now consider the linear system (##). Let $P'_{mm}(\bar{x})$ be defined as above and let further

$$P_{nn}(\bar{x}) = \begin{pmatrix} p_{11}\bar{x} + q_{11} & \cdots & p_{1n}\bar{x} + q_{1n} \\ \vdots & \ddots & \vdots \\ p_{n1}\bar{x} + q_{n1} & \cdots & p_{nn}\bar{x} + q_{nn} \end{pmatrix},$$

$$R_{nn} = \begin{pmatrix} \sum_{1\le j\le n}(p_{1j})_1 & \cdots & \sum_{1\le j\le n}(p_{1j})_n \\ \vdots & \ddots & \vdots \\ \sum_{1\le j\le n}(p_{nj})_1 & \cdots & \sum_{1\le j\le n}(p_{nj})_n \end{pmatrix},$$

$$R'_{mn} = \begin{pmatrix} \sum_{1\le j\le m}(p'_{1j})_1 & \cdots & \sum_{1\le j\le m}(p'_{1j})_n \\ \vdots & \ddots & \vdots \\ \sum_{1\le j\le m}(p'_{mj})_1 & \cdots & \sum_{1\le j\le m}(p'_{mj})_n \end{pmatrix}.$$

Note that for (##) and for $1 \leq i \leq m$, we have

$$\begin{aligned}\sum_{1\leq j\leq m} p'_{ij}\bar{z} &= \sum_{1\leq j\leq m}\sum_{1\leq k\leq n}(p'_{ij})_k\bar{z}_k \\ &= \sum_{1\leq k\leq n}\sum_{1\leq j\leq m}(p'_{ij})_k\bar{z}_k \\ &= \Big(\sum_{1\leq j\leq m}(p'_{ij})_1, \cdots, \sum_{1\leq j\leq m}(p'_{ij})_n\Big)\bar{z} \\ &= (R'_{mn})_i\bar{z}\,.\end{aligned}$$

Analogously, we can prove $\sum_{1\leq j\leq n} p_{ij}\bar{z} = (R_{nn})_i\bar{z}$. We let

$$M(\hat{x},\bar{x},x) = \begin{pmatrix} P'_{mm}(\bar{x}) & R'_{mn} & 0 \\ 0 & P_{nn}(\bar{x}) + R_{nn} & 0 \\ (P'_{mm}(\bar{x}))_l & (P_{nn}(\bar{x}))_k + (R_{nn})_k + (R'_{mn})_l & 0 \end{pmatrix},$$

then the linear system (##) can be written as

$$\begin{pmatrix}\hat{z}\\ \bar{z}\\ z\end{pmatrix} = M(\hat{x},\bar{x},x)\begin{pmatrix}\hat{z}\\ \bar{z}\\ z\end{pmatrix}.$$

Now, we can plug the semiring part $(0,\sigma,\sigma_k)$ of the solution into M. By Theorem 3.2, the semimodule part of the canonical solution of (#), (##) is

$$M(0,\sigma,\sigma_k)^{\omega,t} = \begin{pmatrix} & & \xi^{\omega,t} \\ \begin{pmatrix} P_{nn}(\sigma)+R_{nn} & 0 \\ \chi & 0\end{pmatrix}^* & \begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l\end{pmatrix} & \xi^{\omega,t}\end{pmatrix}$$

with

$$\chi = (P_{nn}(\sigma))_k + (R_{nn})_k + (R'_{mn})_l$$

and

$$\begin{aligned}\xi &= P'_{mm}(\sigma) + \begin{pmatrix}R'_{mn} & 0\end{pmatrix}\begin{pmatrix} P_{nn}(\sigma)+R_{nn} & 0 \\ \chi & 0\end{pmatrix}^*\begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l\end{pmatrix} \\ &= P'_{mm}(\sigma) + \begin{pmatrix}R'_{mn} & 0\end{pmatrix}\begin{pmatrix} (P_{nn}(\sigma)+R_{nn})^* & 0 \\ \chi(P_{nn}(\sigma)+R_{nn})^* & 1\end{pmatrix}\begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l\end{pmatrix} \\ &= P'_{mm}(\sigma) + \begin{pmatrix}R'_{mn}(P_{nn}(\sigma)+R_{nn})^* & 0\end{pmatrix}\begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l\end{pmatrix} \\ &= P'_{mm}(\sigma) + 0 = P'_{mm}(\sigma)\,.\end{aligned}$$

It follows that

$$
\begin{aligned}
M(0,\sigma,\sigma_k)^{\omega,t} &= \begin{pmatrix} & P'_{mm}(\sigma)^{\omega,t} \\ \begin{pmatrix} P_{nn}(\sigma)+R_{nn} & 0 \\ \chi & 0 \end{pmatrix}^* & \begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l \end{pmatrix} P'_{mm}(\sigma)^{\omega,t} \end{pmatrix} \\
&= \begin{pmatrix} & P'_{mm}(\sigma)^{\omega,t} \\ \begin{pmatrix} (P_{nn}(\sigma)+R_{nn})^* & 0 \\ \chi(P_{nn}(\sigma)+R_{nn})^* & 1 \end{pmatrix} & \begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l \end{pmatrix} P'_{mm}(\sigma)^{\omega,t} \end{pmatrix} \\
&= \begin{pmatrix} P'_{mm}(\sigma)^{\omega,t} \\ \begin{pmatrix} 0 \\ (P'_{mm}(\sigma))_l \end{pmatrix} P'_{mm}(\sigma)^{\omega,t} \end{pmatrix} \\
&= \begin{pmatrix} P'_{mm}(\sigma)^{\omega,t} \\ 0 \\ (P'_{mm}(\sigma))_l P'_{mm}(\sigma)^{\omega,t} \end{pmatrix} .
\end{aligned}
$$

Now, we have for the last component

$$
\begin{aligned}
\left(M(0,\sigma,\sigma_k)^{\omega,t}\right)_{m+n+1} &= (P'_{mm}(\sigma))_l P'_{mm}(\sigma)^{\omega,t} \\
&= \left(P'_{mm}(\sigma) P'_{mm}(\sigma)^{\omega,t}\right)_l \\
&= \left(P'_{mm}(\sigma)^{\omega,t}\right)_l = \omega_l ,
\end{aligned}
$$

where the third equality is by Theorem 5.5.1 of Ésik and Kuich (2007a) and the last equality is by (3.13). In summary, the $(n+m+1)^{\text{th}}$ component of the t^{th} canonical solution of (#), (##) is $(\sigma_k, \omega_l) = (s, v)$. As defined for ω-algebraic systems, it then follows that also the t^{th} canonical solution of $(\divideontimes\divideontimes\divideontimes)$ is (s, v). □

As the mixed ω-algebraic system in the preceding proof does not depend on the previous discussion and since we proved that we can construct the Greibach normal form when needed, we infer the following.

Corollary 3.15. *Let (s, v) be a component of a canonical solution of a mixed ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$.*

Then we can construct an ω-algebraic system over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$ (in Greibach normal form) where (s, v) is a component of a canonical solution.

CHAPTER 4

Weighted Simple Pushdown Automata

This chapter investigates weighted simple pushdown automata. As these automata specialize reset pushdown automata of Kuich and Salomaa (1986, Section 13) (see also Ésik and Kuich, 2007a, pp. 96 ff.), weighted simple pushdown automata are called simple reset pushdown automata in this chapter. Similarly, for infinite words, weighted simple ω-pushdown automata are called simple ω-reset pushdown automata in this chapter. These automata do not use ϵ-transitions and utilize only three simple stack commands: popping a symbol, pushing a symbol or leaving the stack unaltered; moreover, it is only possible to read the topmost stack symbol by popping it. Observe that together with the restriction of not allowing ϵ-transitions, restrictions for the actions on the stack are non-trivial.

We show that for every algebraic and for every ω-algebraic series r, there exists a simple reset or a simple ω-reset pushdown automaton, respectively, with behavior r. For this result we will need some restrictions on the weight structure.

The chapter is divided into two sections. We will first (Section 4.1) concentrate on finite words. In Section 4.2, we will reuse these results to evaluate simple ω-reset pushdown automata. All preliminaries are introduced in the preceding chapter, in Section 3.1.

4.1 Finite Words

The goal of this section is to establish a weighted model of simple pushdown automata on finite words. We will prove as the main result, that for every algebraic series $r \in S^{\mathrm{alg}}\langle\langle \Sigma^* \rangle\rangle$, there exists a simple reset pushdown automaton with behavior r.

We will use simple reset pushdown automata in the next section, Section 4.2, to evaluate the expressive power of simple ω-reset pushdown automata.

This section is structured as follows. Subsection 4.1.1 introduces pushdown matrices that will later be used similarly to an adjacency matrix of the graph representing an automaton. For weighted pushdown automata there exists already the notion of a reset pushdown automaton (cf. Kuich and Salomaa, 1986) that starts and ends with an empty pushdown tape and that naturally allows pushing onto an empty tape. We present weighted simple pushdown automata therefore as a specialization of reset pushdown automata. The same subsection, Subsection 4.1.1, also defines the corresponding reset pushdown matrices and proves some basic properties.

The restrictions we discussed above are defined as simple reset pushdown matrices in Subsection 4.1.2. Here we also prove some basic properties for these matrices.

The last subsection, Subsection 4.1.3, defines how the matrix is used in a simple reset pushdown automaton. Afterwards, we prove that simple reset pushdown automata generate all algebraic series (i.e., weighted context-free languages, cf. Kuich and Salomaa, 1986). The proof starts with algebraic systems (cf. Kuich and Salomaa, 1986) in Greibach normal form and constructs for every such system an equivalent simple reset pushdown automaton. Additionally, we introduce a new normal form for algebraic series.

This section is based on Droste, Dziadek and Kuich (2019a).

4.1.1 Reset Pushdown Matrices

Following Kuich and Salomaa (1986) and Kuich (1997), we introduce pushdown transitions matrices (see also Ésik and Kuich, 2007a). These matrices can be considered as adjacency matrices of graphs representing automata. A special form, the reset pushdown matrices, is used for pushdown automata starting with an empty stack and allowing the automaton to push onto the empty stack. Here, we are interested in simple reset pushdown matrices. This simple form allows the automaton only to push one symbol, to pop one symbol or to ignore the stack. The corresponding automata, the simple reset pushdown automata are a generalization of the unweighted automata used in Chapter 2. They do not use ϵ-transitions and do not allow the inspection of the topmost stack symbol.

A matrix $M \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ is called *row-finite* if $\{\pi' \mid M_{\pi,\pi'} \neq 0\}$ is finite for all $\pi \in \Gamma^*$. We denote the *identity matrix* by E.

Let Γ be an alphabet, called *pushdown alphabet*, and let $n \geq 1$. A matrix $\bar{M} \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ is termed a *pushdown matrix* (with pushdown alphabet Γ and *state set* $\{1,\dots,n\}$) if

(i) $\bar{M}$ is row-finite;

(ii) for all $\pi_1, \pi_2 \in \Gamma^*$,

$$\bar{M}_{\pi_1,\pi_2} = \begin{cases} \bar{M}_{p,\pi}, & \text{if there exist } p \in \Gamma, \pi, \pi' \in \Gamma^* \text{ with } \pi_1 = p\pi' \text{ and } \pi_2 = \pi\pi', \\ 0, & \text{otherwise.} \end{cases}$$

Intuitively, here (ii) means that the infinite pushdown matrix $\bar{M}$ is fully represented already by the blocks $\bar{M}_{p,\pi}$ where $p \in \Gamma$, $\pi \in \Gamma^*$, and (i) means that only finitely many such blocks are nonzero.

In the sequel, *S is assumed to be a complete starsemiring*.

Let Γ be a pushdown alphabet and $\{1,\dots,n\}$ for $n \geq 1$ be a set of states.

A *reset matrix* $M_R \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ is a row-finite matrix such that

$$(M_R)_{\pi_1,\pi_2} = 0 \qquad \text{for } \pi_1, \pi_2 \in \Gamma^* \text{ with } \pi_1 \neq \epsilon\,.$$

A *reset pushdown matrix* $M \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ is the sum of a reset matrix M_R and a pushdown matrix $\bar{M}$,

$$M = M_R + \bar{M}\,.$$

Intuitively, a reset pushdown matrix is similar to a pushdown matrix (i.e., it models the LIFO property of the pushdown tape) with the additional possibility to push onto the empty stack, i.e., $M_{\epsilon,\pi}$ is allowed to be nonzero. Note that the entries of reset pushdown matrices are determined by finitely many values because it is row-finite and property (ii) of pushdown matrices ensures that the value of $M_{p\pi',\pi\pi'}$ is equal to (and therefore can be derived from) $M_{p,\pi}$.

For the convenience of the reader, we recall the following result.

Theorem 4.1 (Corollary 2 of Droste, Ésik and Kuich, 2017)**.** *Let $\bar{M} \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ be a pushdown matrix. Then, for all $\pi_1, \pi_2 \in \Gamma^*$,*

$$(\bar{M}^*)_{\pi_1\pi_2,\epsilon} = (\bar{M}^*)_{\pi_1,\epsilon}(\bar{M}^*)_{\pi_2,\epsilon}\,.$$

Now we show

Theorem 4.2. *Let $M = M_R + \bar{M}$ be a reset pushdown matrix. Then*

$$(M^*)_{\pi,\epsilon} = (\bar{M}^*)_{\pi,\epsilon}(M^*)_{\epsilon,\epsilon} \text{ for } \pi \in \Gamma^*\,.$$

Proof. The proof is by induction on the length of π. The case $\pi = \epsilon$ is trivial. We assume that Theorem 4.2 is proved for $\pi \in \Gamma^*$ and derive it for $p\pi$ with $p \in \Gamma$ as follows, where for $t = 1$ we have $M_{p\pi,\pi_1\pi} = M_{p\pi,\pi}$:

$$\begin{aligned}
(M^*)_{p\pi,\epsilon} &= \sum_{t\geq 1} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^+} M_{p\pi,\pi_1\pi}M_{\pi_1\pi,\pi_2\pi}\cdots M_{\pi_{t-1}\pi,\pi}(M^*)_{\pi,\epsilon} \\
&= \sum_{t\geq 1} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^+} M_{p,\pi_1}M_{\pi_1,\pi_2}\cdots M_{\pi_{t-1},\epsilon}(M^*)_{\pi,\epsilon} \\
&= \Big(\sum_{t\geq 1}(\bar{M}^t)_{p,\epsilon}\Big)(\bar{M}^*)_{\pi,\epsilon}(M^*)_{\epsilon,\epsilon} \\
&= (\bar{M}^*)_{p,\epsilon}(\bar{M}^*)_{\pi,\epsilon}(M^*)_{\epsilon,\epsilon} \\
&= (\bar{M}^*)_{p\pi,\epsilon}(M^*)_{\epsilon,\epsilon}\,.
\end{aligned}$$

The last equality above is implied by Theorem 4.1. □

Corollary 4.3. *Let $M = M_R + \bar{M}$ be a reset pushdown matrix. Then*

$$(M^*)_{p_1\ldots p_k,\epsilon} = (\bar{M}^*)_{p_1,\epsilon}\cdots(\bar{M}^*)_{p_k,\epsilon}(M^*)_{\epsilon,\epsilon}\,,$$

for $p_1, \ldots, p_k \in \Gamma$ ($k \geq 0$).

Theorem 4.4. *Let $M = M_R + \bar{M}$ be a reset pushdown matrix. Then the $S^{n\times n}$-algebraic system with variables $x_\epsilon, \bar{x}_p$ ($p \in \Gamma$)*

$$\begin{aligned}
x_\epsilon &= \sum_{\pi\in\Gamma^*} M_{\epsilon,\pi}\bar{x}_\pi x_\epsilon + E\,, \\
\bar{x}_p &= \sum_{\pi\in\Gamma^*} \bar{M}_{p,\pi}\bar{x}_\pi, \quad p \in \Gamma\,,
\end{aligned}$$

where $\bar{x}_\epsilon = E, \bar{x}_{p\pi} = \bar{x}_p \bar{x}_\pi$ *for* $p \in \Gamma$ *and* $\pi \in \Gamma^*$,
has a solution

$$x_\epsilon = (M^*)_{\epsilon,\epsilon}, \bar{x}_p = (\bar{M}^*)_{p,\epsilon}, \quad p \in \Gamma\,.$$

Proof. By Theorem 4.2, we obtain

$$\begin{aligned}(M^*)_{\epsilon,\epsilon} &= \sum_{\pi\in\Gamma^*} M_{\epsilon,\pi}(M^*)_{\pi,\epsilon} + E \\ &= \sum_{\pi\in\Gamma^*} M_{\epsilon,\pi}(\bar{M}^*)_{\pi,\epsilon}(M^*)_{\epsilon,\epsilon} + E\,,\end{aligned}$$

and

$$(\bar{M}^*)_{p,\epsilon} = \sum_{\pi\in\Gamma^*} \bar{M}_{p,\pi}(\bar{M}^*)_{\pi,\epsilon}\,.$$

By Theorem 4.1, we have $\bar{x}_\pi = (\bar{M}^*)_{\pi,\epsilon}$ for each $\pi \in \Gamma^*$. The result follows. □

Corollary 4.5. *Let* S *be a commutative complete starsemiring and* Σ *be an alphabet. If* $M \in ((S\langle\Sigma\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$ *is a reset pushdown matrix, then the algebraic system of Theorem 4.4 has a unique solution.*

Proof. The algebraic system is strict and thus has a unique solution; see Kuich and Salomaa (1986, p. 302) for details. □

Corollary 4.6. *Let* S *be a commutative complete starsemiring and* Σ *be an alphabet. If* $M \in ((S\langle\Sigma\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$ *is a reset pushdown matrix, then the components of the unique solution of the algebraic system of Theorem 4.4*

$$(M^*)_{\epsilon,\epsilon}, (\bar{M}^*)_{p,\epsilon}, \quad p \in \Gamma\,,$$

are in $(S^{alg}\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$.

Proof. This follows from the definition of $S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$, see Kuich (1997, pp. 622–623) for more information. □

4.1.2 Simple Reset Pushdown Matrices

For the rest of this section, *the complete starsemiring* S *is additionally assumed to be commutative*; and Σ denotes an alphabet.

A reset pushdown matrix M is called *simple* if $M \in ((S\langle\Sigma\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$ for some $n \geq 1$ and for all $p, p_1 \in \Gamma$,

$$M_{p,\epsilon},\ M_{p,p} = M_{\epsilon,\epsilon} \text{ and } M_{p,p_1p} = M_{\epsilon,p_1},$$

are the only blocks $M_{\pi,\pi'}$, where $\pi \in \{\epsilon, p\}$ and $\pi' \in \Gamma^*$, that may be unequal to the zero matrix 0.

Hence, a simple reset pushdown matrix M is defined by its blocks $M_{\epsilon,\epsilon}$ and $M_{p,\epsilon}$, $M_{\epsilon,p}$ ($p \in \Gamma$). Intuitively, the automata will only be allowed to ignore the stack (modeled by $M_{\epsilon,\epsilon}$), pop one symbol ($M_{p,\epsilon}$) or push one symbol ($M_{\epsilon,p}$). Note also that the matrix $M \in \big((S\langle\Sigma\rangle)^{n\times n}\big)^{\Gamma^*\times\Gamma^*}$ forbids ϵ-transitions. Moreover, the equalities $M_{p,p} = M_{\epsilon,\epsilon}$ and $M_{p,p_1p} = M_{\epsilon,p_1}$ imply that the next transition does not depend on the topmost symbol of the stack except when popping it (modeled by $M_{p,\epsilon}$). An example of a simple reset pushdown matrix can be found in Example 4.14.

If M is a simple reset pushdown matrix then the algebraic system of Theorem 4.4 has the form (4.1)

$$\begin{aligned} x_\epsilon &= M_{\epsilon,\epsilon}x_\epsilon + \sum_{p\in\Gamma} M_{\epsilon,p}\bar{x}_p x_\epsilon + E\,, \\ \bar{x}_p &= \bar{M}_{p,\epsilon} + \bar{M}_{p,p}\bar{x}_p + \sum_{p_1\in\Gamma} \bar{M}_{p,p_1p}\bar{x}_{p_1}\bar{x}_p\,,\ \ p\in\Gamma. \end{aligned} \tag{4.1}$$

The variables of this system are $x_\epsilon, \bar{x}_p$ ($p \in \Gamma$). They are variables for matrices in $(S\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$.

Our next lemma states that for simple reset pushdown matrices, emptying the pushdown tape with contents p (i.e., applying $(\bar{M}^*)_{p,\epsilon}$) has the same effect as emptying first the pushdown tape with contents ϵ (i.e., applying $(M^*)_{\epsilon,\epsilon}$) and then reading p in a single move (i.e., applying $M_{p,\epsilon}$). This is due to the fact that p can not be replaced by any other pushdown symbol, but can only be erased. Note that the pushdown matrix $\bar{M}$ cannot continue calculations from the pushdown tape ϵ.

Lemma 4.7. *Let $M = M_R + \bar{M}$ be a simple reset pushdown matrix. Then*

$$(\bar{M}^*)_{p,\epsilon} = (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}\,.$$

Proof. We have

$$\begin{aligned} (\bar{M}^*)_{p,\epsilon} &= \sum_{t\geq 0}\sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} \bar{M}_{p,\pi_1p}\cdots\bar{M}_{\pi_{t-1}p,p}\bar{M}_{p,\epsilon} \\ &= \sum_{t\geq 0}\sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} M_{p,\pi_1p}\cdots M_{\pi_{t-1}p,p}M_{p,\epsilon} \\ &= \Big(\sum_{t\geq 0}\sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} M_{\epsilon,\pi_1}\cdots M_{\pi_{t-1},\epsilon}\Big)M_{p,\epsilon} \\ &= \sum_{t\geq 0}(M^t)_{\epsilon,\epsilon}M_{p,\epsilon} = (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}\,. \end{aligned}$$

For $t = 0$ and $t = 1$ the respective summands are $M_{p,\epsilon}$ and $M_{p,p}M_{p,\epsilon}$.

Observe that the bottom p can be never replaced by another pushdown symbol $p_1 \neq p$; it can only be emptied. Also observe that we use $M_{p,p} = M_{\epsilon,\epsilon}$ in the third equality. □

Our next lemma is similar to Lemma 4.7. This time, a simple reset pushdown matrix $(M^*)_{p,\epsilon}$ is considered. Therefore, in the end, it is possible to empty the pushdown tape with contents ϵ (i.e., apply $(M^*)_{\epsilon,\epsilon}$).

Lemma 4.8. *Let M be a simple reset pushdown matrix. Then*

$$(M^*)_{p,\epsilon} = (M^*)_{\epsilon,\epsilon} M_{p,\epsilon} (M^*)_{\epsilon,\epsilon}$$

Proof. We obtain

$$\begin{aligned}(M^*)_{p,\epsilon} &= \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} M_{p,\pi_1 p} \cdots M_{\pi_{t-1}p,p} M_{p,\epsilon} (M^*)_{\epsilon,\epsilon} \\ &= \big(\sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} M_{\epsilon,\pi_1} \cdots M_{\pi_{t-1},\epsilon}\big) M_{p,\epsilon} (M^*)_{\epsilon,\epsilon} \\ &= (M^*)_{\epsilon,\epsilon} M_{p,\epsilon} (M^*)_{\epsilon,\epsilon}\,. \end{aligned}$$

□

Theorem 4.9. *Let M be a simple reset pushdown matrix. Then $(M^*)_{\epsilon,\epsilon}$ is the unique solution of*

$$x = M_{\epsilon,\epsilon} x + \sum_{p\in\Gamma} M_{\epsilon,p} x M_{p,\epsilon} x + E\,.$$

Proof. By Theorem 4.4, $((M^*)_{\epsilon,\epsilon}, ((\bar{M}^*)_{p,\epsilon})_{p\in\Gamma})$ is the solution of (4.1). Hence, we obtain by Lemma 4.7

$$\begin{aligned}(M^*)_{\epsilon,\epsilon} &= M_{\epsilon,\epsilon}(M^*)_{\epsilon,\epsilon} + \sum_{p\in\Gamma} M_{\epsilon,p}(\bar{M}^*)_{p,\epsilon}(M^*)_{\epsilon,\epsilon} + E \\ &= M_{\epsilon,\epsilon}(M^*)_{\epsilon,\epsilon} + \sum_{p\in\Gamma} M_{\epsilon,p}(M^*)_{\epsilon,\epsilon} M_{p,\epsilon}(M^*)_{\epsilon,\epsilon} + E\end{aligned}$$

This proves that $(M^*)_{\epsilon,\epsilon}$ is a solution of the equation of our theorem. Since $M \in ((S\langle\Sigma\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$, this equation is strict and thus has a unique solution. □

Now we consider the $(S\langle\Sigma\cup\{\epsilon\}\rangle)^{n\times n}$-algebraic system of Theorem 4.9. Recall that the variable x is a variable for $(S\langle\langle\Sigma^*\rangle\rangle)^{n\times n}$. So we substitute the $n\times n$-matrix $X = (x_{i,j})_{1\leq i,j\leq n}$ of variables for $S\langle\langle\Sigma^*\rangle\rangle$ for the variable x and we get the strict $S\langle\Sigma\cup\{\epsilon\}\rangle$-algebraic system

$$X = M_{\epsilon,\epsilon} X + \sum_{p\in\Gamma} M_{\epsilon,p} X M_{p,\epsilon} X + E\,. \tag{4.2}$$

Let $Y = \{x_{i,j} \mid 1\leq i,j\leq n\}$ be the set of the variables of (4.2). Then the support of the right sides of equations of (4.2) is contained in $\{\epsilon\}\cup\Sigma Y\cup\Sigma Y\Sigma Y$. Hence, this system is of Greibach normal form type and at the same time of operator normal form type (see Ésik and Kuich, 2007a, Section 2.2.4).

4.1.3 Simple Reset Pushdown Automata

A reset pushdown automaton starts its computations with empty tape and finishes them with empty tape and final states.

A *reset pushdown automaton* (with input alphabet Σ) $\mathfrak{A} = (n, \Gamma, I, M, P)$ is given by

- a *set of states* $\{1, \ldots, n\}$, $n \geq 1$,
- a *pushdown alphabet* Γ,
- a reset pushdown matrix $M \in ((S\langle\Sigma \cup \{\epsilon\}\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$ called *transition matrix*,
- a row vector $I \in (S\langle\{\epsilon\}\rangle)^{1\times n}$, called *initial state vector*,
- a column vector $P \in (S\langle\{\epsilon\}\rangle)^{n\times 1}$, called *final state vector*.

The *behavior* $\|\mathfrak{A}\|$ of a reset pushdown automaton $\mathfrak{A}$ is defined by

$$\|\mathfrak{A}\| = I(M^*)_{\epsilon,\epsilon}P\,.$$

A reset pushdown automaton $\mathfrak{A} = (n, \Gamma, I, M, P)$ is called *simple* if M is a simple reset pushdown matrix.

Given a series $r \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$, we want to construct a simple reset pushdown automaton with behavior r. By Theorems 5.10 and 5.4 of Kuich (1997), r is a component of the unique solution of a strict algebraic system in Greibach normal form.

Definition 4.10. An algebraic system $x_i = p_i(x)$ (for $1 \leq i \leq n$) is in *Greibach normal form* if, for all $1 \leq i \leq n$, we have

$$\mathrm{supp}(p_i(x)) \subseteq \{\epsilon\} \cup \Sigma \cup \Sigma X \cup \Sigma X X\,. \qquad \blacktriangledown$$

We first consider the algebraic power series r to have $(r, \epsilon) = 0$. So we assume without loss of generality that r is the x_1-component of the unique solution of the algebraic system (4.3) with variables $x_1, \ldots, x_n$

$$x_i = p_i(x), \;\; 1 \leq i \leq n,$$

of the form

$$\begin{aligned} x_i = &\sum_{1\leq j,k\leq n}\sum_{a\in\Sigma}(p_i, ax_jx_k)ax_jx_k + \\ &\sum_{1\leq j\leq n}\sum_{a\in\Sigma}(p_i, ax_j)ax_j + \\ &\sum_{a\in\Sigma}(p_i, a)a\,. \end{aligned} \tag{4.3}$$

We now show the construction of the simple reset pushdown automaton $\mathfrak{A}_m = (n{+}1, \Gamma, I_m, M, P)$ for $1 \leq s \leq n$ with $r = \|\mathfrak{A}_1\|$:
We let $\Gamma = \{x_1, \ldots, x_n\}$; we also denote the state $n{+}1$ by f; the entries of M of the form $(M_{x_k,x_k})_{i,j}$, $(M_{x_k,\epsilon})_{i,j}$, $(M_{\epsilon,x_k})_{i,j}$, $(M_{\epsilon,\epsilon})_{i,j}$, $(M_{\epsilon,\epsilon})_{i,f}$ for $1 \leq i, j, k \leq n$, that may be unequal to 0 are

$$\begin{aligned} (M_{\epsilon,x_k})_{i,j} &= \sum_{a\in\Sigma}(p_i, ax_jx_k)a\,, \\ (M_{x_k,x_k})_{i,j} = (M_{\epsilon,\epsilon})_{i,j} &= \sum_{a\in\Sigma}(p_i, ax_j)a\,, \\ (M_{x_k,\epsilon})_{i,k} = (M_{x_k,x_k})_{i,f} = (M_{\epsilon,\epsilon})_{i,f} &= \sum_{a\in\Sigma}(p_i, a)a\,; \end{aligned}$$

we further put $(I_m)_m = \epsilon$, $(I_m)_i = 0$, for $1 \leq i \leq m-1$ and $m+1 \leq i \leq n+1$; finally let $P_f = \epsilon$ and $P_j = 0$ for $1 \leq j \leq n$.

Intuitively, the variables in the algebraic system are simulated by states in the simple reset pushdown automaton $\mathfrak{A}_m$. By the Greibach normal form, only two variables on the right-hand side are allowed. The first is modeled directly by changing the state, the second is pushed to the pushdown tape and the state is changed to it later when the variable is popped again. The special final state f will only be used as the last state.

Note that $(M_{x_k,x_k})_{i,f}$ allows to change to the final state with a non-empty pushdown tape. This is an artificial addition to fit the definition of simple reset pushdown matrices. Even though the automaton can enter the final state too early, it can not continue from there as it is a sink.

Observe that $\|\mathfrak{A}_m\| = I_m(M^*)_{\epsilon,\epsilon}P = ((M^*)_{\epsilon,\epsilon})_{s,f}$ for $1 \leq s \leq n$. Subsequently we will show that $\|\mathfrak{A}_1\| = r$.

This simple reset pushdown matrix M is called the simple pushdown matrix *induced* by the Greibach normal form (4.3). The simple reset pushdown automata $\mathfrak{A}_m$, $1 \leq s \leq n$, are called the simple reset pushdown automata *induced* by the Greibach normal form (4.3).

The next lemma formalizes the meaning of the pushdown tape for induced simple reset pushdown matrices. Intuitively, going from state j to the final state f and erasing the variable x_k from the pushdown tape on the way (i.e., applying $((M^*)_{x_k,\epsilon})_{j,f}$) has the same effect as first going from state j to the final state f without changing the pushdown tape (i.e., applying $((M^*)_{\epsilon,\epsilon})_{j,f}$) and then restarting in state k (i.e., applying $((M^*)_{\epsilon,\epsilon})_{k,f}$). This results from the definition of $\mathfrak{A}_m$: popping a variable from the pushdown tape and changing to its state has the same weight as changing to the final state instead. It allows the automaton to process the variables in the algebraic system individually.

Lemma 4.11. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3)*. Then, for all $1 \leq j,k \leq n$,*

$$((M^*)_{x_k,\epsilon})_{j,f} = ((M^*)_{\epsilon,\epsilon})_{j,f}((M^*)_{\epsilon,\epsilon})_{k,f}\,.$$

Proof. We obtain

$$\begin{aligned}((M^*)_{\epsilon,\epsilon})_{j,f} &= ((M^+)_{\epsilon,\epsilon})_{j,f} = ((M^*M)_{\epsilon,\epsilon})_{j,f} \\ &= \sum_{1\leq t_1\leq f} ((M^*)_{\epsilon,\epsilon})_{j,t_1}(M_{\epsilon,\epsilon})_{t_1,f} + \sum_{1\leq t_1\leq f}\sum_{1\leq t\leq n} ((M^*)_{\epsilon,x_t})_{j,t_1}(M_{x_t,\epsilon})_{t_1,f} \\ &= ((M^*)_{\epsilon,\epsilon}M_{\epsilon,\epsilon})_{j,f}\end{aligned}$$

since $(M_{x_t,\epsilon})_{t_1,f} = 0$ for all $1 \leq t_1 \leq f$ and $1 \leq t \leq n$ by our construction.

We now obtain, by Lemma 4.8,

$$\begin{aligned}((M^*)_{x_k,\epsilon})_{j,f} &= \sum_{1\le t_1,t_2\le f} ((M^*)_{\epsilon,\epsilon})_{j,t_1}(M_{x_k,\epsilon})_{t_1,t_2}((M^*)_{\epsilon,\epsilon})_{t_2,f} \\ &= \sum_{1\le t_1\le f} ((M^*)_{\epsilon,\epsilon})_{j,t_1}(M_{\epsilon,\epsilon})_{t_1,f}((M^*)_{\epsilon,\epsilon})_{k,f} \\ &= ((M^*)_{\epsilon,\epsilon}M_{\epsilon,\epsilon})_{j,f}((M^*)_{\epsilon,\epsilon})_{k,f} \\ &= ((M^*)_{\epsilon,\epsilon})_{j,f}((M^*)_{\epsilon,\epsilon})_{k,f}\,.\end{aligned}$$

The second equality is implied by the fact that

$$(M_{x_k,\epsilon})_{t_1,k} = (M_{\epsilon,\epsilon})_{t_1,f} \text{ and}$$

$$(M_{x_k,\epsilon})_{t_1,t_2} = 0 \text{ for } t_2 \neq k\,. \qquad \square$$

Now we show that the constructed automata realize the algebraic system (4.3).

Theorem 4.12.

$$(\|\mathfrak{A}_1\|,\ldots,\|\mathfrak{A}_n\|) = (((M^*)_{\epsilon,\epsilon})_{1,f},\ldots,((M^*)_{\epsilon,\epsilon})_{n,f})$$

is the unique solution of the algebraic system (4.3)*. In particular,* $r = \|\mathfrak{A}_1\|$.

Proof. We obtain, for $1 \le i \le n$, by substituting into the right sides of (4.3) and by Lemma 4.11,

$$\begin{aligned}&\sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i a x_j x_k)a((M^*)_{\epsilon,\epsilon})_{j,f}((M^*)_{\epsilon,\epsilon})_{k,f} + \\ &\qquad\qquad \sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, a x_j)a((M^*)_{\epsilon,\epsilon})_{j,f} + \sum_{a\in\Sigma}(p_i,a)a \\ &= \sum_{1\le j,k\le n}(M_{\epsilon,x_k})_{i,j}((M^*)_{x_k,\epsilon})_{j,f} + \sum_{1\le j\le n}(M_{\epsilon,\epsilon})_{i,j}((M^*)_{\epsilon,\epsilon})_{j,f} + (M_{\epsilon,\epsilon})_{i,f} \\ &= \sum_{1\le k\le n}(M_{\epsilon,x_k}(M^*)_{x_k,\epsilon})_{i,f} + (M_{\epsilon,\epsilon}(M^*)_{\epsilon,\epsilon})_{i,f} \\ &= ((M^+)_{\epsilon,\epsilon})_{i,f} = ((M^*)_{\epsilon,\epsilon})_{i,f}\,.\end{aligned}$$

Here in the second equality, we have replaced $(M_{\epsilon,\epsilon})_{i,f}$ by $(M_{\epsilon,\epsilon})_{i,f}((M^*)_{\epsilon,\epsilon})_{f,f}$, since $((M^*)_{\epsilon,\epsilon})_{f,f} = \epsilon$; also note that $(M_{\epsilon,x_t})_{i,f} = 0$. Since the algebraic system (4.3) is strict, it has a unique solution. In particular, $r = \|\mathfrak{A}_1\|$. $\square$

Note that the automaton $\mathfrak{A}_m$ used in Theorem 4.12 is induced by the Greibach normal form (4.3) for the series r with $(r,\epsilon) = 0$. We now consider the second case.

If we are given a series $r \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle$, where $(r,\epsilon) \neq 0$, then we modify the reset pushdown automaton $\mathfrak{A}_1$ to obtain $\mathfrak{A}' = (n+2, \Gamma, I', M', P')$. The new state $n+2$ is an

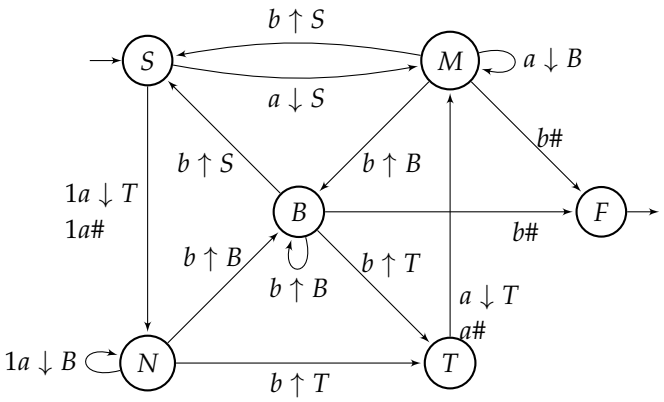


Figure 4.1: Example 4.14: Simple reset pushdown automaton, where $\downarrow X$ means push symbol X, $\uparrow X$ means pop X, and # leaves the stack unaltered. All shown transitions have a weight equal to the natural number 0 except the two transitions reading letter a and pushing B onto the stack that have weight 1. All other possible transitions have weight $-\infty$.

isolated state, i.e., no moves to $n+2$ or from $n+2$ are possible. This means that, for all $\pi_1, \pi_2 \in \Gamma^*$,

$$M'_{\pi_1,\pi_2} = \begin{pmatrix} M_{\pi_1,\pi_2} & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$(M'^*)_{\pi_1,\pi_2} = \begin{pmatrix} (M^*)_{\pi_1,\pi_2} & 0 \\ 0 & \delta_{\pi_1,\pi_2} \end{pmatrix},$$

where δ_{π_1,π_2} is the Kronecker delta. Moreover let $I' = \begin{pmatrix} I_1 & \epsilon \end{pmatrix}$ and $P' = \begin{pmatrix} P \\ (r,\epsilon)\epsilon \end{pmatrix}$. Hence, we obtain

$$\|\mathfrak{A}'\| = I'(M'^*)_{\epsilon,\epsilon}P' = I_1(M^*)_{\epsilon,\epsilon}P + (r,\epsilon)\epsilon = r\,.$$

This proves

Corollary 4.13. *Let $r \in S^{alg}\langle\langle\Sigma^*\rangle\rangle$. Then there exists a simple reset pushdown automaton with behavior r.*

Example 4.14. Consider the semiring $\bar{\mathbb{N}}\langle\langle\Sigma^*\rangle\rangle$ for the arctic semiring $\langle\bar{\mathbb{N}}, \max, +, -\infty, 0\rangle$ with $\bar{\mathbb{N}} = \mathbb{N} \cup \{-\infty, \infty\}$. Analogously to Example 3.7, we let $\mathbb{0} = -\infty$ and $\mathbb{1} = 0$ and we note that in the following, 1 stands for the natural number 1.

We define the algebraic system

$$\begin{aligned} S &= aMS + 1aNT + 1aN & T &= aMT + aM \\ M &= b + aMB & N &= b + 1aNB \\ B &= b \end{aligned}$$

with the variables S, T, M, N, B. These variable facilitate reading the equations, but to make it fit into equation (4.3), consider the variable mapping $x_1 = T, x_2 = S, x_3 = N, x_4 = M, x_5 = B$.

Now, the variable M derives a string $a^n b^{n+1}$ for $n \in \mathbb{N}$. The variable N does the same but at the same time produces the weight n. The variables S and T add another a.

Let $L = \{a^n b^n \mid n \geq 1\}$. In total, the second component (i.e., with S being the start variable) of the least solution is u with $(u, a^{n_1} b^{n_1} a^{n_2} b^{n_2} \dots a^{n_k} b^{n_k}) = \max n_i$ for $k \geq 1$ and $(u, w) = -\infty$ for $w \notin L^+$.

From this, we can construct a simple reset pushdown automaton $\mathfrak{A}_2 = (n, \Gamma, I, M, P)$ as shown in Figure 4.1. Thus, we have $n = 6$, $\Gamma = \{T, S, N, M, B\}$. The initial state vector is $I_2 = \epsilon$ and $I_i = 0$ for $i \neq 2$. The final state vector is $P_6 = \epsilon$ and $P_i = 0$ for $i \neq 6$. The simple reset pushdown matrix is defined as

$$M = \begin{pmatrix} M_{\epsilon,\epsilon} & M_{\epsilon,T} & M_{\epsilon,S} & M_{\epsilon,N} & M_{\epsilon,M} & M_{\epsilon,B} & \cdots \\ M_{T,\epsilon} & M_{\epsilon,\epsilon} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \cdots \\ M_{S,\epsilon} & \mathbb{0} & M_{\epsilon,\epsilon} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \cdots \\ M_{N,\epsilon} & \mathbb{0} & \mathbb{0} & M_{\epsilon,\epsilon} & \mathbb{0} & \mathbb{0} & \cdots \\ M_{M,\epsilon} & \mathbb{0} & \mathbb{0} & \mathbb{0} & M_{\epsilon,\epsilon} & \mathbb{0} & \cdots \\ M_{B,\epsilon} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & M_{\epsilon,\epsilon} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

with, for instance

$$M_{\epsilon,\epsilon} = \begin{pmatrix} \mathbb{0} & \mathbb{0} & \mathbb{0} & a & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & 1a & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & b \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & b \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \end{pmatrix} \text{ and } M_{\epsilon,B} = \begin{pmatrix} \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & 1a & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & a & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \end{pmatrix}.$$

The rest of the matrix M can be inferred by the rules of pushdown matrices. The behavior $\|\mathfrak{A}_2\|$ is equal to the second component of the least solution of the algebraic system above. ▽

Theorem 4.12 in connection with Theorem 4.9 yields a new normal form for algebraic power series.

Theorem 4.15. *Let $r \in S^{alg}\langle\langle \Sigma^* \rangle\rangle$ with $(r, \epsilon) = 0$. Then for some $n \geq 2$, there exist matrices $M_0, M_{1,t}, M_{2,t} \in (S\langle\langle \Sigma \rangle\rangle)^{n \times n}$, $1 \leq t \leq n-1$ such that r is the $(1, n)$-component of the unique solution of the algebraic system*

$$X = M_0 X + \sum_{1 \leq t \leq n-1} M_{1,t} X M_{2,t} X + E,$$

where X is an $n \times n$-matrix of variables.

Proof. Assume that r equals the first component of the unique solution of the algebraic system (4.3) with $n-1$ variables $x_1, \ldots, x_{n-1}$. Let $M \in ((S\langle\Sigma\rangle)^{n\times n})^{\Gamma^*\times\Gamma^*}$, with $\Gamma = \{x_1, \ldots, x_{n-1}\}$, be the simple pushdown matrix induced by (4.3). Then by Theorem 4.12, $((M^*)_{\epsilon,\epsilon})_{1,n}$ is the first component of the solution of (4.3) and $r = ((M^*)_{\epsilon,\epsilon})_{1,n}$.

By Theorem 4.9, $(M^*)_{\epsilon,\epsilon}$ is the solution of equation (4.2). Let now $M_0 = M_{\epsilon,\epsilon}$, $M_{1,t} = M_{\epsilon,x_t}$ and $M_{2,t} = M_{x_t,\epsilon}$ for $1 \le t \le n-1$.

Then equation (4.2) now reads

$$X = M_0 X + \sum_{1\le t\le n-1} M_{1,t} X M_{2,t} X + E \tag{4.4}$$

and r is the $(1,n)$-component of its unique solution. □

In language theory, the *restricted Dyck languages* $D_n'^*$ $(n \ge 1)$ are formed of the words over n pairs of associated parentheses which are "well-formed" in the usual sense. Here a word is considered to be "well-formed" iff successive deletions of subwords of associated parentheses not containing any further parentheses, say $(,)$, $[,]$, ... yield the empty word. By Theorem II. 3.7. of Berstel (1979), $D_n'^*$ $(n \ge 1)$ is generated by the context-free grammar with productions

$$x \to \epsilon, \quad x \to a_k x \bar{a}_k x, \qquad 1 \le k \le n\,.$$

Here a_k and $\bar{a}_k$ are the pairs of associated parentheses. By Theorem VII. 1.2. of Berstel (1979), any of the languages $D_n'^*$ $(n \ge 2)$ is a cone generator of the principal cone of context-free languages.

These results were transferred by Kuich and Salomaa (1986) to algebraic power series over commutative semirings S. The *restricted Dyck series* $D_n'^*$ $(n \ge 1)$ are now the unique solutions of the strict algebraic systems

$$x = \sum_{1\le k\le n} a_k x \bar{a}_k x + \epsilon$$

and $D_2'^*$, and hence $D_n'^*$ for $n \ge 2$, are cone generators of the principal cone $S^{\mathrm{alg}}\langle\langle\Sigma_\infty^*\rangle\rangle$ of algebraic power series. (See Theorem 13.15 of Kuich and Salomaa, 1986.)

In Example 14.3 of Kuich and Salomaa (1986), it is described how a "master system" generates a normal form for algebraic system. The "master system" generating equation (4.4) reads now, for a given $n \ge 2$,

$$x = ax + \sum_{1\le k\le n-1} a_k x \bar{a}_k x + \epsilon\,.$$

The important difference to the normal form given in this Example 14.3 is that now all M-matrices contain no ϵ-terms.

4.2 Infinite Words

This section evaluates the expressive power of weighted simple pushdown automata of infinite words, i.e., simple ω-reset pushdown automata. In our main result of this section, we show that these simple ω-reset pushdown automata recognize all weighted ω-context-free languages.

For our proof, we use that ω-algebraic systems can be brought into Greibach normal form as shown in Chapter 3. Our construction of simple ω-reset pushdown automata is deduced from the construction in Section 4.1.

This automaton model will be needed in Chapter 5 for an equivalence result between weighted ω-context-free languages and weighted logical formalisms for infinite words.

In Subsection 4.2.1, we describe basic properties of reset pushdown matrices if applied infinitely many times. The results described there are extensions of the results given in Subsections 4.1.1 and 4.1.2.

Simple ω-reset pushdown automata are introduced in Subsection 4.2.2. The main result of this subsection and of the whole chapter is that for each ω-algebraic series r it is possible to construct a simple ω-reset pushdown automaton with behavior r. The proof of this main result is performed by the following proof method using the uniqueness of l^{th} canonical solutions of ω-algebraic systems: The proof that each of two expressions is the m^{th} component of the l^{th} canonical solution implies the equality of these two expressions. (Compare this with the proof method in continuous semirings: The proof that each of two expressions is the m^{th} component of the least solution of an algebraic system implies the equality of these two expressions.) We consider an ω-algebraic series that is the m^{th} component of the l^{th} canonical solution of an ω-algebraic system in Greibach normal form and construct a simple ω-reset pushdown automaton whose moves depend only on the coefficients of this Greibach normal form. We prove that the behavior of this simple ω-reset pushdown automaton equals the m^{th} component of the l^{th} canonical solution of this Greibach normal form. Hence, we conclude that for each ω-algebraic series r we can construct a simple ω-reset pushdown automaton with behavior r.

This section is based on Droste, Dziadek and Kuich (2019b, 2020c).

4.2.1 Infinite Applications of Simple Reset Pushdown Matrices

In this subsection we prove some results for infinite applications of simple reset pushdown matrices.

In this section, (S, V) *is a complete semiring-semimodule pair*.

We will use sets P_l comprising infinite sequences over $\{1, \ldots, n\}$ as defined by Droste, Ésik and Kuich (2017):

$$P_l = \{(j_1, j_2, \ldots) \in \{1, \ldots, n\}^\omega \mid j_t \leq l \text{ for infinitely many } t \geq 1\}\,.$$

We obtain, for a reset pushdown matrix $M \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$, $\pi \in \Gamma^+$ and for $1 \leq j \leq n$,

$$((M^{\omega,l})_\pi)_j = \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} \sum_{(j_1,j_2,\dots)\in P_l} (M_{\pi,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}(M_{\pi_2,\pi_3})_{j_2,j_3}\cdots . \tag{4.5}$$

Observe the following summation identity: Assume that $M_1, M_2, \dots$ are matrices in $S^{n\times n}$. Then for $0 \leq l \leq n$, $1 \leq j \leq n$, and $m \geq 1$, we have

$$\sum_{(j_1,j_2,\dots)\in P_l} (M_1)_{j,j_1}(M_2)_{j_1,j_2}\cdots = \sum_{1\leq j_1,\dots,j_m\leq n} (M_1)_{j,j_1}\cdots(M_m)_{j_{m-1},j_m} \sum_{(j_{m+1},j_{m+2},\dots)\in P_l} (M_{m+1})_{j_m,j_{m+1}}\cdots .$$

By Theorem 5.5.1 of Ésik and Kuich (2007a) we obtain, for a finite matrix M and for $0 \leq l \leq n$, the equality $MM^{\omega,l} = M^{\omega,l}$. By Theorem 6 of Droste, Ésik and Kuich (2017), we have a similar result for pushdown matrices. We will now show the same equality for a reset pushdown matrix M.

Theorem 4.16. *Let (S, V) be a complete semiring-semimodule pair and let further $M \in (S^{n\times n})^{\Gamma^*\times\Gamma^*}$ be a reset pushdown transition matrix. Then, for $0 \leq l \leq n$,*

$$M^{\omega,l} = MM^{\omega,l}\,.$$

Proof. We obtain for $\pi_0 \in \Gamma^*$ and $1 \leq j_0 \leq n$,

$$\begin{aligned}((MM^{\omega,l})_{\pi_0})_{j_0} &= \sum_{\pi\in\Gamma^*}\sum_{1\leq j\leq n}(M_{\pi_0,\pi})_{j_0,j}\sum_{\pi_1,\pi_2,\dots\in\Gamma^*}\sum_{(j_1,j_2\dots)\in P_l}(M_{\pi,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}\cdots\\ &= \sum_{\pi,\pi_1,\pi_2\dots\in\Gamma^*}\sum_{(j,j_1,j_2,\dots)\in P_l}(M_{\pi_0,\pi})_{j_0,j}(M_{\pi,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}\cdots\\ &= ((M^{\omega,l})_{\pi_0})_{j_0}.\end{aligned}$$

□

The following two theorems compare reset pushdown matrices to pushdown matrices in the course of an infinite application. They state that either the topmost stack symbol p is popped or the reset pushdown matrix behaves similar to pushdown matrices.

Theorem 4.17. *Let (S, V) be a complete semiring-semimodule pair. Let M be a reset pushdown matrix. Then*

$$(M^\omega)_p = (\bar{M}^\omega)_p + (\bar{M}^*)_{p,\epsilon}(M^\omega)_\epsilon, \text{ for any } p \in \Gamma\,.$$

Proof. We obtain, for $p \in \Gamma$,

$$\begin{aligned}(M^\omega)_p &= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^+} M_{p,\pi_1}M_{\pi_1,\pi_2}\cdots + \sum_{t\geq 1}\sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^+} M_{p,\pi_1}\cdots M_{\pi_{t-1},\epsilon}(M^\omega)_\epsilon\\ &= (\bar{M}^\omega)_p + \sum_{t\geq 1}(\bar{M}^t)_{p,\epsilon}(M^\omega)_\epsilon\\ &= (\bar{M}^\omega)_p + (\bar{M}^*)_{p,\epsilon}(M^\omega)_\epsilon\,.\end{aligned}$$

□

Theorem 4.18. *Let (S, V) be a complete semiring-semimodule pair. Let M be a reset pushdown matrix and $0 \leq l \leq n$. Then*

$$(M^{\omega,l})_p = (\bar{M}^{\omega,l})_p + (\bar{M}^*)_{p,\epsilon}(M^{\omega,l})_\epsilon, \text{ for any } p \in \Gamma .$$

Proof. We use the proof of Theorem 4.17. We obtain, for $p \in \Gamma$, $0 \leq l \leq n$ and $1 \leq j \leq n$,

$$\begin{aligned}
((M^{\omega,l})_p)_j &= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}\cdots \\
&= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^+} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}\cdots \\
&\quad + \sum_{t\geq 1} \sum_{\substack{\pi_1,\dots,\pi_{t-1}\in\Gamma^+ \\ \pi_{t+1},\pi_{t+2},\cdots\in\Gamma^*}} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1})_{j,j_1}\cdots(M_{\pi_{t-1},\epsilon})_{j_{t-1},j_t}\cdot \\
&\qquad\qquad (M_{\epsilon,\pi_{t+1}})_{j_t,j_{t+1}}(M_{\pi_{t+1},\pi_{t+2}})_{j_{t+1},j_{t+2}}\cdots \\
&= ((\bar{M}^{\omega,l})_p)_j + \sum_{t\geq 1} \sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^+} \sum_{1\leq j_1,\dots,j_t\leq n} (M_{p,\pi_1})_{j,j_1}\cdots(M_{\pi_{t-1},\epsilon})_{j_{t-1},j_t}\cdot \\
&\qquad \sum_{\pi_{t+1},\pi_{t+2},\cdots\in\Gamma^*} \sum_{(j_{t+1},j_{t+2},\dots)\in P_l} (M_{\epsilon,\pi_{t+1}})_{j_t,j_{t+1}}(M_{\pi_{t+1},\pi_{t+2}})_{j_{t+1},j_{t+2}}\cdots \\
&= ((\bar{M}^{\omega,l})_p)_j + \sum_{t\geq 1} \sum_{1\leq j'\leq n} ((\bar{M}^t)_{p,\epsilon})_{j,j'}((M^{\omega,l})_\epsilon)_{j'} \\
&= ((\bar{M}^{\omega,l})_p + (\bar{M}^*)_{p,\epsilon}(M^{\omega,l})_\epsilon)_j .
\end{aligned}$$

□

For simple reset pushdown matrices, the following two lemmas state that infinite paths starting with symbol p on the pushdown tape can either ignore that symbol or pop it and then continue with an infinite path from the empty tape.

Lemma 4.19. *Let (S, V) be a complete semiring-semimodule pair. Let M be a simple reset pushdown matrix. Then, for $p \in \Gamma$,*

$$(M^\omega)_p = (M^\omega)_\epsilon + (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}(M^\omega)_\epsilon .$$

Proof. We obtain, for $p \in \Gamma$,

$$\begin{aligned}
(M^\omega)_p &= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} M_{p,\pi_1}M_{\pi_1,\pi_2}\cdots \\
&= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} M_{p,\pi_1 p}M_{\pi_1 p,\pi_2 p}\cdots + \sum_{t\geq 0} \sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^*} M_{p,\pi_1 p}\cdots M_{\pi_{t-1}p,p}M_{p,\epsilon}(M^\omega)_\epsilon \\
&= (M^\omega)_\epsilon + \Big(\sum_{t\geq 0} \sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^*} M_{\epsilon,\pi_1}\cdots M_{\pi_{t-1},\epsilon}\Big)M_{p,\epsilon}(M^\omega)_\epsilon \\
&= (M^\omega)_\epsilon + \sum_{t\geq 0}(M^t)_{\epsilon,\epsilon}M_{p,\epsilon}(M^\omega)_\epsilon \\
&= (M^\omega)_\epsilon + (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}(M^\omega)_\epsilon .
\end{aligned}$$

□

Lemma 4.20. *Let (S, V) be a complete semiring-semimodule pair. Let M be a simple reset pushdown matrix. Then, for $p \in \Gamma$ and $0 \leq l \leq n$,*

$$(M^{\omega,l})_p = (M^{\omega,l})_\epsilon + (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}(M^{\omega,l})_\epsilon .$$

Proof. We use the proof of Lemma 4.19. We obtain, for $p \in \Gamma$, $0 \leq l \leq n$ and $1 \leq j \leq n$,

$$\begin{aligned}
((M^{\omega,l})_p)_j &= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1})_{j,j_1}(M_{\pi_1,\pi_2})_{j_1,j_2}\cdots \\
&= \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1 p})_{j,j_1}(M_{\pi_1 p,\pi_2 p})_{j_1,j_2}\cdots \\
&\quad + \sum_{t\geq 0} \sum_{\pi_1,\pi_2,\cdots\in\Gamma^*} \sum_{(j_1,j_2,\dots)\in P_l} (M_{p,\pi_1 p})_{j,j_1}\cdots(M_{\pi_{t-1}p,p})_{j_{t-1},j_t}\cdot \\
&\qquad (M_{p,\epsilon})_{j_t,j_{t+1}}(M_{\epsilon,\pi_{t+2}})_{j_{t+1},j_{t+2}}(M_{\pi_{t+2},\pi_{t+3}})_{j_{t+2},j_{t+3}}\cdots \\
&= ((M^{\omega,l})_\epsilon)_j \\
&\quad + \sum_{t\geq 0} \sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^*} \sum_{1\leq j_1,\dots,j_{t+1}\leq n} (M_{p,\pi_1 p})_{j,j_1}\cdots(M_{\pi_{t-1}p,p})_{j_{t-1},j_t}(M_{p,\epsilon})_{j_t,j_{t+1}}\cdot \\
&\qquad \sum_{\pi_{t+2},\pi_{t+3},\cdots\in\Gamma^*} \sum_{(j_{t+2},j_{t+3},\dots)\in P_l} (M_{\epsilon,\pi_{t+2}})_{j_{t+1},j_{t+2}}(M_{\pi_{t+2},\pi_{t+3}})_{j_{t+2},j_{t+3}}\cdots \\
&= ((M^{\omega,l})_\epsilon)_j + \sum_{t\geq 0} \sum_{\pi_1,\dots,\pi_{t-1}\in\Gamma^*} \sum_{1\leq j_1,\dots,j_{t+1}\leq n} (M_{\epsilon,\pi_1})_{j,j_1}\cdots(M_{\pi_{t-1},\epsilon})_{j_{t-1},j_t}\cdot \\
&\qquad (M_{p,\epsilon})_{j_t,j_{t+1}}((M^{\omega,l})_\epsilon)_{j_{t+1}} \\
&= ((M^{\omega})_\epsilon)_j + \sum_{t\geq 0} \sum_{1\leq j',j''\leq n} ((M^t)_{\epsilon,\epsilon})_{j,j'}(M_{p,\epsilon})_{j',j''}((M^{\omega,l})_\epsilon)_{j''} \\
&= ((M^{\omega})_\epsilon + (M^*)_{\epsilon,\epsilon}M_{p,\epsilon}(M^{\omega,l})_\epsilon)_j\,.
\end{aligned}$$

□

4.2.2 Simple ω-Reset Pushdown Automata

In this section, we introduce simple ω-reset pushdown automata, and the main theorem will show that they can recognize all ω-algebraic series.

An ω-reset pushdown automaton

$$\mathfrak{A} = (n, \Gamma, I, M, P, l)$$

is given by a reset pushdown automaton (n, Γ, I, M, P) and an integer l with $0 \leq l \leq n$, which indicates that $1, \dots, l$ are the repeated states of $\mathfrak{A}$. The behavior $\|\mathfrak{A}\|$ of this ω-reset pushdown automaton $\mathfrak{A}$ is defined by

$$\|\mathfrak{A}\| = I(M^*)_{\epsilon,\epsilon}P + I(M^{\omega,l})_\epsilon\,.$$

The ω-reset pushdown automaton $\mathfrak{A} = (n, \Gamma, I, M, P, l)$ is called *simple* if M is a simple reset pushdown matrix.

Example 4.21. Figure 4.2 shows a simple ω-reset pushdown automaton $\mathcal{A} = (4, \Gamma, I, M, P, 1)$ over the quemiring $\mathbb{N}^\infty\langle\langle\Sigma^*\rangle\rangle \times \mathbb{N}^\infty\langle\langle\Sigma^\omega\rangle\rangle$ for the tropical semiring $\langle\mathbb{N}^\infty, \min, +, \mathbb{0} = \infty, \mathbb{1} = 0\rangle$ with $\Sigma = \{a, b, c\}$, $\Gamma = \{Z_0, X\}$, $I_2 = 0$, $I_i = \infty$ for $i \neq 2$ and $P_i = \infty$ for all $1 \leq i \leq 4$. Then the adjacency matrix M of the automaton shown in Figure 4.2 is

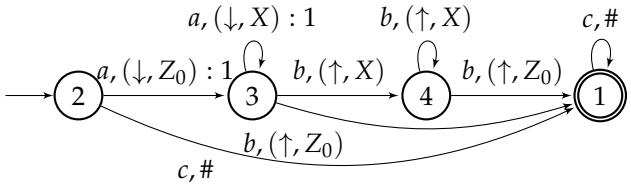


Figure 4.2: Example 4.21: Simple ω-reset pushdown automaton, where $(\downarrow, X)$ means push symbol X, $(\uparrow, X)$ means pop X, and # leaves the stack unaltered. All transitions shown have a weight equal to the natural number 0 except the two transitions reading letter a and pushing a symbol onto the stack that have weight 1. All other possible transitions have weight ∞.

a simple reset pushdown matrix. As an indication, M is defined with $(M_{\epsilon,\epsilon})_{1,1} = 0c$, $(M_{\epsilon,\epsilon})_{2,1} = 0c$, $(M_{\epsilon,Z_0})_{2,3} = 1a$, etc., resulting in e.g.,

$$M_{\epsilon,\epsilon} = \begin{pmatrix} 0c & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ 0c & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \\ \mathbb{0} & \mathbb{0} & \mathbb{0} & \mathbb{0} \end{pmatrix} \text{ and finally } M = \begin{pmatrix} M_{\epsilon,\epsilon} & M_{\epsilon,Z_0} & M_{\epsilon,X} & \cdots \\ M_{Z_0,\epsilon} & M_{\epsilon,\epsilon} & \mathbb{0} & \cdots \\ M_{X,\epsilon} & \mathbb{0} & M_{\epsilon,\epsilon} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where the excluded part of M can be derived from the rules of pushdown and simple reset pushdown matrices. The automaton $\mathcal{A}$ has the behavior $a^n b^n c^\omega \mapsto n$, similar to the mixed ω-algebraic system in Example 3.4. ▽

Example 4.22. Reconsider Example 4.14. We define the simple ω-reset pushdown automaton $\mathfrak{A}_2 = (6, \Gamma, I, M, P, 1)$ where we assume the state ordering T, S, N, M, B, F to make state T Büchi-accepting. The behavior in the semiring part is equal to before; the behavior in the semimodule part is u with $(u, a^{n_1} b^{n_1} a^{n_2} b^{n_2} \ldots) = \max n_i$ and $(u, w) = -\infty$ for $w \notin \{a^n b^n \mid n \geq 1\}^\omega$. ▽

Example 4.23. Consider the ω-algebraic system

$$\begin{aligned} y_1 &= a + cy_1 \\ y_2 &= ay_1 y_2 + ay_1 \,. \end{aligned} \tag{4.6}$$

We will consider the second component of the first canonical solution, i.e., variable y_1 is Büchi-accepting and variable y_2 is the start variable.

The ω-algebraic system induces the following mixed ω-algebraic system

$$\begin{aligned} x_1 &= a + cx_1 & z_1 &= cz_1 \\ x_2 &= ax_1 x_2 + ax_1 & z_2 &= az_1 + ax_1 z_2 \,. \end{aligned} \tag{4.7}$$

The least solution of $x = p(x)$ is

$$\sigma = \begin{pmatrix} c^* a \\ (ac^* a)^+ \end{pmatrix}.$$

Now, we write the linear system $z = \rho(\sigma)z$ in the matrix form and compute the first canonical solution.

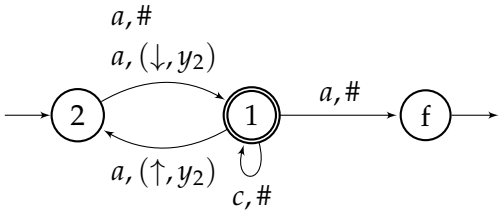
$$\begin{aligned}\rho(\sigma)^{\omega,1} &= \begin{pmatrix} c & 0 \\ a & a(c^*a) \end{pmatrix}^{\omega,1} \\ &= \begin{pmatrix} (c + 0(aa)^*a)^\omega \\ (ac^*a)^*a(c + 0(aa)^*a)^\omega \end{pmatrix} \\ &= \begin{pmatrix} c^\omega \\ (ac^*a)^*ac^\omega \end{pmatrix} =: \begin{pmatrix} \omega_1^{(1)} \\ \omega_2^{(1)} \end{pmatrix} = \omega^{(1)}\end{aligned}$$

Note that the second component, $\omega_2^{(1)}$, does not contain the ω-words $(ac^*a)^\omega$ even though for an unweighted ω-context-free grammar corresponding to (4.6), the derivation

$$y_2 \to ay_1y_2 \to (aa)y_2 \to (aa)ay_1y_2 \to (aa)^2y_2 \to^\omega a^\omega$$

would be successful even with only y_1 Büchi-accepting. The difference is due to the fact that y_1 is not *significant* in the ω-algebraic system above because it is exchanged by x_1 in the derivation (for more information, see Ésik and Kuich, 2007a, pp. 140 ff.).

Now, we look at the simple ω-reset pushdown automaton induced by the ω-algebraic system (4.6):



The behavior of this automaton is

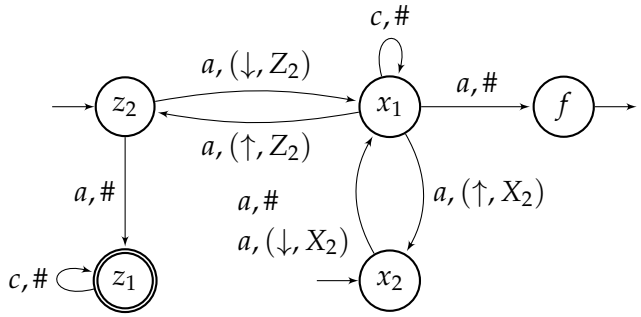
$$\begin{aligned}&(((M^*)_{\epsilon,\epsilon})_{1,f}, ((M^*)_{\epsilon,\epsilon})_{2,f}; ((M^{\omega,1})_\epsilon)_1, ((M^{\omega,1})_\epsilon)_2) \\ =\,&(c^*a, (ac^*a)^+; c^\omega, (ac^*a)^*ac^\omega + (ac^*a)^\omega)\end{aligned}$$

Here, the first two components are equal to σ, as desired. But the last component differs from $\omega_2^{(1)}$; the last component is however equal to the behavior of unweighted ω-context-free grammars.

Note that the desired component $\omega_2^{(1)} = (ac^*a)^*ac^\omega$ is not recognized by this automaton, even when changing the Büchi-accepting states. If no states are Büchi-accepting, the behavior is 0, if all of them are Büchi-accepting, we have the same behavior as above. If only state 2 is Büchi-accepting (can be achieved by renaming), we only recognize $(ac^*a)^\omega$.

We now propose a different construction; this new construction models exactly the behavior of ω-algebraic systems. The following is the simple ω-reset pushdown automaton induced by the *mixed* ω-algebraic system (4.7); this new construction will

be defined after the example. Intuitively, the construction is similar to the old construction but it differentiates between variables x and z; it therefore uses the states $x_1, \ldots, x_n, z_1, \ldots, z_n$:



This simple ω-reset pushdown automaton has exactly the behavior $(\sigma, \omega^{(1)})$. This means, if only z_1 is Büchi-accepting, then the automaton does not allow the run $(ac^*a)^\omega$.

The rest of the chapter will show that in general, the l^{th} canonical solution of a mixed ω-algebraic system $x = p(x), z = \rho(x)z$ is exactly the behavior of the simple ω-reset pushdown automaton induced by $x = p(x), z = \rho(x)z$. ▽

Given a series $r \in S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\text{alg}}\langle\langle\Sigma^\omega\rangle\rangle$, we want to construct a simple ω-reset pushdown automaton with behavior r. By Theorem 3.14 and Theorem 3.5, r is a component of a canonical solution of an ω-algebraic system (4.8) (compare this to the algebraic system (4.3)) in Greibach normal form over the quemiring $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$,

$$\begin{aligned} y_i = &\sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i, ay_jy_k)ay_jy_k + \\ &\sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, ay_j)ay_j + \\ &\sum_{a\in\Sigma}(p_i, a)a\,. \end{aligned} \tag{4.8}$$

The variables of this system are y_i, $(1 \le i \le n)$; they are variables for $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$. The system (4.8) induces the following mixed ω-algebraic system:

$$\begin{aligned} x_i = &\sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i, ay_jy_k)ax_jx_k + \\ &\sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, ay_j)ax_j + \\ &\sum_{a\in\Sigma}(p_i, a)a\,. \end{aligned} \tag{4.3}$$

and

$$z_i = \sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i, ay_jy_k)a(z_j + x_jz_k) + \sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, ay_j)az_j \tag{4.9}$$

But this system hides information, for instance, $y_j y_k$ will never be derived by two consecutive variables $z_j z_k$ of $S\langle\langle\Sigma^\omega\rangle\rangle$. Our new construction is therefore based on the following mixed ω-algebraic system that can be gained from the last system (4.3), (4.9) by renaming:

$$\begin{aligned} x_i = &\sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i, ax_jx_k)ax_jx_k + \\ &\sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, ax_j)ax_j + \\ &\sum_{a\in\Sigma}(p_i, a)a\,. \end{aligned} \tag{4.3}$$

and

$$z_i = \sum_{1\le j,k\le n}\sum_{a\in\Sigma}(p_i, ax_jz_k)ax_jz_k + \sum_{1\le j\le n}\sum_{a\in\Sigma}(p_i, az_j)az_j \tag{4.10}$$

We now want to construct a simple ω-reset pushdown automaton. Here, we introduce our new construction. Let $\mathfrak{A}^l_m = (2n+1, \Gamma, I_m, M, P, l)$, $1 \le m \le n$, $0 \le l \le n$, be defined as follows:
We let $\Gamma = \{X_1, \ldots, X_n, Z_1, \ldots, Z_n\}$; we denote the states $1, \ldots, 2n+1$ by $z_1, \ldots, z_n$, $x_1, \ldots, x_n, f$; the entries of M of the form $(M_{\pi,\pi'})_{v,v'}$ for $1 \le v, v' \le 2n+1$ and for $\pi, \pi' \in \Gamma^*$ with $|\pi|, |\pi'| \le 1$ that may be unequal to 0 are

$$\begin{aligned} (M_{\epsilon,X_k})_{x_i,x_j} &= \sum_{a\in\Sigma}(p_i, ax_jx_k)a\,, \\ (M_{Z_k,Z_k})_{x_i,x_j} = (M_{X_k,X_k})_{x_i,x_j} = (M_{\epsilon,\epsilon})_{x_i,x_j} &= \sum_{a\in\Sigma}(p_i, ax_j)a\,, \\ (M_{Z_k,\epsilon})_{x_i,z_k} = (M_{X_k,\epsilon})_{x_i,x_k} = (M_{Z_k,Z_k})_{x_i,f} = (M_{X_k,X_k})_{x_i,f} = (M_{\epsilon,\epsilon})_{x_i,f} &= \sum_{a\in\Sigma}(p_i, a)a\,, \\ (M_{Z_k,Z_k})_{z_i,z_j} = (M_{X_k,X_k})_{z_i,z_j} = (M_{\epsilon,\epsilon})_{z_i,z_j} &= \sum_{a\in\Sigma}(p_i, az_j)a\,, \\ (M_{\epsilon,Z_k})_{z_i,x_j} &= \sum_{a\in\Sigma}(p_i, ax_jz_k)a\,, \end{aligned}$$

for $1 \le i, j, k \le n$; we further put $(I_m)_{x_m} = (I_m)_{z_m} = \epsilon$, and $(I_m)_{x_i} = (I_m)_{z_i} = 0$ for $1 \le i \le m-1$ and $m+1 \le i \le n$ and $(I_m)_f = 0$; finally let $P_f = \epsilon$ and $P_j = 0$ for $1 \le j \le 2n$;

In the following, we assume that $r \in S^{\mathrm{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\mathrm{alg}}\langle\langle\Sigma^\omega\rangle\rangle$ is the m^{th} component of the l^{th} canonical solution of (4.8). We want to show that for the l^{th} canonical solution $\tau = (\sigma, \omega)$ of (4.3), (4.10), and therefore also of (4.8), we have $\tau_m = \sigma_m + \omega_m = \|\mathfrak{A}^l_m\|$.

This simple reset pushdown matrix M is called the simple reset pushdown matrix *induced* by the Greibach normal form (4.3), (4.10). The simple ω-reset pushdown automata $\mathfrak{A}^l_m$ ($1 \le m \le n$, $0 \le l \le n$) are called the simple ω-reset pushdown automata *induced* by the Greibach normal form (4.3), (4.10).

For the rest of the chapter, we will use the following notation (cf. Kuich and Salomaa, 1986, p. 179). Note that $M \in (S^{k\times k})^{\Gamma^*\times\Gamma^*}$ for $k = 2n+1$. By isomorphism, we can

transform this into $\widehat{M} \in (S^{\Gamma^* \times \Gamma^*})^{k \times k}$. We then have $(M_{\pi,\pi'})_{v,v'} = (\widehat{M}_{v,v'})_{\pi,\pi'}$ for $\pi, \pi' \in \Gamma^*$ and $1 \leq v, v' \leq 2n+1$. (By the notation $1 \leq v \leq 2n+1$, we mean v can be any of the states $z_1, \ldots, z_n, x_1, \ldots, x_n, f$.)

Example 4.24. This notation allows us to add up matrices with suitable pushdown indexes while still keeping the information of the states. For instance, note that

$$\sum_{1 \leq k \leq n} \sum_{\pi \in \Gamma^*} (\widehat{M}_{z_i,x_k})_{\epsilon,\pi} (\widehat{M}_{x_k,z_j})_{\pi,\epsilon} = \sum_{1 \leq k \leq n} \left(\widehat{M}_{z_i,x_k} \widehat{M}_{x_k,z_j}\right)_{\epsilon,\epsilon} .$$

Now consider the term

$$\sum_{1 \leq k \leq n} \sum_{\pi \in \Gamma^*} (M_{\epsilon,\pi})_{z_i,x_k} (M_{\pi,\epsilon})_{x_k,z_j} ,$$

which cannot be simplified because $\sum_{\pi \in \Gamma^*} (M_{\epsilon,\pi} M_{\pi,\epsilon})_{z_i,z_j}$ does no longer hold the information that the path passes only through states x_i, i.e., it contains also the path $(M_{\epsilon,\pi})_{z_i,z_k} (M_{\pi,\epsilon})_{z_k,z_j}$ (for all $1 \leq k \leq n$). In the proofs below, we will specifically need to distinguish paths that pass through states x_i and those that pass through states z_i as in the mixed ω-algebraic system, we also distinguish between variables x_i for finite derivations and variables z_i for infinite derivations. ▽

Lemma 4.25. *Let $M \in (S^{k \times k})^{\Gamma^* \times \Gamma^*}$ be a reset pushdown matrix. Then,*

$$\widehat{M^*} = \widehat{M}^* .$$

Proof. For $1 \leq v, v' \leq m$ and for $\pi, \pi' \in \Gamma^*$, we obtain

$$\begin{aligned}
((\widehat{M^*})_{v,v'})_{\pi,\pi'} &= ((M^*)_{\pi,\pi'})_{v,v'} \\
&= \sum_{n \geq 0} ((M^n)_{\pi,\pi'})_{v,v'} \\
&= \sum_{n \geq 0} ((\widehat{M}^n)_{v,v'})_{\pi,\pi'} \\
&= ((\widehat{M}^*)_{v,v'})_{\pi,\pi'} .
\end{aligned}$$

□

Similarly, we need the above result for another operator.

Lemma 4.26. *Let $M \in (S^{k \times k})^{\Gamma^* \times \Gamma^*}$ be a reset pushdown matrix. Then, for $1 \leq l \leq k$,*

$$\widehat{M^{\omega,l}} = \widehat{M}^{\omega,l} .$$

Proof. For $1 \leq v \leq k$ and for $\pi \in \Gamma^*$, we obtain

$$\begin{aligned}
((\widehat{M^{\omega,l}})_v)_\pi &= ((M^{\omega,l})_\pi)_v \\
&= \sum_{\pi_1, \pi_2, \ldots \in \Gamma^*} \sum_{(v_1, v_2, \ldots) \in P_l} (M_{\pi,\pi_1})_{v,v_1} (M_{\pi_1,\pi_2})_{v_1,v_2} (M_{\pi_2,\pi_3})_{v_2,v_3} \cdots \\
&= \sum_{\pi_1, \pi_2, \ldots \in \Gamma^*} \sum_{(v_1, v_2, \ldots) \in P_l} (\widehat{M}_{v,v_1})_{\pi,\pi_1} (\widehat{M}_{v_1,v_2})_{\pi_1,\pi_2} (\widehat{M}_{v_2,v_3})_{\pi_2,\pi_3} \cdots \\
&= ((\widehat{M}^{\omega,l})_v)_\pi .
\end{aligned}$$

□

Let M be a simple reset pushdown matrix induced by the Greibach normal form (4.3), (4.10). We define some blocks of the matrix $\widehat{M}$ to make the following argumentation easier. We take the idea of the above-mentioned isomorphism and divide $\widehat{M}$ like

$$\widehat{M} = \begin{pmatrix} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{pmatrix}, \tag{4.11}$$

where the respective blocks are defined as

$$\widehat{M}_{z,z} = \begin{pmatrix} \widehat{M}_{z_1,z_1} & \cdots & \widehat{M}_{z_1,z_n} \\ \vdots & \ddots & \vdots \\ \widehat{M}_{z_n,z_1} & \cdots & \widehat{M}_{z_n,z_n} \end{pmatrix}, \quad \widehat{M}_{z,x} = \begin{pmatrix} \widehat{M}_{z_1,x_1} & \cdots & \widehat{M}_{z_1,x_n} \\ \vdots & \ddots & \vdots \\ \widehat{M}_{z_n,x_1} & \cdots & \widehat{M}_{z_n,x_n} \end{pmatrix},$$

$$\widehat{M}_{x,z} = \begin{pmatrix} \widehat{M}_{x_1,z_1} & \cdots & \widehat{M}_{x_1,z_n} \\ \vdots & \ddots & \vdots \\ \widehat{M}_{x_n,z_1} & \cdots & \widehat{M}_{x_n,z_n} \end{pmatrix}, \quad \widehat{M}_{x,x} = \begin{pmatrix} \widehat{M}_{x_1,x_1} & \cdots & \widehat{M}_{x_1,x_n} \\ \vdots & \ddots & \vdots \\ \widehat{M}_{x_n,x_1} & \cdots & \widehat{M}_{x_n,x_n} \end{pmatrix}, \quad \widehat{M}_{x,f} = \begin{pmatrix} \widehat{M}_{x_1,f} \\ \vdots \\ \widehat{M}_{x_n,f} \end{pmatrix},$$

and where each $\widehat{M}_{v,v'} \in S^{\Gamma^* \times \Gamma^*}$ for $1 \leq v, v' \leq 2n+1$. For notational convenience, we also set

$$\widehat{M}_{z_i,x} = \begin{pmatrix} \widehat{M}_{z_i,x_1} & \cdots & \widehat{M}_{z_i,x_n} \end{pmatrix}, \qquad \widehat{M}_{x,z_i} = \begin{pmatrix} \widehat{M}_{x_1,z_i} \\ \vdots \\ \widehat{M}_{x_n,z_i} \end{pmatrix}.$$

Note that we have not defined the blocks $\widehat{M}_{z,f}$, $\widehat{M}_{f,z}$, $\widehat{M}_{f,x}$ and $\widehat{M}_{f,f}$ as they would all be zero by our construction for simple reset pushdown matrices induced by the Greibach normal form (4.3), (4.10).

Analogously, let $M_{z,z}, M_{z,x}, M_{x,z}, M_{x,x}, M_{x,f} \in (S^{(2n+1)\times(2n+1)})^{\Gamma^* \times \Gamma^*}$ be the isomorphic copy of $\widehat{M}_{z,z}, \widehat{M}_{z,x}, \widehat{M}_{x,z}, \widehat{M}_{x,x}, \widehat{M}_{x,f}$, respectively. Then, for $u, v \in \{x, z\}$ and for $\pi, \pi' \in \Gamma^*$, the matrix $(M_{u,v})_{\pi,\pi'}$ is $M_{\pi,\pi'}$ restricted to the variables u_i, v_j (for $1 \leq i, j \leq n$). Similarly, $M_{x,f}$ is M restricted to variables x_i, f (for $1 \leq i \leq n$). For instance, $(\widehat{M}_{x,x})^*$ and equally $(M_{x,x})^*$ consider only paths passing through states x_i and no paths through y_i or f (for $1 \leq i \leq n$). Their only difference is the order of indexes.

The following theorem computes the behavior of induced simple ω-reset pushdown automata.

Theorem 4.27. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10). *Then, for all $1 \leq i \leq n$ and $0 \leq l \leq n$,*

$$\left((M^{\omega,l})_\epsilon\right)_{x_i} = \left((M^{\omega,l})_\epsilon\right)_f = 0,$$

and

$$\left((M^{\omega,l})_\epsilon\right)_{z_i} = \left(\left(\left(M_{z,z} + M_{z,x}(M^*)_{x,x}M_{x,z}\right)^{\omega,l}\right)_\epsilon\right)_i.$$

Proof. For the matrix M, we have, by above notation (4.11),

$$\widehat{M} = \begin{pmatrix} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{pmatrix} = \left(\begin{array}{c|cc} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \hline \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{array}\right) .$$

Thus, by Theorem 3.2 and Lemma 4.26, we obtain

$$M^{\omega,l} = \begin{pmatrix} & & \alpha^{\omega,l} \\ \begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* & \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} & \alpha^{\omega,l} \end{pmatrix} ,$$

where

$$\begin{aligned} \alpha &= M_{z,z} + \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} \\ &= M_{z,z} + \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \begin{pmatrix} (M_{x,x})^* & (M_{x,x})^* M_{x,f} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} \\ &= M_{z,z} + \begin{pmatrix} M_{z,x}(M_{x,x})^* & M_{z,x}(M_{x,x})^* M_{x,f} \end{pmatrix} \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} \\ &= M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z} . \end{aligned} \tag{4.12}$$

Now, we continue with the term from before and get

$$\begin{aligned} M^{\omega,l} &= \begin{pmatrix} & & \alpha^{\omega,l} \\ \begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* & \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} & \alpha^{\omega,l} \end{pmatrix} \\ &= \begin{pmatrix} & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \\ \begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* & \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \end{pmatrix} \\ &= \begin{pmatrix} & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \\ \begin{pmatrix} (M_{x,x})^* & (M_{x,x})^* M_{x,f} \\ 0 & 1 \end{pmatrix} & \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \end{pmatrix} \\ &= \begin{pmatrix} & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \\ \begin{pmatrix} (M_{x,x})^* M_{x,z} \\ 0 \end{pmatrix} & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \end{pmatrix} \\ &= \begin{pmatrix} & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \\ ((M_{x,x})^* M_{x,z}) & & (M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z})^{\omega,l} \\ & 0 & \end{pmatrix} . \end{aligned}$$

Then, we start the run of the automaton with an empty stack and get

$$
\begin{aligned}
(M^{\omega,l})_\epsilon &= \begin{pmatrix} \left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l} \\ \left((M_{x,x})^* M_{x,z}\right)\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l} \\ 0 \end{pmatrix}_\epsilon \\
&= \begin{pmatrix} \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\epsilon \\ \left(\left((M_{x,x})^* M_{x,z}\right)\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\epsilon \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\epsilon \\ \sum_{\pi\in\Gamma^*} \left((M_{x,x})^* M_{x,z}\right)_{\epsilon,\pi} \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\pi \\ 0 \end{pmatrix} \\
&\overset{4}{=} \begin{pmatrix} \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\epsilon \\ \sum_{\pi\in\Gamma^*} 0 \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\pi \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} \left(\left(M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}\right)^{\omega,l}\right)_\epsilon \\ 0 \\ 0 \end{pmatrix}
\end{aligned}
$$

where the fourth equality uses the fact that $((M_{x,x})^* M_{x,z})_{\epsilon,\pi} = 0$ which is because $(M_{x,z})_{\pi,\pi'} = 0$ for all $\pi \neq Z_k\pi''$ ($1 \leq k \leq n$ and $\pi'' \in \Gamma^*$) and at the same time, $((M_{x,x})^*)_{\epsilon,Z_k\pi''} = 0$ because only $(M_{z,x})_{\epsilon,Z_k} \neq 0$ by construction.

The vector $(M^{\omega,l})_\epsilon$ is indexed by $z_1, \ldots, z_n, x_1, \ldots, x_n, f$, thus completing the proof. □

We want to apply the results from Section 4.1. The following three lemmas investigate the star operation applied to simple reset pushdown matrices M induced by the Greibach normal form (4.3), (4.10). The lemmas state that in a computation $(M^*)_{\epsilon,\epsilon}$, the new states y_k are never reached when starting in a state x_i and therefore, these computations are equivalent to the computations $(M'^*)_{\epsilon,\epsilon}$ for M' being induced by the Greibach normal form (4.3), i.e., for M' built by the old construction.

Lemma 4.28. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10). *Then, for all* $1 \leq i, k \leq n$,

$$((M^*)_{\epsilon,\epsilon})_{x_k,x_i} = (((M_{x,x})^*)_{\epsilon,\epsilon})_{x_k,x_i}\,.$$

Proof. Let $\Delta = \{X_1, \ldots, X_n\}$. We have

$$
\begin{aligned}
((M^*)_{\epsilon,\epsilon})_{x_k,x_i} &= \sum_{t\geq 0} ((M^t)_{\epsilon,\epsilon})_{x_k,x_i} \\
&= \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} \left(M_{\epsilon,\pi_1} M_{\pi_1,\pi_2} \cdots M_{\pi_{t-1},\epsilon}\right)_{x_k,x_i} \\
&= \sum_{t\geq 0} \sum_{\substack{\pi_1\in\Delta^* \\ \pi_2,\ldots,\pi_{t-1}\in\Gamma^*}} \sum_{1\leq j_1\leq n} (M_{\epsilon,\pi_1})_{x_k,x_{j_1}} \left(M_{\pi_1,\pi_2} \cdots M_{\pi_{t-1},\epsilon}\right)_{x_k,x_i}
\end{aligned}
$$

$$= \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Delta^*} \sum_{1\leq j_1,\ldots,j_{t-1}\leq n} (M_{\epsilon,\pi_1})_{x_k,x_{j_1}} (M_{\pi_1,\pi_2})_{x_{j_1},x_{j_2}} \cdots (M_{\pi_{t-1},\epsilon})_{x_{j_{t-1}},x_i}$$
$$= \big(\big(\sum_{t\geq 0} (M_{x,x})^t\big)_{\epsilon,\epsilon}\big)_{x_k,x_i} = (((M_{x,x})^*)_{\epsilon,\epsilon})_{x_k,x_i}\,,$$

where the third equality (and similarly the fourth equality) is by definition of induced pushdown matrices; the blocks $(M_{\epsilon,X_k})_{x_i,x_j}$, $(M_{X_k,X_k})_{x_i,x_j}$ and $(M_{\epsilon,\epsilon})_{x_i,x_j}$ are the only non-null blocks that describe a step in the matrix starting from a state x_i and having ϵ or X_k as the topmost stack symbol. □

Lemma 4.29. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10). *Then, we have*

$$((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^*)_{\epsilon,\epsilon} = ((M_{x,x})^*)_{\epsilon,\epsilon}\,.$$

Proof. Let $\Delta = \{X_1, \ldots, X_n\}$. In some sense similar to the proof of Lemma 4.28, we have

$$\begin{aligned}
&((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^*)_{\epsilon,\epsilon} \\
&= \Big(\sum_{t\geq 0} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^t\Big)_{\epsilon,\epsilon} \\
&= \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\epsilon,\pi_1} \cdots (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} \\
&= \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Gamma^*} \Big((M_{x,x})_{\epsilon,\pi_1} + \big(\sum_{\pi,\pi'\in\Gamma^*} (M_{x,z})_{\epsilon,\pi}((M_{z,z})^*)_{\pi,\pi'}(M_{z,x})_{\pi,\pi_1}\big)\Big) \\
&\qquad\qquad \cdots (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} \\
&\overset{4}{=} \sum_{t\geq 0} \sum_{\substack{\pi_1\in\Delta^* \\ \pi_2,\ldots,\pi_{t-1}\in\Gamma^*}} (M_{x,x})_{\epsilon,\pi_1}(M_{x,x}+M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_1,\pi_2}\cdots(M_{x,x}+M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} \\
&= \sum_{t\geq 0} \sum_{\substack{\pi_1\in\Delta^* \\ \pi_2,\ldots,\pi_{t-1}\in\Gamma^*}} (M_{x,x})_{\epsilon,\pi_1}\Big((M_{x,x})_{\pi_1,\pi_2} + \big(\sum_{\pi,\pi'\in\Gamma^*} (M_{x,z})_{\pi_1,\pi}((M_{z,z})^*)_{\pi,\pi'}(M_{z,x})_{\pi,\pi_2}\big)\Big) \\
&\qquad\qquad \cdots (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} \\
&\overset{6}{=} \sum_{t\geq 0} \sum_{\pi_1,\ldots,\pi_{t-1}\in\Delta^*} (M_{x,x})_{\epsilon,\pi_1}(M_{x,x})_{\pi_1,\pi_2}\cdots(M_{x,x})_{\pi_{t-1},\epsilon} = ((M_{x,x})^*)_{\epsilon,\epsilon}\,,
\end{aligned}$$

where the fourth equality is because $(M_{x,z})_{\epsilon,\pi} = 0$ for all $\pi \in \Gamma^*$. Similarly, for the sixth equality, we use the fact that $(M_{x,z})_{\pi_i,\pi} = 0$ for all $\pi_i \in \Delta^*$ (and $\pi \in \Gamma^*$). □

Lemma 4.30. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10) *and M' be induced by the Greibach normal form* (4.3). *Then, for all $1 \leq i \leq n$,*

$$((M^*)_{\epsilon,\epsilon})_{x_i,f} = ((M'^*)_{\epsilon,\epsilon})_{i,f}\,.$$

Proof. Note that by construction, we have

$$\widehat{M}' = \begin{pmatrix} \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 \end{pmatrix}.$$

By applying Lemma 4.25, we infer

$$M'^* = \begin{pmatrix} (M_{x,x})^* & (M_{x,x})^* M_{x,f} \\ 0 & 1 \end{pmatrix},$$

and we get

$$((M'^*)_{\epsilon,\epsilon})_{i,f} = \left(((M_{x,x})^* M_{x,f})_{\epsilon,\epsilon}\right)_i. \tag{4.13}$$

At the same time, we have

$$\begin{aligned} \widehat{M} &= \begin{pmatrix} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{pmatrix} \\ &= \left(\begin{array}{c|cc} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \hline \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{array}\right). \end{aligned}$$

By Lemma 4.25, we obtain

$$M^* = \begin{pmatrix} \alpha^* & \alpha^* \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \\ \beta^* \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} & \beta^* \end{pmatrix},$$

with

$$\begin{aligned} \alpha &= M_{z,z} + \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} \\ &= M_{z,z} + M_{z,x}(M_{x,x})^* M_{x,z}, \end{aligned}$$

by (4.12) in the proof of Theorem 4.27 and

$$\begin{aligned} \beta^* &= \left(\begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} (M_{z,z})^* \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \right)^* \\ &= \left(\begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} M_{x,z}(M_{z,z})^* M_{z,x} & 0 \\ 0 & 0 \end{pmatrix} \right)^* \\ &= \begin{pmatrix} M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x} & M_{x,f} \\ 0 & 0 \end{pmatrix}^* \\ &= \begin{pmatrix} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* & (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* M_{x,f} \\ 0 & 1 \end{pmatrix}. \end{aligned} \tag{4.14}$$

We deduce that

$$\begin{aligned}((M^*)_{\epsilon,\epsilon})_{x_i,f} &= \left(((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* M_{x,f}\right)_{\epsilon,\epsilon})_i \\ &= \left(((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^*)_{\epsilon,\epsilon}(M_{x,f})_{\epsilon,\epsilon}\right)_i \\ &= \left(((M_{x,x})^*)_{\epsilon,\epsilon}(M_{x,f})_{\epsilon,\epsilon}\right)_i \\ &= \left(((M_{x,x})^* M_{x,f})_{\epsilon,\epsilon}\right)_i \\ &= ((M'^*)_{\epsilon,\epsilon})_{i,f}\,,\end{aligned}$$

where the third equality is by Lemma 4.29 and the last equality is by (4.13). This concludes the proof. □

The following lemma investigates the final state f in infinite paths. It states that a finite run of induced simple ω-reset pushdown automata is equivalent to another path only through states x and with symbol Z_j initially on the pushdown tape and ending in state z_j with an empty pushdown tape.

Lemma 4.31. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10). *Then, for all $1 \leq j, k \leq n$,*

$$((M^*)_{\epsilon,\epsilon})_{x_k,f} = \left(((M_{x,x})^*)_{Z_j,Z_j} M_{Z_j,\epsilon}\right)_{x_k,z_j}.$$

Proof. The beginning of the proof is similar to the proof of Lemma 4.11. We obtain

$$\begin{aligned}((M^*)_{\epsilon,\epsilon})_{x_k,f} &= ((M^+)_{\epsilon,\epsilon})_{x_k,f} = ((M^*M)_{\epsilon,\epsilon})_{x_k,f} \\ &= \sum_{1\leq v_1\leq 2n+1} ((M^*)_{\epsilon,\epsilon})_{x_k,v_1}(M_{\epsilon,\epsilon})_{v_1,f} + \sum_{1\leq v_1\leq 2n+1}\sum_{P\in\Gamma} ((M^*)_{\epsilon,P})_{x_k,v_1}(M_{P,\epsilon})_{v_1,f} \\ &\overset{4}{=} \sum_{1\leq v_1\leq 2n+1} ((M^*)_{\epsilon,\epsilon})_{x_k,v_1}(M_{\epsilon,\epsilon})_{v_1,f} \\ &\overset{5}{=} \sum_{1\leq i\leq n} ((M^*)_{\epsilon,\epsilon})_{x_k,x_i}(M_{\epsilon,\epsilon})_{x_i,f} \\ &\overset{6}{=} \sum_{1\leq i\leq n} (((M_{x,x})^*)_{\epsilon,\epsilon})_{x_k,x_i}(M_{\epsilon,\epsilon})_{x_i,f} \\ &\overset{7}{=} \sum_{1\leq i\leq n} (((M_{x,x})^*)_{Z_j,Z_j})_{x_k,x_i}(M_{Z_j,\epsilon})_{x_i,z_j} = (((M_{x,x})^*)_{Z_j,Z_j} M_{Z_j,\epsilon})_{x_k,z_j}\,,\end{aligned}$$

where the fourth equality is since $(M_{P,\epsilon})_{v_1,f} = 0$ for all $1 \leq v_1 \leq 2n+1$ and $P \in \Gamma$ by our construction. In the fifth equality, we use the fact that $(M_{\epsilon,\epsilon})_{v_1,f} = 0$ for $v_1 \neq x_i$ $(1 \leq i \leq n)$. The sixth equality is by Lemma 4.28. The seventh equality is also by construction and by the definition of pushdown matrices. □

The following lemma treats a case similar to the previous lemma. It states that an infinite path starting with symbol Z_k on the pushdown tape is equivalent to a finite run starting with an empty pushdown tape and ending in state f followed by an infinite run that starts in state z_k with an empty pushdown tape.

Lemma 4.32. *Let M be a simple reset pushdown matrix induced by the Greibach normal form* (4.3), (4.10)*. Then, for all $1 \leq j,k \leq n$ and $0 \leq l \leq n$,*

$$((M^{\omega,l})_{Z_k})_{x_j} = ((M^*)_{\epsilon,\epsilon})_{x_j,f}((M^{\omega,l})_\epsilon)_{z_k}\,.$$

Proof. By Lemma 4.20, we have

$$\begin{aligned}((M^{\omega,l})_{Z_k})_{x_j} &= \left[(M^{\omega,l})_\epsilon + (M^*)_{\epsilon,\epsilon}M_{Z_k,\epsilon}(M^{\omega,l})_\epsilon\right]_{x_j} \\ &= \left((M^{\omega,l})_\epsilon\right)_{x_j} + \left((M^*)_{\epsilon,\epsilon}M_{Z_k,\epsilon}(M^{\omega,l})_\epsilon\right)_{x_j}.\end{aligned}$$

Consider the first summand. By Theorem 4.27, we know that

$$((M^{\omega,l})_\epsilon)_{x_j} = 0\,.$$

Now, we consider the second summand. For $1 \leq j,k \leq n$, we have

$$\begin{aligned}\left((M^*)_{\epsilon,\epsilon}M_{Z_k,\epsilon}(M^{\omega,l})_\epsilon\right)_{x_j} &= \sum_{1\leq v_1,v_2\leq 2n+1}((M^*)_{\epsilon,\epsilon})_{x_j,v_1}(M_{Z_k,\epsilon})_{v_1,v_2}((M^{\omega,l})_\epsilon)_{v_2} \\ &= \sum_{1\leq v_1\leq 2n+1}((M^*)_{\epsilon,\epsilon})_{x_j,v_1}(M_{Z_k,\epsilon})_{v_1,z_k}((M^{\omega,l})_\epsilon)_{z_k} \\ &= \sum_{1\leq v_1\leq 2n+1}((M^*)_{\epsilon,\epsilon})_{x_j,v_1}(M_{\epsilon,\epsilon})_{v_1,f}((M^{\omega,l})_\epsilon)_{z_k} \\ &= ((M^*)_{\epsilon,\epsilon}M_{\epsilon,\epsilon})_{x_j,f}((M^{\omega,l})_\epsilon)_{z_k} \\ &= ((M^*)_{\epsilon,\epsilon})_{x_j,f}((M^{\omega,l})_\epsilon)_{z_k}\,,\end{aligned}$$

where the second equality holds because we defined $(M_{Z_k,\epsilon})_{v_1,v_2} = 0$ for $v_2 \neq z_k$ and the third equality is because we have $(M_{Z_k,\epsilon})_{v_1,z_k} = (M_{\epsilon,\epsilon})_{v_1,f}$ for induced simple pushdown matrices. The result follows. □

We now discuss the behaviors of our constructed simple ω-reset pushdown automata.

Lemma 4.33. *Let the simple ω-reset pushdown automata $\mathfrak{A}^l_m = (2n+1, \Gamma, I_m, M, P, l)$, for $1 \leq m \leq n$ and $0 \leq l \leq n$, be induced by the Greibach normal form* (4.3), (4.10)*. We then have*

$$\|\mathfrak{A}^l_m\| = ((M^*)_{\epsilon,\epsilon})_{x_m,f} + ((M^{\omega,l})_\epsilon)_{z_m}\,.$$

Proof. Let $1 \leq m \leq n$ and $0 \leq l \leq n$. We obtain

$$\begin{aligned}\|\mathfrak{A}^l_m\| &= I(M^*)_{\epsilon,\epsilon}P + I(M^{\omega,l})_\epsilon \\ &= ((M^*)_{\epsilon,\epsilon})_{x_m,f} + ((M^*)_{\epsilon,\epsilon})_{z_m,f} + ((M^{\omega,l})_\epsilon)_{x_m} + ((M^{\omega,l})_\epsilon)_{z_m}\,, \\ &= ((M^*)_{\epsilon,\epsilon})_{x_m,f} + ((M^*)_{\epsilon,\epsilon})_{z_m,f} + ((M^{\omega,l})_\epsilon)_{z_m}\,.\end{aligned}$$

where the last equality is by Theorem 4.27.

It remains to show that $((M^*)_{\epsilon,\epsilon})_{z_m,f} = 0$. We have

$$\widehat{M} = \left(\begin{array}{c|cc} \widehat{M}_{z,z} & \widehat{M}_{z,x} & 0 \\ \hline \widehat{M}_{x,z} & \widehat{M}_{x,x} & \widehat{M}_{x,f} \\ 0 & 0 & 0 \end{array}\right).$$

Now let

$$M^* = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix},$$

where we are only interested in the second component of β. By lemma 4.25 and by (4.14) in the proof of Lemma 4.30, we have

$$\begin{aligned}
\beta &= (M_{z,z})^* \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \left[\begin{pmatrix} M_{x,x} & M_{x,f} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} M_{x,z} \\ 0 \end{pmatrix} (M_{z,z})^* \begin{pmatrix} M_{z,x} & 0 \end{pmatrix} \right]^* \\
&= \begin{pmatrix} (M_{z,z})^* M_{z,x} & 0 \end{pmatrix} \begin{pmatrix} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* & (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* M_{x,f} \\ 0 & 1 \end{pmatrix} \\
&= \Big((M_{z,z})^* M_{z,x} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^*, \\
&\qquad (M_{z,z})^* M_{z,x} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* M_{x,f} \Big).
\end{aligned}$$

Now, we obtain

$$\begin{aligned}
&((M^*)_{\epsilon,\epsilon})_{z_m,f} \\
&= \Big(\Big((M_{z,z})^* M_{z,x} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* M_{x,f} \Big)_{\epsilon,\epsilon} \Big)_m \\
&= \Big(\big((M_{z,z})^* M_{z,x} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* \big)_{\epsilon,\epsilon} (M_{x,f})_{\epsilon,\epsilon} \Big)_m \\
&= \Big(((M_{z,z})^*)_{\epsilon,\epsilon} \big(M_{z,x} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* \big)_{\epsilon,\epsilon} (M_{x,f})_{\epsilon,\epsilon} \Big)_m \\
&= \sum_{1 \le i \le n} \Big(((M_{z,z})^*)_{\epsilon,\epsilon} (M_{z,x})_{\epsilon,Z_i} \big((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* \big)_{Z_i,\epsilon} (M_{x,f})_{\epsilon,\epsilon} \Big)_m, \quad (4.15)
\end{aligned}$$

where in the second equality, we have $(M_{x,f})_{\pi,\epsilon} = 0$ for $\pi \neq \epsilon$. The third equality uses that $(M_{z,z})^*)_{\epsilon,\pi} = 0$ for $\pi \neq \epsilon$. In the fourth equality, we have $(M_{z,x})_{\epsilon,\pi} = 0$ for $\pi \notin \{Z_i \mid 1 \le i \le n\}$.

We concentrate on the factor in the center, where we have

$$\begin{aligned}
&\big((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^* \big)_{Z_i,\epsilon} \\
&= \sum_{t \ge 0} \sum_{\pi_1,\ldots,\pi_{t-1} \in \Gamma^*} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{Z_i,\pi_1} \cdots (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} \\
&= \sum_{t \ge 0} \sum_{\pi_1,\ldots,\pi_{t-2} \in \Gamma^*} (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{Z_i,\pi_1} \cdots (M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-2},\epsilon} (M_{x,x})_{\epsilon,\epsilon} \\
&= \sum_{t \ge 0} (M_{x,x})_{Z_i,\epsilon} \cdots (M_{x,x})_{\epsilon,\epsilon} (M_{x,x})_{\epsilon,\epsilon} = 0,
\end{aligned}$$

where in the second (and similarly in the third) equality we have $(M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} = (M_{x,x})_{\epsilon,\epsilon}$ because $(M_{x,x})_{\pi_{t-1},\epsilon} = 0$ for $\pi_{t-1} \neq \epsilon$ and because

$$(M_{x,z}(M_{z,z})^* M_{z,x})_{\pi_{t-1},\epsilon} = \sum_{\pi,\pi' \in \Gamma^*} (M_{x,z})_{\pi_{t-1},\pi}((M_{z,z})^*)_{\pi,\pi'}(M_{z,x})_{\pi',\epsilon} = 0$$

as $(M_{z,x})_{\pi',\epsilon} = 0$ for all π'. In the last equality, $(M_{x,x})_{Z_i,\epsilon} = 0$.

We now plug this into (4.15) and obtain

$$\begin{aligned}
&((M^*)_{\epsilon,\epsilon})_{z_m,f} \\
&\quad = \sum_{1\leq i\leq n} \Big(((M_{z,z})^*)_{\epsilon,\epsilon}(M_{z,x})_{\epsilon,Z_i} \big((M_{x,x} + M_{x,z}(M_{z,z})^* M_{z,x})^*\big)_{Z_i,\epsilon}(M_{x,f})_{\epsilon,\epsilon} \Big)_m \\
&\quad = \sum_{1\leq i\leq n} \Big(((M_{z,z})^*)_{\epsilon,\epsilon}(M_{z,x})_{\epsilon,Z_i} 0 (M_{x,f})_{\epsilon,\epsilon} \Big)_m = 0\,.
\end{aligned}$$

This completes the proof. □

The following theorem compares the behavior of induced simple ω-reset pushdown automata with the solutions of system (4.8).

Theorem 4.34. *Let (S, V) be a complete semiring-semimodule pair. Let the simple ω-reset pushdown automata $\mathfrak{A}^l_m$, for $1 \leq m \leq n$ and $0 \leq l \leq n$, be induced by the Greibach normal form (4.3), (4.10).*

Then, for $0 \leq l \leq n$,

$$(\|\mathfrak{A}^l_1\|, \ldots, \|\mathfrak{A}^l_n\|) = \big(((M^*)_{\epsilon,\epsilon})_{x_1,f} + ((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f} + ((M^{\omega,l})_\epsilon)_{z_n}\big)$$

is a solution of (4.8).

Proof. We show that

$$(((M^*)_{\epsilon,\epsilon})_{x_1,f}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f}) \qquad \text{and} \qquad (((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^{\omega,l})_\epsilon)_{z_n})$$

are solutions of the mixed ω-algebraic system (4.3), (4.10).

Let now M' be induced by the Greibach normal form (4.3). By Theorem 4.12, $(((M'^*)_{\epsilon,\epsilon})_{1,f}, \ldots, ((M'^*)_{\epsilon,\epsilon})_{n,f})$ is a solution of (4.3). By Lemma 4.30, we deduce that $(((M^*)_{\epsilon,\epsilon})_{x_1,f}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f})$ is also a solution of (4.3).

We now show that $(((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^{\omega,l})_\epsilon)_{z_n})$ is a solution of (4.10) and substitute it into the right sides of (4.10):

$$\begin{aligned}
&\sum_{1\leq j,k\leq n} \sum_{a\in\Sigma} (p_i, a x_j z_k) a \sigma_j \omega_k + \sum_{1\leq j\leq n} \sum_{a\in\Sigma} (p_i, a z_j) a \omega_j \\
&= \sum_{1\leq j,k\leq n} (M_{\epsilon,Z_k})_{z_i,x_j} ((M^*)_{\epsilon,\epsilon})_{x_j,f} ((M^{\omega,l})_\epsilon)_{z_k} + \sum_{1\leq j\leq n} (M_{\epsilon,\epsilon})_{z_i,z_j} ((M^{\omega,l})_\epsilon)_{z_j} \\
&= \sum_{1\leq j,k\leq n} (M_{\epsilon,Z_k})_{z_i,x_j} ((M^{\omega,l})_{Z_k})_{x_j} + \sum_{1\leq j\leq n} (M_{\epsilon,\epsilon})_{z_i,z_j} ((M^{\omega,l})_\epsilon)_{z_j} \\
&= \sum_{1\leq k\leq n} (M_{\epsilon,Z_k}(M^{\omega,l})_{Z_k})_{z_i} + (M_{\epsilon,\epsilon}(M^{\omega,l})_\epsilon)_{z_i} \\
&= \big(M M^{\omega,l})_\epsilon\big)_{z_i} = ((M^{\omega,l})_\epsilon)_{z_i}, \qquad \text{for each } 1 \leq i \leq n\,,
\end{aligned}$$

where the second equality is by Lemma 4.32, the last equality is by Theorem 4.16. □

The theorem above, Theorem 4.34, is not sufficient for our main result. The following theorem extends the previous theorem by stating that $(\|\mathfrak{A}_1^l\|, \ldots, \|\mathfrak{A}_n^l\|)$ is a canonical solution of (4.8).

Theorem 4.35. *Let (S, V) be a complete semiring-semimodule pair. Let the simple ω-reset pushdown automata $\mathfrak{A}_m^l$, for $1 \leq m \leq n$ and $0 \leq l \leq n$, be induced by the Greibach normal form* (4.3), (4.10).

Then, for $0 \leq l \leq n$,

$$(\|\mathfrak{A}_1^l\|, \ldots, \|\mathfrak{A}_n^l\|) = \big(((M^*)_{\epsilon,\epsilon})_{x_1,f} + ((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f} + ((M^{\omega,l})_\epsilon)_{z_n}\big)$$

is the l^{th} canonical solution of (4.8).

Proof. We show that

$$(((M^*)_{\epsilon,\epsilon})_{x_1,f}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f}) \qquad \text{and} \qquad (((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^{\omega,l})_\epsilon)_{z_n})$$

is the l^{th} canonical solution of the mixed ω-algebraic system (4.3), (4.10).

Let M' be induced by the Greibach normal form (4.3). Then, by Theorem 4.12, $(((M'^*)_{\epsilon,\epsilon})_{1,f}, \ldots, ((M'^*)_{\epsilon,\epsilon})_{n,f})$ is the unique (and therefore least) solution of (4.3). By Lemma 4.30, we can conclude that $\sigma = (((M^*)_{\epsilon,\epsilon})_{x_1,f}, \ldots, ((M^*)_{\epsilon,\epsilon})_{x_n,f})$ is also the least solution of (4.3).

Fix l with $1 \leq l \leq n$ for the remainder of the proof. It remains to show that for the system (4.10), written as $z = \rho(x)z$, we have

$$\rho(\sigma)^{\omega,l} = (((M^{\omega,l})_\epsilon)_{z_1}, \ldots, ((M^{\omega,l})_\epsilon)_{z_n})$$

We start with the right side of equation (4.10). We have, for $1 \leq i \leq n$,

$$\begin{aligned}
\rho(\sigma)_i z &= \sum_{1\leq j,k\leq n} \sum_{a\in\Sigma} (p_i, ax_jz_k)a\sigma_j z_k + \sum_{1\leq j\leq n} \sum_{a\in\Sigma} (p_i, az_j)az_j \\
&= \sum_{1\leq j,k\leq n} \sum_{a\in\Sigma} (p_i, ax_kz_j)a\sigma_k z_j + \sum_{1\leq j\leq n} \sum_{a\in\Sigma} (p_i, az_j)az_j \\
&= \sum_{1\leq j\leq n} \Big(\sum_{1\leq k\leq n} \sum_{a\in\Sigma} (p_i, ax_kz_j)a\sigma_k + \sum_{a\in\Sigma} (p_i, az_j)a \Big) z_j \\
&= \sum_{1\leq j\leq n} \Big(\sum_{1\leq k\leq n} (M_{\epsilon,Z_j})_{z_i,x_k}((M^*)_{\epsilon,\epsilon})_{x_k,f} + (M_{\epsilon,\epsilon})_{z_i,z_j} \Big) z_j \\
&\overset{5}{=} \sum_{1\leq j\leq n} \Big(\sum_{1\leq k\leq n} (M_{\epsilon,Z_j})_{z_i,x_k}(((M_{x,x})^*)_{Z_j,Z_j} M_{Z_j,\epsilon})_{x_k,z_j} + (M_{\epsilon,\epsilon})_{z_i,z_j} \Big) z_j \\
&= \sum_{1\leq j\leq n} \Big(\sum_{1\leq k,k'\leq n} (M_{\epsilon,Z_j})_{z_i,x_k}(((M_{x,x})^*)_{Z_j,Z_j})_{x_k,x_{k'}} (M_{Z_j,\epsilon})_{x_{k'},z_j} + (M_{\epsilon,\epsilon})_{z_i,z_j} \Big) z_j \\
&= \sum_{1\leq j\leq n} \Big(\sum_{1\leq k,k'\leq n} (\widehat{M}_{z_i,x_k})_{\epsilon,Z_j}(((\widehat{M}_{x,x})^*)_{x_k,x_{k'}})_{Z_j,Z_j} (\widehat{M}_{x_{k'},z_j})_{Z_j,\epsilon} + (\widehat{M}_{z_i,z_j})_{\epsilon,\epsilon} \Big) z_j \\
&\overset{8}{=} \sum_{1\leq j\leq n} \Big(\sum_{1\leq k,k'\leq n} \sum_{P\in\Gamma} (\widehat{M}_{z_i,x_k})_{\epsilon,P}(((\widehat{M}_{x,x})^*)_{x_k,x_{k'}})_{P,P} (\widehat{M}_{x_{k'},z_j})_{P,\epsilon} + (\widehat{M}_{z_i,z_j})_{\epsilon,\epsilon} \Big) z_j
\end{aligned}$$

$$= \sum_{1\le j\le n} \Big(\sum_{1\le k,k'\le n} \big(\widehat{M}_{z_i,x_k}((\widehat{M}_{x,x})^*)_{x_k,x_{k'}}\widehat{M}_{x_{k'},z_j}\big)_{\epsilon,\epsilon} + (\widehat{M}_{z_i,z_j})_{\epsilon,\epsilon}\Big) z_j$$
$$= \sum_{1\le j\le n} \Big(\widehat{M}_{z_i,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_j} + \widehat{M}_{z_i,z_j}\Big)_{\epsilon,\epsilon} z_j\,,$$

where the fifth equality is by Lemma 4.31. The eighth equality is because for $P \neq Z_j$, we have $(\widehat{M}_{x_{k'},z_j})_{P,\epsilon} = 0$.

Now for ρ of the system $z = \rho(x)z$, we obtain

$$\rho(\sigma) = \begin{pmatrix} \big(\widehat{M}_{z_1,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_1} + \widehat{M}_{z_1,z_1}\big)_{\epsilon,\epsilon} & \cdots & \big(\widehat{M}_{z_1,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_n} + \widehat{M}_{z_1,z_n}\big)_{\epsilon,\epsilon} \\ \vdots & \ddots & \vdots \\ \big(\widehat{M}_{z_n,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_1} + \widehat{M}_{z_n,z_1}\big)_{\epsilon,\epsilon} & \cdots & \big(\widehat{M}_{z_n,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_n} + \widehat{M}_{z_n,z_n}\big)_{\epsilon,\epsilon} \end{pmatrix}$$
$$= \Big(M_{z,x}(M_{x,x})^*M_{x,z} + M_{z,z}\Big)_{\epsilon,\epsilon}.$$

Then, we have

$$\begin{aligned}
&\big(\rho(\sigma)^{\omega,l}\big)_j \\
&= \Big(\big((M_{z,x}(M_{x,x})^*M_{x,z} + M_{z,z}\big)_{\epsilon,\epsilon}\big)^{\omega,l}\Big)_j \\
&= \sum_{(j_1,j_2,\ldots)\in P_l} \big(\widehat{M}_{z_j,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_{j_1}} + \widehat{M}_{z_j,z_{j_1}}\big)_{\epsilon,\epsilon}\big(\widehat{M}_{z_{j_1},x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_{j_2}} + \widehat{M}_{z_{j_1},z_{j_2}}\big)_{\epsilon,\epsilon}\cdots \\
&\overset{4}{=} \sum_{(j_1,j_2,\ldots)\in P_l}\ \sum_{\pi_1,\pi_2,\ldots\in\Gamma^*} \big(\widehat{M}_{z_j,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_{j_1}} + \widehat{M}_{z_j,z_{j_1}}\big)_{\epsilon,\pi_1}\big(\widehat{M}_{z_{j_1},x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_{j_2}} + \widehat{M}_{z_{j_1},z_{j_2}}\big)_{\pi_1,\pi_2}\cdots \\
&= \Big(\big((M_{z,x}(M_{x,x})^*M_{x,z} + M_{z,z})^{\omega,l}\big)_\epsilon\Big)_j\,, \qquad (4.16)
\end{aligned}$$

where the fourth equality uses the fact that $\big(\widehat{M}_{z_i,x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_j} + \widehat{M}_{z_i,z_j}\big)_{\epsilon,\pi} = 0$ for $\pi \neq \epsilon$ which is because $(\widehat{M}_{z_i,z_j})_{\epsilon,\pi} = 0$ for $\pi \neq \epsilon$ by definition and because, by our construction, we have

$$M_{z,x}(M_{x,x})^*M_{x,z} = \sum_{1\le j\le n} (M_{z,x})_{\epsilon,Z_j}((M_{x,x})^*)_{Z_j,Z_j}(M_{x,z})_{Z_j,\epsilon}\,.$$

Inductively, the above argument can be applied to all factors $\big(\widehat{M}_{z_{j_i},x}(\widehat{M}_{x,x})^*\widehat{M}_{x,z_{j_{i+1}}} + \widehat{M}_{z_{j_i},z_{j_{i+1}}}\big)_{\pi_i,\pi_{i+1}}$ because we learn from the preceding factor that $\pi_i = \epsilon$.

Now, we proceed from the other direction. From Theorem 4.27, we know that for the simple ω-reset pushdown automaton $\mathfrak{A}^l_m$ and a variable z_j, we have

$$\begin{aligned}
((M^{\omega,l})_\epsilon)_{z_j} &= \Big(\big(\big(M_{z,z} + M_{z,x}(M_{x,x})^*M_{x,z}\big)^{\omega,l}\big)_\epsilon\Big)_j \\
&= \rho(\sigma)^{\omega,l}_j\,,
\end{aligned}$$

where the last equality is by (4.16). This completes the proof. □

We now combine our previous discussion and Theorem 4.35 to get our second main result.

Corollary 4.36. *Let S be a continuous star-omega semiring with the underlying semiring S being commutative and let $r \in S^{alg}\langle\langle\Sigma^*\rangle\rangle \times S^{alg}\langle\langle\Sigma^\omega\rangle\rangle$.*

Then there exists a simple ω-reset pushdown automaton with behavior r.

Proof. Let $r \in S^{\text{alg}}\langle\langle\Sigma^*\rangle\rangle \times S^{\text{alg}}\langle\langle\Sigma^\omega\rangle\rangle$. As discussed on page 81, by Theorem 3.14 (and Theorem 3.5), r is a component of a canonical solution of an ω-algebraic system in Greibach normal form over $S\langle\langle\Sigma^*\rangle\rangle \times S\langle\langle\Sigma^\omega\rangle\rangle$. Let (4.8) be such a system and assume that the m^{th} component of the l^{th} canonical solution of (4.8) is r, i.e., assume $\tau_m = r$ for the l^{th} canonical solution τ.

Now, we can construct the simple ω-reset pushdown automata $\mathfrak{A}_m^l$ induced by the Greibach normal form (4.3), (4.10), for which, by Theorem 4.35, $(\|\mathfrak{A}_1^l\|, \ldots, \|\mathfrak{A}_n^l\|)$ is the l^{th} canonical solution of (4.8). As the l^{th} canonical solution is unique, we can conclude that

$$\|\mathfrak{A}_m^l\| = \tau_m = r\,. \qquad \square$$

CHAPTER 5

Weighted Logic

This chapter is devoted to a logical characterization of weighted simple ω-pushdown automata. Here, we will extend the weighted automaton model that we used in Chapter 4 by using ω-valuation monoids as weight structure. They include complete semirings but also discounted and average behavior. Valuation monoids first appeared in Droste and Meinecke (2012) but their idea is based on Chatterjee, Doyen and Henzinger (2008). By an example, we show how a basic web server and its average response time for requests can be modeled by a simple ω-pushdown automaton with weights in a suitable ω-valuation monoid. This chapter contains a comparison of the weighted simple ω-pushdown automata defined in this chapter and the simple ω-reset pushdown automata in the preceding chapter.

Our first main result in this chapter is the expressive equivalence of Büchi and Muller acceptance for weighted simple ω-pushdown automata.

Then, we show several closure properties for our automaton model. Our second main result of this chapter is a Nivat-like decomposition theorem Nivat (1968) that shows that by the help of a morphism, we can express the behavior of every weighted simple ω-pushdown automaton as the intersection of an unweighted ω-pushdown automaton and a very simple ω-series. Nivat's theorem was extended to weighted automata of finite words over semirings by Droste and Kuske (to appear).

In this chapter, as the third main result, we extend the BET-Theorem (Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961) to weighted simple ω-pushdown automata. We extend the logic in Lautemann, Schwentick and Thérien (1994) and Droste and Dück (2017) and prove its equivalence to weighted simple ω-pushdown automata. For the proof, we do not reinvent the wheel but use the already existing BET-Theorem for weighted nested ω-word automata (Droste and Dück, 2017). The application of a projection allows us to lift the result on weighted nested ω-word automata to weighted simple ω-pushdown automata. We show how the quantitative behavior of the basic web server example mentioned above can be described in our weighted matching ω-MSO logic.

We structure the chapter as follows. We give basic definitions and compare Muller and Büchi acceptance in Section 5.1. Then, we prove several closure properties and finally, the Nivat-like result is in Section 5.3. The next Section 5.4 defines our weighted logic. For the convenience of the reader, Section 5.5 summarizes the known results about weighted nested ω-word languages and also shows the new projection. Finally, Section 5.6 proves the equivalence between the logic proposed in Section 5.4 and the automaton model defined in Section 5.1.

This chapter is based on Droste, Dziadek and Kuich (2020b).

5.1 ω-Valuation Monoids

This section gives basic definitions and introduces our weight structure, the ω-valuation monoids (introduced by Droste and Meinecke, 2012).

Recall from Chapter 3 (see page 36) that a monoid $(D, +, \mathbb{0})$ is called complete if it has infinitary sum operations (i) that are an extension of the finite sums and (ii) that are associative and commutative.

For a set D we denote by $C \subseteq_{\text{fin}} D$ that C is a finite subset of D. Let $(D_{\text{fin}})^\omega = \bigcup_{C \subseteq_{\text{fin}} D} C^\omega$.

Definition 5.1. An *ω-valuation monoid* $(D, +, \text{Val}^\omega, \mathbb{0})$ consists of a complete monoid $(D, +, \mathbb{0})$ and an ω-valuation function $\text{Val}^\omega \colon (D_{\text{fin}})^\omega \to D$ such that $\text{Val}^\omega(d_i)_{i\in\mathbb{N}} = \mathbb{0}$ whenever $d_i = \mathbb{0}$ for some $i \in \mathbb{N}$. ▼

A monoid $(D, +, \mathbb{0})$ is called *idempotent* if $d + d = d$ for all $d \in D$. An ω-valuation monoid $(D, +, \text{Val}^\omega, \mathbb{0})$ is equally called *idempotent* if its underlying monoid $(D, +, \mathbb{0})$ is idempotent.

A *product ω-valuation monoid* (*ω-pv-monoid*) is a tuple $(D, +, \text{Val}^\omega, \diamond, \mathbb{0}, \mathbb{1})$ where $(D, +, \text{Val}^\omega, \mathbb{0})$ is an ω-valuation monoid, $\diamond \colon D^2 \to D$ is a product function and further $\mathbb{1} \in D$, $Val^\omega(\mathbb{1}^\omega) = \mathbb{1}$ and $\mathbb{0} \diamond d = d \diamond \mathbb{0} = \mathbb{0}$, $\mathbb{1} \diamond d = d \diamond \mathbb{1} = d$ for all $d \in D$.

Example 5.2 (**ω-valuation monoids**)**.** The first two examples are inspired by Chatterjee, Doyen and Henzinger (2008).

1. Let $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$ and let $-\infty + \infty = -\infty$. Then $(\bar{\mathbb{R}}, \sup, \limavg, +, -\infty, 0)$ is an ω-pv-monoid where
$$\limavg(d_i)_{i\in\mathbb{N}} = \liminf_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} d_i \,.$$
2. Let $\bar{\mathbb{R}}_+ = \{x \in \mathbb{R} \mid x \geq 0\} \cup \{-\infty\}$. Then $(\bar{\mathbb{R}}_+, \sup, \text{disc}_\lambda, +, -\infty, 0)$ for $0 < \lambda < 1$ is an ω-pv-monoid where
$$\text{disc}_\lambda(d_i)_{i\in\mathbb{N}} = \lim_{n\to\infty} \sum_{i=0}^{n} \lambda^i d_i \,.$$
3. By taking the infinite product as ω-valuation function, we get that every complete star-omega semiring $(S, \oplus, \otimes, \mathbb{0}, \mathbb{1})$ is an ω-pv-monoid $(S, \oplus, \bigotimes, \otimes, \mathbb{0}, \mathbb{1})$. ▽

The following definitions are taken from Droste and Dück (2017) and Droste and Meinecke (2012). Let $(D, +, \text{Val}^\omega, \diamond, \mathbb{0}, \mathbb{1})$ be an ω-pv-monoid. We call D *associative* (or *commutative*) if $\diamond$ is associative (or commutative). The ω-pv-monoid D is called *left-+-distributive* if for all $d \in D$, for any index set I and $(d_i)_{i\in I} \in D^I$:
$$d \diamond \sum_{i\in I} d_i = \sum_{i\in I} (d \diamond d_i) \,.$$

Analogously, we define *right-+-distributivity*. D is *+-distributive* if D is left- and right-+-distributive. We call D *left-Val$^\omega$-distributive* if for all $d \in D$ and $(d_i)_{i\in\mathbb{N}} \in D^\omega$:

$$d \diamond \mathrm{Val}^\omega((d_i)_{i\in\mathbb{N}}) = \mathrm{Val}^\omega((d \diamond d_i)_{i\in\mathbb{N}})\,.$$

D is called *left-multiplicative* if for all $d \in D$ and $(d_i)_{i\in\mathbb{N}} \in D^\omega$:

$$d \diamond \mathrm{Val}^\omega((d_i)_{i\in\mathbb{N}}) = \mathrm{Val}^\omega(d \diamond d_0, (d_i)_{i\geq 1})\,.$$

We further call D *conditionally commutative* if we have for all $(d_i)_{i\in\mathbb{N}}, (d'_i)_{i\in\mathbb{N}} \in D^\omega$ with $d_i \diamond d'_j = d'_j \diamond d_i$ for all $j < i$,

$$\mathrm{Val}^\omega((d_i)_{i\in\mathbb{N}}) \diamond \mathrm{Val}^\omega((d'_i)_{i\in\mathbb{N}}) = \mathrm{Val}^\omega((d_i \diamond d'_i)_{i\in\mathbb{N}})\,.$$

The ω-pv-monoid D is called *left-distributive* if D is left-+-distributive and either left-Val$^\omega$-distributive or left-multiplicative. If D is +-distributive and associative then $(D, +, \diamond, 0, 1)$ is a complete semiring and we call $(D, +, \mathrm{Val}^\omega, \diamond, 0, 1)$ an *ω-valuation semiring*. A *cc-ω-valuation semiring* is an ω-valuation semiring D that is conditionally commutative and left-distributive.

Example 5.3. We discuss here the properties of the ω-pv-monoids of Example 5.2:

1. $(\bar{\mathbb{R}}, \sup, \limsup, +, -\infty, 0)$ is left-distributive but not conditionally commutative,
2. $(\bar{\mathbb{R}}_+, \sup, \mathrm{disc}_\lambda, +, -\infty, 0)$ is a left-multiplicative cc-ω-valuation semiring,
3. $(S, \oplus, \bigotimes, \otimes, \mathbb{0}, \mathbb{1})$ is a cc-ω-valuation semiring. ▽

5.2 Weighted Simple ω-Pushdown Automata

We will now introduce the weighted automata we want to discuss in this chapter. At the end of this section, we give our first main result, the comparison of Muller and Büchi acceptance. As it simplifies the logical characterization, we follow the same approach as in Chapter 2 (cf. Droste and Perevoshchikov, 2015a; Droste, Dziadek and Kuich, 2020a) and use a restricted type of pushdown automaton. We call it *simple ω-pushdown automaton*. For the unweighted setting, we proved in Chapter 2 that this automaton model is expressively equivalent to general ω-pushdown automata; for finite words, this equivalence is hidden in a proof by Blass and Gurevich (2006). For continuous commutative star-omega semirings we could show in Chapter 4 that for every ω-algebraic series r, there exists a simple ω-reset pushdown automaton with behavior r (Droste, Dziadek and Kuich, 2019a,b, 2020c). A comparison of simple reset pushdown automata and weighted simple pushdown automata is given in this section.

Simple ω-pushdown automata are realtime, i.e. they do not use ϵ-transitions. Additionally, we restrict transitions in a way to only allow either to keep the stack unaltered, to push one symbol or to pop one symbol. Thus, let $\mathcal{S}(\Gamma) = (\{\downarrow\} \times \Gamma) \cup \{\#\} \cup (\{\uparrow\} \times \Gamma)$ be the set of *stack commands* for a stack alphabet Γ (exactly as in Chapter 2). Note that this implies that the automaton can only read the top of the stack when popping it.

Additionally, for technical reasons, we start runs with an empty stack and therefore allow to push onto the empty stack.

We will use Definition 2.6 of an ωPDA from Chapter 2.

Definition 5.4. A *weighted* (*simple*) *ω-pushdown automaton* (ωWPDA) over the alphabet Σ and the ω-valuation monoid $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ is a tuple $M = (Q, \Gamma, T, I, F, \mathrm{wt})$ where

- (Q, Γ, T, I, F) is an unweighted ω-pushdown automaton (ωPDA) over Σ,
- $\mathrm{wt}\colon T \to D$ is a weight function. ▼

Definition 5.5. A *Muller-accepting* ω-pushdown automaton over the alphabet Σ is a tuple $M = (Q, \Gamma, T, I, \mathcal{F})$ where Q, Γ, T, I are defined as for ωPDA, but $\mathcal{F} \subseteq 2^Q$ is a set of Muller-accepting subsets of Q. Similarly, a weighted *Muller-accepting* ω-pushdown automaton over the alphabet Σ and the ω-valuation monoid D is a tuple $M = (Q, \Gamma, T, I, \mathcal{F}, \mathrm{wt})$. ▼

The following definitions are similar to the ones from Chapter 2.

A *configuration* of an ωWPDA is a pair (q, γ), where $q \in Q$ and $\gamma \in \Gamma^*$. We define the transition relation between configurations as follows. Let $\gamma \in \Gamma^*$ and $t \in T$. For $t = (q, a, q', (\downarrow, A))$, we write $(q, \gamma) \vdash_M^t (q', A\gamma)$. For $t = (q, a, q', \#)$, we write $(q, \gamma) \vdash_M^t (q', \gamma)$. Finally, for $t = (q, a, q', (\uparrow, A))$, we write $(q, A\gamma) \vdash_M^t (q', \gamma)$. These three types of transitions are called *push*, *internal* and *pop* transitions, respectively.

We denote by $\mathrm{label}(q, a, q', s) = a$ the *label* and by $\mathrm{state}(q, a, q', s) = q$ the *state* of a transition. Both, as well as the function wt will be extended to infinite sequences of transitions by letting $\mathrm{label}((t_i)_{i\geq 0}) = (\mathrm{label}(t_i))_{i\geq 0} \in \Sigma^\omega$ for the infinite word constructed from the labels and $\mathrm{state}((t_i)_{i\geq 0}) = (\mathrm{state}(q_i))_{i\geq 0} \in Q^\omega$ for the infinite sequence of states of the transitions and finally, $\mathrm{wt}((t_i)_{i\geq 0}) = (\mathrm{wt}(t_i))_{i\geq 0} \in D^\omega$ for the infinite sequence of transitions weights.

An infinite sequence of transitions $\rho = (t_i)_{i\geq 0}$ with $t_i \in T$ is called a *run* of the ωWPDA M on $w = \mathrm{label}(\rho)$ iff there exists an infinite sequence of configurations $(p_i, \gamma_i)_{i\geq 0}$ with $p_0 \in I$ and $\gamma_0 = \epsilon$ such that $(p_i, \gamma_i) \vdash_M^{t_i} (p_{i+1}, \gamma_{i+1})$ for each $i \geq 0$.

We abbreviate a run $\rho = (t_i)_{i\geq 0}$ with $(p_0, \gamma_0) \vdash_M^{t_0} (p_1, \gamma_1) \vdash_M^{t_1} \cdots$ where $\mathrm{label}(t_i) = a_i$ by $\rho\colon (p_0, \gamma_0) \xrightarrow{a_0} (p_1, \gamma_1) \xrightarrow{a_1} \cdots$ such that the word becomes visible.

For an infinite sequence of states $(q_i)_{i\geq 0}$, let $\mathrm{Inf}((q_i)_{i\geq 0}) = \{q \mid q = q_i \text{ for infinitely many } i \geq 0\}$ be the set of states that occur infinitely often. For Büchi-accepting automata, a run ρ is called *successful* if $\mathrm{Inf}(\mathrm{state}(\rho)) \cap F \neq \emptyset$. For Muller-accepting automata, a run ρ is called successful if $\mathrm{Inf}(\mathrm{state}(\rho)) \in \mathcal{F}$.

An ωWPDA M is called *unambiguous* if there exists at most one successful run of M on every word of the input alphabet.

For an ωWPDA M, we introduce the following function $\|M\|\colon \Sigma^\omega \to D$ which is called the *behavior* of M and which is defined by $\|M\|(w) = \sum(\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } w)$. If there is no successful run on w, then $\|M\|(w) = \mathbb{0}$.

Every series $s\colon \Sigma^\omega \to D$ which is the behavior of some ωWPDA over D is called *ωWPDA-recognizable*.

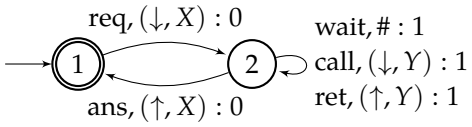


Figure 5.1: Example 5.6: Weighted ω-pushdown automaton over the alphabet $\Sigma = \{\text{req}, \text{ans}, \text{call}, \text{ret}, \text{wait}\}$ and the ω-valuation monoid $\bar{\mathbb{R}}$. The value after the ":" are the used weights 0 and 1.

An ωWPDA $M = (Q, \Gamma, T, I, F, \text{wt})$ that only uses internal transitions, i.e., for which $\Gamma = \emptyset$ and for all transitions $t = (q, a, q', s) \in T$ holds $s = \#$, is called a weighted finite ω-automaton, or short *ωWFA*.

Series that are the behavior of some ωWFA are called *ωWFA-recognizable*.

Example 5.6. We extend the ω-pv-monoid 1 of Example 5.2 as $(\bar{\mathbb{R}}, \sup, \text{specialavg}, +, -\infty, 0)$ where we define a new ω-valuation function to count and take the average of the counted values. Let h be a function that maps natural numbers to strings as follows.

$$\begin{aligned} h\colon \mathbb{N} &\to \{0,1\}^* \\ n &\mapsto 0\underbrace{11\ldots1}_{n\text{-times}}0 \end{aligned}$$

Then we extend h to infinite sequences of natural numbers $h\colon \mathbb{N}^\omega \to \{0,1\}^\omega$ in the natural way. We will consider its inverse where we have for instance

$$h^{-1}(01110011000011110\ldots) = 3204\ldots.$$

Then let $\text{specialavg} = \lim \text{avg} \circ h^{-1}$. For $w \notin (01^*0)^\omega$ we set $\text{specialavg}(w) = -\infty$.

Now, we define an automaton $\mathcal{A}$ as shown in Figure 5.1. We let $\mathcal{A} = (\{1,2\}, \{X,Y\}, T, \{1\}, \{1\}, \text{wt})$ be an ωWPDA over the alphabet $\Sigma = \{\text{req}, \text{ans}, \text{call}, \text{ret}, \text{wait}\}$, where T is defined as shown in the Figure and the weights are indicated after the colon symbol.

The automaton simulates some kind of (web) server that takes *req*uests from clients and *ans*wers them. For every request, the server has to *call* some amount of other services and a*wait* their *ret*urns. Only when all calls have been returned, the server answers the original request. This is a context-free property. Only runs that always eventually return to state 1 to serve new clients are considered valid.

Every call, return, or wait takes one second to operate and this operation time is accounted for in the weight. The specialavg operation sums up all the waiting time per request and returns the long run average response time. ▽

We now discuss how the ωWPDA defined in this chapter compare to the simple ω-reset pushdown automata in Chapter 4. The main difference is mainly a traditional one: The ω-algebraic systems combine a semiring and semimodule part because the semimodule part of their behavior depends on the semiring part. We therefore defined

ω-reset pushdown automata to also include this semiring part. But for a comparison here, we can safely ignore this component.

The second obvious difference is the weight structure. As given in the Examples 5.2 and 5.3, the complete star-omega semirings form a class of cc-ω-valuation semirings by taking the infinite product as the ω-valuation function. Therefore, we have to restrict ωWPDA to complete star-omega semirings for the following comparison.

The behavior of a simple ω-reset pushdown automaton $\mathfrak{A}$ is defined as

$$\|\mathfrak{A}\| = I(M^*)_{\epsilon,\epsilon} P + I(M^{\omega,l})_\epsilon \,.$$

Focusing on infinite words, we get $I(M^{\omega,l})_\epsilon$ as the interesting parts. In comparison, ωWPDA, as defined in this chapter, do equally start with an empty stack ϵ, and they follow the Büchi-acceptance conditions defined also for the operation ${}^{\omega,l}$.

However, ωWPDA do not use weights I for initial states. The missing weights of initial states can be mitigated as seen in the following lemma.

Lemma 5.7. *ωWPDA with additional initial weights are expressively equivalent to ωWPDA.*

We define an *ωWPDA with additional initial weights* over the alphabet Σ and the ω-valuation monoid $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ as a tuple $M = (Q, \Gamma, T, i, F, \mathrm{wt})$ similar to our original ωWPDA without initial weights but with

$$i \colon Q \to D \,.$$

We let the weight for one run

$$\rho \colon (p_0, \gamma_0) \vdash_M^{t_0} (p_1, \gamma_1) \vdash_M^{t_1} \dots$$

be defined as

$$\mathrm{Val}^\omega(\mathrm{wt}(\rho)) = \mathrm{Val}^\omega(i(p_0), (\mathrm{wt}(t_i))_{i \geq 0}) \,.$$

Proof. Let $M = (Q, \Gamma, T, i, F, \mathrm{wt})$ be an ωWPDA with additional initial weights. We construct an ωWPDA (without initial weights) $M' = (Q', \Gamma, T', I', F', \mathrm{wt}')$ with

$$\begin{aligned}
Q' &= Q \cup (Q \times T) \,, \\
F' &= F \times T \,, \\
I' &= \{q \in Q \mid i(q) \neq 0\} \,, \\
T' &= \{(p, a, (p', t'), s) \mid t' = (p, a, p', s) \in T\} \\
&\quad \cup \{((p, t), a, (p', t'), s) \mid t' = (p, a, p', s) \in T, t \in T\} \,, \\
\mathrm{wt}'(p, a, (p', t'), s) &= i(p) \,, \\
\mathrm{wt}'((p, t), a, (p', t'), s) &= \mathrm{wt}(t) \,.
\end{aligned}$$

The aim of the construction is to "shift" the weighs of M by one transition to the right.

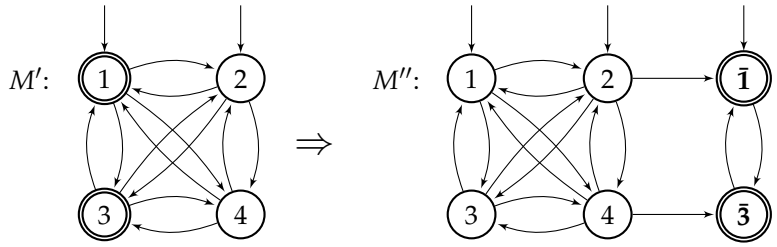


Figure 5.2: Proof of Theorem 5.9: The states 1, 2, 3, and 4 stand for the set of states that are initial and final, initial but not final, final but not initial, or neither initial nor final, respectively. Groups 1 and 3 are copied into $\bar{1}$ and $\bar{3}$. Transitions into $\bar{1}$ and $\bar{3}$ are only allowed from originally non-accepting states.

Consider a run ρ of M

$$\rho\colon (p_0,\gamma_0) \vdash_M^{t_0} (p_1,\gamma_1) \vdash_M^{t_1} (p_2,\gamma_2) \vdash_M^{t_2} \cdots,$$

with the weight

$$\mathrm{Val}^\omega(\mathrm{wt}(\rho)) = \mathrm{Val}^\omega(i(p_0),\ \mathrm{wt}(t_0),\quad \mathrm{wt}(t_1),\quad \mathrm{wt}(t_2),\ \ldots).$$

The new ωWPDA M' then has a corresponding run

$$\rho'\colon (p_0,\gamma_0) \vdash_{M'}^{t'_0} ((p_1,t_0),\gamma_1) \vdash_{M'}^{t'_1} ((p_2,t_1),\gamma_2) \vdash_{M'}^{t'_2} \cdots,$$

with the weight

$$\begin{aligned}\mathrm{Val}^\omega(\mathrm{wt}'(\rho')) &= \mathrm{Val}^\omega(i(p_0), \quad \mathrm{wt}(t_0), \quad \mathrm{wt}(t_1),\ \ldots)\\ &= \mathrm{Val}^\omega(\mathrm{wt}(\rho)).\end{aligned}$$

As we do not create new runs, our result follows. □

As a consequence of the preceding discussion, we obtain the following corollary.

Corollary 5.8. *ωWPDA over complete star-omega-semirings and simple ω-reset pushdown automata are expressively equivalent.*

We now state the first main result of this chapter.

Theorem 5.9. *Let $s\colon \Sigma^\omega \to D$ be a series. The following are equivalent:*

- *s is recognizable by a Büchi-accepting ωWPDA,*
- *s is recognizable by a Muller-accepting ωWPDA.*

Proof. There are two directions to show. The transformation of Büchi-accepting automata into Muller-accepting ones simply changes the acceptance condition. Take a Büchi-accepting ωWPDA $M = (Q,\Gamma,T,I,F,\mathrm{wt})$ and transform it into a Muller-accepting ωWPDA $M' = (Q,\Gamma,T,I,\mathcal{F},\mathrm{wt})$ with

$$\mathcal{F} = \{S \subseteq Q \mid S \cap F \neq \emptyset\}.$$

As exactly the same runs are successful in both automata, this approach works also in the weighted case.

The standard approach for transforming Muller-accepting automata into Büchi-accepting automata needs to be adjusted in the weighted case as it creates infinitely many possible runs in the Büchi automaton for every run in the Muller automaton. The construction usually creates a special set of accepting states that have to be traversed to be accepted. But in the weighted case, one has to be very careful to only allow one entry point into this special set of accepting states.

One solution to this problem was presented in Droste and Rahonis (2006). The new automaton then contains a group of accepting states for every $F \in \mathcal{F}$. Entering this group of accepting states is forbidden from a state that is already in F. In this way, the only successful runs are the ones that switch from the original states to the new group of accepting states at the last possible moment.

In contrast to Droste and Rahonis (2006), we cannot assume an initially normalized automaton to solve the remaining question of the initial states that are also final. Instead, the automaton decides non-deterministically if it will eventually see a non-final state in the run. If no, and only in this case, it already starts in the new group of accepting states.

Figure 5.2 depicts the idea of the construction.

Let $M = (Q, \Gamma, T, I, \mathcal{F}, \mathrm{wt})$ be a Muller-accepting ωWPDA. Now, we will construct an ωWPDA $M' = (Q', \Gamma, T', I', F', \mathrm{wt}')$ with $\|M'\| = \|M\|$.

We first observe that ωWPDA-recognizable series are closed under union. Indeed, taking the disjoint union of several automata $\mathcal{A}_i$ recognizes the sum of their recognized series $\sum \|\mathcal{A}_i\|$. We can therefore assume that $|\mathcal{F}| = 1$, i.e., $\mathcal{F} = \{F\}$ for $F \subseteq Q$.

We let $Q' = Q \cup (Q \times \mathcal{P}(Q))$, $F' = \{(q, F) \mid q \in F\}$ and $I' = I \cup \{(q, \{q\}) \mid q \in I\}$; by $\mathcal{P}(Q)$ we here mean the power set of Q. Furthermore, we let

$$\begin{aligned} T' = {} & \big\{\big(q, a, q', s\big) \mid (q, a, q', s) \in T\big\} \\ & \cup \big\{\big((q, R), a, (q', R \cup \{q'\}), s\big) \mid (q, a, q', s) \in T, q, q' \in F, R \subsetneq F\big\} \\ & \cup \big\{\big((q, F), a, (q', \{q'\}), s\big) \mid (q, a, q', s) \in T, q, q' \in F\big\} \\ & \cup \big\{\big(q, a, (q', \{q'\}), s\big) \mid (q, a, q', s) \in T, q \in Q \setminus F, q' \in F\big\}\,, \end{aligned}$$

and

$$\begin{aligned} \mathrm{wt}'(q, a, q', s) &= \mathrm{wt}(q, a, q', s) \\ \mathrm{wt}'(q, a, (q', R), s) &= \mathrm{wt}(q, a, q', s) \\ \mathrm{wt}'\big((q, R), a, (q', R), s\big) &= \mathrm{wt}(q, a, q', s)\,. \end{aligned}$$

This construction ensures that no run exists that switches from new states (q, R) back to original states q'. Furthermore it is forbidden to switch from q to (q', R) if $q \in F$. That means that there is one specific entry point into the set of new states (q, R) and every run entered this set will loop there. The first state in a run is allowed to be from the new set if and only if all states in the run are in F.

The subsets R stored in the second component of the new states (q, R) ensure that all states $p \in F$ are traversed (cf Droste and Rahonis (2006) for details). It follows that M' has the same amount of successful runs as M and as the weights are transferred one-to-one, we have $\|M'\| = \|M\|$. □

5.3 Closure Properties

Let Σ, Δ be alphabets and $h\colon \Sigma \to \Delta$ a mapping. We can extend h to infinite words in the natural way by setting $h(w) = h(a_0)h(a_1)h(a_2)\cdots \in \Delta^\omega$ for $w = a_0a_1a_2\cdots \in \Sigma^\omega$.

For a series $s\colon \Delta^\omega \to D$, we define the series $h^{-1}(s)\colon \Sigma^\omega \to D$ by $h^{-1}(s)(w) = s(h(w))$ for all $w \in \Sigma^\omega$.

Lemma 5.10. *Let D be an ω-valuation monoid, Σ, Δ two alphabets and $h\colon \Sigma \to \Delta$ a mapping. If $s\colon \Delta^\omega \to D$ is ωWPDA-recognizable, then so is $h^{-1}(s)\colon \Sigma^\omega \to D$.*

Proof. We follow the approach of Droste and Meinecke (2012). As s is ωWPDA-recognizable, there exists $M = (Q, \Gamma, T, I, F, wt)$ over Δ with behavior $\|M\| = s$. We construct $M' = (Q, \Gamma, T', I, F, wt')$ over Σ where

$$T' = \{(q, a, p, s) \mid (q, h(a), p, s) \in T\},$$
$$\mathrm{wt}'\big((q, a, p, s)\big) = \mathrm{wt}\big((q, h(a), p, s)\big).$$

As the weights are translated one-to-one, it holds:

$$\begin{aligned}\|M'\|(w) &= \sum(\mathrm{Val}^\omega(\mathrm{wt}'(\rho')) \mid \rho' \text{ successful run of } M' \text{ on } w)\\ &= \sum(\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } h(w))\\ &= \|M\|(h(w))\\ &= h^{-1}(\|M\|)(w)\end{aligned}$$ □

Let now $h\colon \Delta \to \Sigma$ and let $h^{-1}(w) = \{v \in \Delta^\omega \mid h(v) = w\}$. Then for a series $s\colon \Delta^\omega \to D$, we define the series $h(s)\colon \Sigma^\omega \to D$ by $h(s)(w) = \sum_{v \in h^{-1}(w)} s(v)$ for all $w \in \Sigma^\omega$.

Lemma 5.11. *Let Δ, Σ be alphabets, $(D, +, Val^\omega, \mathbb{0})$ an ω-valuation monoid and $h\colon \Delta \to \Sigma$ a mapping. If $s\colon \Delta^\omega \to D$ is ωWPDA-recognizable, then so is $h(s)\colon \Sigma^\omega \to D$.*

Proof. We follow the approach of Droste and Vogler (2010). As s is ωWPDA-recognizable, there exists $M = (Q, \Gamma, T, I, F, wt)$ over Δ with behavior $\|M\| = s$. Fix $a_0 \in \Delta$. We construct $M' = (Q', \Gamma, T', I', F', wt')$ over Σ where $Q' = Q \times \Delta$, $F' = F \times \Delta$ and $I' = I \times \{a_0\}$.

$$T' = \{((q, a), b, (p, a'), s) \mid b = h(a') \text{ and } (q, a', p, s) \in T\}$$
$$\mathrm{wt}'\big(((q, a), b, (p, a'), s)\big) = \mathrm{wt}\big((q, a', p, s)\big)$$

Note that in this construction, the new states contain letters $a' \in \Delta$ because otherwise, the function wt' would not be well-defined; it would not always be possible to guess the correct letter a' only from b because h is not necessarily injective.

Also note that we fixed a_0 to make sure that the first state is fixed and the amount of successful runs is not changed by our construction.

Let $b \in \Sigma^\omega$. Every successful run ρ' of M' on a word $b = (b_i)_{i \geq 1}$ is of the form

$$\rho' \colon ((q_0, a_0), \epsilon) \xrightarrow{b_1} ((q_1, a_1), \gamma_1) \xrightarrow{b_2} \dots$$

where $h(a_i) = b_i$ and by the above construction, every such run corresponds to runs ρ of M on words $a \in \Delta^\omega$ with $a \in h^{-1}(b)$ of the form

$$\rho \colon (q_0, \epsilon) \xrightarrow{a_1} (q_1, \gamma_1) \xrightarrow{a_2} \dots$$

Note that every run ρ' of M' can correspond to multiple runs ρ of M because h does not need to be injective. By definition, the weight of two corresponding runs is the same.

Finally, for a word $b \in \Sigma^\omega$,

$$\begin{aligned}
\|M'\|(b) &= \sum(\mathrm{Val}^\omega(\mathrm{wt}'(\rho')) \mid \rho' \text{ successful run of } M' \text{ on } b) \\
&= \sum(\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } a \text{ and } a \in h^{-1}(b)) \\
&= \sum_{a \in h^{-1}(b)} \|M\|(a) \\
&= h(\|M\|)(b)\,.
\end{aligned}$$

□

Let $g \colon \Sigma \to D$ be a mapping. We denote by $\mathrm{Val}^\omega \circ g \colon \Sigma^\omega \to D$ the series defined for all $w \in \Sigma^\omega$ by $(\mathrm{Val}^\omega \circ g)(w) = \mathrm{Val}^\omega(g(w))$.

Lemma 5.12. *Let Σ be an alphabet, $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ an ω-valuation monoid and $g \colon \Sigma \to D$ a mapping. Then $\mathrm{Val}^\omega \circ g$ is ωWFA-recognizable by an ωWFA with only one state.*

Proof. We construct an ωWFA $M = (Q, \Gamma, T, I, F, \mathrm{wt})$ over Σ with $Q = I = F = \{q_0\}$. Let further $T = \{(q_0, a, q_0, \#) \mid a \in \Sigma)\}$ and $\mathrm{wt}((q_0, a, q_0, \#)) = g(a)$. The stack will not be used, thus $\Gamma = \emptyset$.

Let $w = (a_i)_{i \geq 0} \in \Sigma^\omega$. For M, there is only one run r on w. This run is $r \colon (q_0, \epsilon) \xrightarrow{a_0} (q_0, \epsilon) \xrightarrow{a_1} \dots$. It follows, that r is successful and the behavior of M is

$$\begin{aligned}
\|M\|(w) &= \sum(\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } w) \\
&= \mathrm{Val}^\omega(\mathrm{wt}(r)) \\
&= \mathrm{Val}^\omega((\mathrm{wt}(q, a_i, q, \#))_{i \geq 0}) \\
&= \mathrm{Val}^\omega((g(a_i))_{i \geq 0}) \\
&= \mathrm{Val}^\omega(g(w)) \\
&= (\mathrm{Val}^\omega \circ g)(w)\,.
\end{aligned}$$

Hence $\mathrm{Val}^\omega \circ g$ is ωWFA-recognizable and the ωWFA M has only one state. □

Let $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ be an ω-valuation monoid, $s\colon \Sigma^\omega \to D$ an ωWFA-recognizable series and $L \subseteq \Sigma^\omega$ an ωPDA-recognizable language. By $s \cap L\colon \Sigma^\omega \to D$, we denote the series that assigns the weights of s to the words accepted by L. Formally, for words $u \in \Sigma^\omega$,

$$(s \cap L)(u) = \begin{cases} s(u), & \text{if } u \in L \\ \mathbb{0}, & \text{otherwise}. \end{cases}$$

Lemma 5.13. *Let $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ be an ω-valuation monoid, $s\colon \Sigma^\omega \to D$ an ωWFA-recognizable series and $L \subseteq \Sigma^\omega$ an ωPDA-recognizable language.*

1. *If L is unambiguous, then the series $s \cap L\colon \Sigma^\omega \to D$ is ωWPDA-recognizable.*
2. *If D is idempotent, then the series $s \cap L\colon \Sigma^\omega \to D$ is ωWPDA-recognizable.*

Proof. To allow final states of both original Büchi-accepting automata to be visited alternately, we use a standard construction for intersecting unweighted Büchi automata of infinite words (cf. Thomas, 1990). The proof that this construction also works in the weighted case is in some sense similar to Babari and Droste (2019).

Assume that $M = (Q, \emptyset, T, I, F, \mathrm{wt})$ is an ωWFA with $\mathcal{L}(M) = s$ and $M' = (Q', \Gamma', T', I', F')$ is an ωPDA with $\mathcal{L}(M') = L$. Note that M' does not contain ϵ-transitions by definition.

We construct the ωWPDA $\tilde{M} = (Q \times Q' \times \{0,1,2\}, \Gamma', \tilde{T}, I \times I' \times \{0\}), Q \times Q' \times \{2\}, \tilde{\mathrm{wt}})$, where

$$\begin{aligned}
\tilde{T} = &\{((q,q',0),a,(p,p',0),s') \mid (q,a,p,\#) \in T, (q',a,p',s') \in T', q \notin F\} \cup \\
&\{((q,q',0),a,(p,p',1),s') \mid (q,a,p,\#) \in T, (q',a,p',s') \in T', q \in F\} \cup \\
&\{((q,q',1),a,(p,p',1),s') \mid (q,a,p,\#) \in T, (q',a,p',s') \in T', q' \notin F'\} \cup \\
&\{((q,q',1),a,(p,p',2),s') \mid (q,a,p,\#) \in T, (q',a,p',s') \in T', q' \in F'\} \cup \\
&\{((q,q',2),a,(p,p',0),s') \mid (q,a,p,\#) \in T, (q',a,p',s') \in T'\}
\end{aligned}$$

and

$$\tilde{\mathrm{wt}}\big((q,q',i),a,(p,p',j),s'\big) = \mathrm{wt}(q,a,p,\#)\,.$$

As only the ωPDA M' uses the stack, the new transitions contain only the stack commands of T'.

This construction ensures that final states of M and final states of M' are visited infinitely often. From the other side, for every successful run $\rho = (q)_{i\in\mathbb{N}}$ of M and every successful run $\rho' = (q')_{i\in\mathbb{N}}$ of M', there is a new successful run $\tilde{\rho} = (q,q',k)_{i\in\mathbb{N}}$ of $\tilde{M}$, where k is alternating between 0,1 and 2. The weight of the runs ρ and $\tilde{\rho}$ are equal because the weights are directly transferred.

We consider two cases:

1. If L is unambiguous, then for every word $u \in \Sigma^\omega$, there is either one or no successful run of the ωPDA M' on u. Therefore, the resulting automaton $\tilde{M}$ has as many runs for a word u as the original ωWFA M had. As the weights of every run are not changed, for $u \in L$, we have $\|\tilde{M}\|(u) = \|M\|(u) = (s \cap L)(u)$.

2. Now let D be idempotent. For words $u \in \Sigma^\omega \setminus L$, there is no successful run of the ωPDA M' on u and therefore, $\|\tilde{M}\|(u) = \mathbb{0}$. Now let $n > 0$ be the number of successful runs of M' on $u \in L$. Then, $\|\tilde{M}\|(u) = \sum_{i=1}^{n} \|M\|(u)$ because every corresponding run of $\tilde{M}$ has the same weight. As D is idempotent, $\sum_{i=1}^{n} \|M\|(u) = \|M\|(u) = (s \cap L)(u)$. □

Note that closure under the more general intersection where both automata use the stack does not hold. Otherwise, over the Boolean semiring, $\{a^n b^n c^m d^\omega \mid n, m \in \mathbb{N}\}$ could be intersected with $\{a^n b^m c^m d^\omega \mid n, m \in \mathbb{N}\}$ to gain $\{a^n b^n c^n d^\omega \mid n \in \mathbb{N}\}$ which is not ωWPDA-recognizable anymore.

Intersection with inherently ambiguous languages over non-idempotent ω-valuation monoids might not be ωWPDA-recognizable:

Example 5.14. Let $(\mathbb{N}^\infty, +, \prod, 0)$ be an ω-pv-monoid where $+$ is the natural addition and $\prod$ is the infinite product and $\Sigma = \{a, b, c, d\}$. Let further $s(u) = 1$ for all $u \in \Sigma^\omega$ and

$$L = \{a^i b^j c^k d^\omega \mid i = j \text{ or } j = k\}.$$

L is inherently ambiguous because the finite language $\{a^i b^j c^k \mid i = j \text{ or } j = k\}$ is already inherently ambiguous. The series s is ωWPDA-recognizable by an automaton with only one state and one self-loop with weight 1. Then,

$$(s \cap L)(u) = \begin{cases} s(u), & \text{if } u \in L \\ 0, & \text{otherwise}, \end{cases} = \begin{cases} 1, & \text{if } u \in L \\ 0, & \text{otherwise}, \end{cases}$$

which is not ωWPDA-recognizable anymore because the ω-valuation monoid does not permit negative weights, instead, only one run with weight 1 is allowed. But for the words $a^n b^n c^n d^\omega$, multiple runs are necessary because L is inherently ambiguous, i.e., for a contradiction, consider an ωWPDA M for the series $(s \cap L)$. We have $(s \cap L)(u) = 1$ for every word $u \in L$ and we furthermore have $\|M\|(u) = \sum(\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } w)$. But the only sum in $\mathbb{N}^\infty$ yielding 1 is 1 itself plus possibly some zeros. When stripping M of its weights while only keeping transitions with non-zero weight, we can construct an unweighted pushdown automaton M'. The new automaton M' has only one successful run for every input $u \in L$ and is thus unambiguous; additionally, M' accepts the language L. This contradicts L being inherently ambiguous. ▽

Definition 5.15. Let Σ be an alphabet and $(D, +, \mathrm{Val}^\omega, \mathbb{0})$ an ω-valuation monoid.

We denote by $D^{\mathrm{rec}}\langle\langle\Sigma^\omega\rangle\rangle$ the family of ωWPDA-recognizable series over Σ and D. Let further $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle$ (with $\mathcal{N}$ meaning Nivat) denote the set of series $s\colon \Sigma^\omega \to D$ over D such that there exist an alphabet Δ, mappings $h\colon \Delta \to \Sigma$ and $r\colon \Delta \to D$ and an ωPDA-recognizable language $L \subseteq \Delta^\omega$ such that

$$s = h((\mathrm{Val}^\omega \circ r) \cap L).$$

We define $D^{\mathcal{N}}_{\text{UNAMB}}\langle\langle\Sigma^\omega\rangle\rangle$ like $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle$ with the difference that L is an unambiguous ωPDA-recognizable language. Finally, $D^{\mathcal{N}}_{\text{DET}}\langle\langle\Sigma^\omega\rangle\rangle$ is defined like $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle$ with the difference that L is a deterministic ωPDA-recognizable language. ▼

Example 5.16. We extend the ω-pv-monoid 1 of Example 5.2 as $(\dot{\mathbb{R}}, \sup, \text{partialavg}, +, -\infty, 0)$ where we add a new value d that will later be ignored, i.e., $\dot{\mathbb{R}} = \bar{\mathbb{R}} \cup \{d\}$. We set $\sup(-\infty, d) = d$ and $\sup(r, d) = r$ for every $r \in \mathbb{R}$. We define a new ω-valuation function to ignore d and take the average of the remaining values. Let h be a function that maps numbers $\dot{\mathbb{R}}$ to strings of numbers $\bar{\mathbb{R}}$ as follows.

$$h\colon \dot{\mathbb{R}} \to \bar{\mathbb{R}}^*, \qquad \begin{aligned} r &\mapsto r, \quad \text{for } r \in \bar{\mathbb{R}} \\ d &\mapsto \epsilon \end{aligned}$$

Then we extend h to infinite sequences $h\colon \dot{\mathbb{R}}^\omega \to \bar{\mathbb{R}}^\omega$ in the natural way. Now let $\text{partialavg} = \lim \text{avg} \circ h$.

Let $\Sigma = \{a, b\}$. We make the following definitions:

- $\Delta = \Sigma \times \{0, 1, \ldots, 6\}$,
- $L = \left\{(\sigma_1, d_1)(\sigma_2, d_2)(\sigma_3, d_3)\cdots \mid d_i = i \bmod 7,\ \sigma_i \in \Sigma\right\}$,
- $r(b, i) = d$ for all $i \in \{0, \ldots, 6\}$ and $r(a, i) = \begin{cases} 1, & \text{if } 5 \le i \le 6 \\ 0, & \text{otherwise}, \end{cases}$
- $h(\sigma, i) = \sigma$

The language $L \subseteq \Delta^\omega$ is obviously ωPDA-recognizable. As we will see in the following theorem, the series $s = h((\text{Val}^\omega \circ r) \cap L) \in \dot{\mathbb{R}}^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle$ is ωWPDA-recognizable because $\dot{\mathbb{R}}$ is idempotent. The series s calculates the greatest accumulation point of the ratio of events a happening at the weekend (days 5 and 6) compared to all occurrences of events a. ▽

The following is the second main result of this chapter, a Nivat-like decomposition theorem.

Theorem 5.17. *Let Σ be an alphabet and $(D, +, Val^\omega, \mathbb{0})$ an ω-valuation monoid. Then,*

$$D^{rec}\langle\langle\Sigma^\omega\rangle\rangle = D^{\mathcal{N}}_{\text{DET}}\langle\langle\Sigma^\omega\rangle\rangle = D^{\mathcal{N}}_{\text{UNAMB}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle\,.$$

If D is idempotent, $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle = D^{rec}\langle\langle\Sigma^\omega\rangle\rangle$.

Proof. First, we show $D^{\text{rec}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathcal{N}}_{\text{DET}}\langle\langle\Sigma^\omega\rangle\rangle$: Let $s \in D^{\text{rec}}\langle\langle\Sigma^\omega\rangle\rangle$. Thus there exists an ωWPDA $M = (Q, \Gamma, T, I, F, wt)$ over Σ such that $\|M\| = s$. We will show that there exist Δ, h, r and L such that $s = h((\text{Val}^\omega \circ r) \cap L)$.

Let $\Delta = T$ and let $r = \text{wt}\colon \Delta \to D$. We define $h\colon \Delta \to \Sigma$ by $h((q, a, q', s)) = a$. Note that the automaton does not allow ϵ-transitions and therefore, h is well-defined. We construct an unweighted ωPDA $M' = (Q, \Gamma, T', I, F)$ over Δ with

$$T' = \{(q, (q, a, p, s), p, s) \mid (q, a, p, s) \in T\}\,.$$

Note that M' accepts exactly the successful runs of M. As there is at most one transition of M' with label (q, a, p, s), M' is unambiguous and deterministic. Define $L = \mathcal{L}(M')$.

Let $w \in \Sigma^\omega$. Therefore,

$$\begin{aligned} h((\mathrm{Val}^\omega \circ r) \cap L)(w) &= \sum \big(((\mathrm{Val}^\omega \circ r) \cap L)(w') \mid w' \in \Sigma^\omega \text{ and } h(w') = w\big) \\ &= \sum \big((\mathrm{Val}^\omega \circ r)(w') \mid w' \in L \text{ and } h(w') = w\big) \\ &= \sum \big(\mathrm{Val}^\omega(\mathrm{wt}(w')) \mid w' \in L \text{ and } h(w') = w\big) \\ &= \sum (\mathrm{Val}^\omega(\mathrm{wt}(w')) \mid w' \text{ successful run of } M \text{ on } w) \\ &= \|M\|(w)\,. \end{aligned}$$

It follows, $h((\mathrm{Val}^\omega \circ r) \cap L) = \|M\| = s$.

The inclusions $D^{\mathcal{N}}_{\mathrm{DET}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathcal{N}}_{\mathrm{UNAMB}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle$ is true by definition. The converse $D^{\mathcal{N}}_{\mathrm{UNAMB}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathrm{rec}}\langle\langle\Sigma^\omega\rangle\rangle$ is proven by the closure properties of Lemmas 5.11, 5.12 and 5.13(1).

If D is idempotent, by Lemmas 5.11, 5.12 and 5.13(2), we get $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathrm{rec}}\langle\langle\Sigma^\omega\rangle\rangle$. □

The inclusion $D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle \subseteq D^{\mathrm{rec}}\langle\langle\Sigma^\omega\rangle\rangle$ does not hold in general for non-idempotent D. As a counterexample, take the alphabet Σ, language L and ω-valuation monoid $\mathbb{N}^\infty$ from Example 5.14 and let $\Delta = \Sigma$, $r(\sigma) = 1$ for all $\sigma \in \Sigma$ and $h(\sigma) = \sigma$ for all $\sigma \in \Sigma$. Then $h((\mathrm{Val}^\omega \circ r) \cap L) \in D^{\mathcal{N}}\langle\langle\Sigma^\omega\rangle\rangle \setminus D^{\mathrm{rec}}\langle\langle\Sigma^\omega\rangle\rangle$ as stated in Example 5.14.

5.4 Logic for Weighted ω-Pushdown Automata

The third main goal of this chapter is a logical characterization of weighted simple ω-pushdown automata. This section introduces the corresponding logic. This logic is based on a logic for unweighted context-free languages of finite words by Lautemann, Schwentick and Thérien (1994). We adapted it to infinite words in Chapter 2 (Droste, Dziadek and Kuich, 2020a).

Our logic has three components. The first component is a monadic second-order logic (MSO) that adds set variables to first-order logic. By Büchi (1960, 1966), Elgot (1961) and Trakhtenbrot (1961), MSO has the same expressive power on finite and infinite words as finite automata.

The second component adds the weights to the logic. Here, this is done by a new layer of formulas that are to be interpreted quantitatively, using the operations of the ω-pv-monoid. Formulas of the unweighted part of the logic will be interpreted as $\mathbb{0}$ or $\mathbb{1}$ in the ω-pv-monoid.

The third component is a dyadic second-order predicate – a binary relation that is called matching relation. Every formula will be allowed to use exactly one such predicate to link positions in words. A matching relation has a specific shape that makes it possible to argue about the stack in pushdown automata or the brackets in

Dyck languages (cf. Chomsky and Schützenberger, 1963) or even about the nesting in nested words.

Our logic in Chapter 2 used only the first and the third component of the three components mentioned above.

Let $w \in \Sigma^\omega$ be an ω-word. The set of all positions of w is $\mathbb{N}$. As in Chapter 2, Definition 2.20, a binary relation $M \subseteq \mathbb{N} \times \mathbb{N}$ is a *matching* if M is compatible with $<$, each element belongs to at most one pair in M and M is noncrossing. We let $\mathrm{Match}(\mathbb{N})$ denote the set of all matchings in $\mathbb{N} \times \mathbb{N}$.

Let V_1, V_2 denote countable and pairwise disjoint sets of first-order and second-order variables, respectively. We fix a *matching variable* $\mu \notin V_1 \cup V_2$. Let $\mathcal{V} = V_1 \cup V_2 \cup \{\mu\}$. Furthermore, D is always an ω-pv-monoid $(D, +, \mathrm{Val}^\omega, \diamond, \mathbb{0}, \mathbb{1})$.

Definition 5.18. Let Σ be an alphabet. The set $\omega\mathrm{MSO}(\Sigma, D)$ of *weighted matching ω-MSO formulas* over Σ and D is defined by the extended Backus-Naur form

$$\beta ::= P_a(x) \mid x \leq y \mid x \in X \mid \mu(x,y) \mid \neg\beta \mid \beta \vee \beta \mid \exists x.\, \beta \mid \exists X.\, \beta$$
$$\varphi ::= d \mid \beta \mid \varphi \oplus \varphi \mid \varphi \otimes \varphi \mid \bigoplus_x \varphi \mid \bigoplus_X \varphi \mid \mathrm{Val}_x\, \varphi$$

where $a \in \Sigma$, $d \in D$, $x, y \in V_1$ and $X \in V_2$. We call all formulas β *boolean formulas*. ▼

Note that the set of boolean formulas is equal to $\omega\mathrm{MSO}(\Sigma)$ as defined in Definition 2.21 for the unweighted logic.

Variables denote positions in the word. $P_a(x)$ is a predicate indicating that the x^{th} letter of the word is a. Furthermore, $\mu(x,y)$ says that x and y will be matched. The operations $\oplus$ and $\otimes$ evaluate to the operations $+$ and $\diamond$ of the ω-pv-monoid D, respectively. The formulas $\bigoplus_x$ and $\bigoplus_X$ sum up over all possible instances of x and X, respectively. $\mathrm{Val}_x\, \varphi$ applies Val^ω to the sequence of infinitely many φ, each of them instantiated with a position $x \in \mathbb{N}$.

Similar to Chapter 2, page 26, we make the following definitions. A *$\mathcal{V}$-assignment* is a mapping $\sigma\colon \mathcal{V} \to \mathbb{N} \cup 2^{\mathbb{N}} \cup \mathrm{Match}(\mathbb{N})$ such that $\sigma(V_1) \subseteq \mathbb{N}$, $\sigma(V_2) \subseteq 2^{\mathbb{N}}$ and $\sigma(\mu) \in \mathrm{Match}(\mathbb{N})$. Let $\bar{\mathcal{V}}$ be the collection of all such mappings σ.

Let σ be a $\mathcal{V}$-assignment. For $x \in V_1$ and $j \in \mathbb{N}$, the update $\sigma[x/j]$ is the $\mathcal{V}$-assignment σ' with $\sigma'(x) = j$ and $\sigma'(y) = \sigma(y)$ for all $y \in \mathcal{V} \setminus \{x\}$. The update $\sigma[X/J]$ for $X \in V_2$ and $J \subseteq \mathbb{N}$ and the update $\sigma[\mu/M]$ for $M \in \mathrm{Match}(\mathbb{N})$ are defined similar.

Let $\varphi \in \omega\mathrm{MSO}(\Sigma, D)$ be a boolean formula, $w = a_0 a_1 a_2 \ldots \in \Sigma^\omega$ and let σ be a $\mathcal{V}$-assignment. Similarly to our unweighted logic, we inductively define $(w, \sigma) \models \varphi$ over the structure of φ as shown in Table 2.1 on page 27, where $a \in \Sigma$, $d \in D$, $x, y \in V_1$ and $X \in V_2$. The logical counterparts $\wedge$, $\rightarrow$, $\forall x.\, \varphi$ and $\forall X.\, \varphi$ can be gained from negation and the existing operators.

Now let $\varphi \in \omega\mathrm{MSO}(\Sigma, D)$ be a non-boolean formula. The semantics of φ is the mapping $[\![\varphi]\!]\colon \Sigma^\omega \times \bar{\mathcal{V}} \to D$ defined inductively on the structure of φ as shown in Table 5.1.

Note how formulas $\phi \otimes \psi$ are evaluated by the product operation $\diamond$ in the ω-pv-monoid and also note that our ωWPDAs do not have direct access to this operation.

$$
\begin{aligned}
[\![d]\!](w,\sigma) &= d \\
[\![\beta]\!](w,\sigma) &= \begin{cases} \mathbb{1}, & \text{if } (w,\sigma) \models \beta, \\ \mathbb{0}, & \text{otherwise} \end{cases} \\
[\![\varphi \oplus \psi]\!](w,\sigma) &= [\![\varphi]\!](w,\sigma) + [\![\psi]\!](w,\sigma) \\
[\![\varphi \otimes \psi]\!](w,\sigma) &= [\![\varphi]\!](w,\sigma) \diamond [\![\psi]\!](w,\sigma) \\
[\![\bigoplus_x \varphi]\!](w,\sigma) &= \textstyle\sum_{i\in\mathbb{N}}([\![\varphi]\!](w,\sigma[x/i])) \\
[\![\bigoplus_X \varphi]\!](w,\sigma) &= \textstyle\sum_{I\subseteq\mathbb{N}}([\![\varphi]\!](w,\sigma[X/I])) \\
[\![\mathrm{Val}_x\ \varphi]\!](w,\sigma) &= \mathrm{Val}^\omega(([\![\varphi]\!](w,\sigma[x/i]))_{i\in\mathbb{N}})
\end{aligned}
$$

Table 5.1: The semantics of weighted $\omega\mathrm{MSO}(\Sigma, D)$ formulas

However, the first two layers of our logic, the $\omega\mathrm{MSO}(D, \Sigma)$ formulas, will be translated into weighted nested ω-word automata and simple series of those automata are closed under intersection and therefore, $\diamond$ can be translated by a product construction.

We will use the boolean formula $\mathrm{MATCHING}(\mu) \in \omega\mathrm{MSO}(\Sigma, D)$ which ensures that $\mu \in \mathrm{Match}(\mathbb{N})$. For details, see Definition 2.22 in Chapter 2.

Definition 5.19. The set of formulas of *weighted matching ω-logic* over Σ and D, $\omega ML(\Sigma, D)$, denotes the set of all formulas ψ of the form

$$\psi = \bigoplus_\mu (\varphi \otimes \mathrm{MATCHING}(\mu))\,,$$

for short $\psi = \bigoplus_\mu^{\mathrm{match}} \varphi$, where $\varphi \in \omega\mathrm{MSO}(D, \Sigma)$. ▼

Let $\psi = \bigoplus_\mu^{\mathrm{match}} \varphi$, $w \in \Sigma^\omega$ and let σ be a $\mathcal{V}$-assignment. Then,

$$[\![\psi]\!](w,\sigma) = \sum_{M\in\mathrm{Match}(\mathbb{N})} ([\![\varphi]\!](w,\sigma[\mu/M]))\,.$$

Let $\psi \in \omega\mathrm{ML}(\Sigma, D)$. We denote by $\mathrm{Free}(\psi) \subseteq \mathcal{V}$ the set of *free variables* of ψ. A formula ψ with $\mathrm{Free}(\psi) = \emptyset$ is called a *sentence*. For a sentence ψ, $[\![\psi]\!](w,\sigma)$ does not depend on σ. It will therefore be omitted and we only write $[\![\psi]\!](w)$ where the series $[\![\psi]\!] \colon \Sigma^\omega \to D$ is called *defined* by ψ. A series $s \colon \Sigma^\omega \to D$ is *weighted-ωML-definable* if there exists a sentence $\psi \in \omega\mathrm{ML}(\Sigma, D)$ such that $[\![\psi]\!] = s$.

Example 5.20. We here define a logical sentence that defines the same series as in Example 5.6. Consider the same ω-pv-monoid $(\bar{\mathbb{R}}, \sup, \mathrm{specialavg}, +, -\infty, 0)$ as there.

Now, we define some subformulas first. The formula p_{start} ensures that the first symbol is a request and the formula $p_{\mathrm{structure}}$ ensures that requests occur directly after answers. The formula p_{matching} relates corresponding call and returns and forbids calls

without returns and vice versa. Furthermore, calls must be returned before giving the answer to the clients. Finally, the server has to serve clients infinitely often.

$$\begin{aligned}
\mathrm{next}(x,y) &= x < y \wedge \neg(\exists z.\, x \leq z \leq y)\\
\mathrm{first}(x) &= \forall y.\, x \leq y\\
p_{\mathrm{start}} &= \forall x.\, (\mathrm{first}(x) \to P_{\mathrm{req}}(x))\\
p_{\mathrm{structure}} &= \forall x \forall y.\, \mathrm{next}(x,y) \to (P_{\mathrm{ans}}(x) \leftrightarrow P_{\mathrm{req}}(y))\\
p_{\mathrm{matching}} &= \forall x.\, P_{\mathrm{call}}(x) \to \exists y.\, P_{\mathrm{ret}}(y) \wedge \mu(x,y)\\
&\quad \wedge \forall y.\, P_{\mathrm{ret}}(y) \to \exists x.\, P_{\mathrm{call}}(x) \wedge \mu(x,y)\\
&\quad \wedge \forall x.\, \forall y.\, \big[\mu(x,y) \to \neg\big(\exists z.\, x \leq z \leq y \wedge P_{\mathrm{ans}}(z)\big)\big]\\
&\quad \wedge \forall x.\, \forall y.\, \big[\mu(x,y) \to \big((P_{\mathrm{req}}(x) \wedge P_{\mathrm{ans}}(y)) \vee (P_{\mathrm{call}}(x) \wedge P_{\mathrm{ret}}(y))\big)\big]\\
p_{\mathrm{inf_serving}} &= \forall x.\, \exists y.\, (x < y \wedge P_{\mathrm{req}}(y))\\
\varphi_{\mathrm{unweighted}} &= p_{\mathrm{start}} \wedge p_{\mathrm{structure}} \wedge p_{\mathrm{matching}} \wedge p_{\mathrm{inf_serving}}
\end{aligned}$$

The weights of the words are distributed depending on the symbol and the formula $\mathrm{wt}(x)$ takes care of that for a position x and $\varphi_{\mathrm{weighted}}$ manages that for every position in the word and already applies the Val^{ω} function to the resulting sequence of weights.

$$\begin{aligned}
\mathrm{wt}(x) &= \big(P_{\mathrm{req}}(x) \vee P_{\mathrm{ans}}(x)\big) \oplus \big((P_{\mathrm{call}}(x) \vee P_{\mathrm{ret}}(x) \vee P_{\mathrm{wait}}(x)) \otimes 1\big)\\
\varphi_{\mathrm{weighted}} &= \mathrm{Val}_x\, \mathrm{wt}(x)
\end{aligned}$$

Then, we quantify over the matching variable and only consider the weight calculated in $\varphi_{\mathrm{weighted}}$ if the formula $\varphi_{\mathrm{unweighted}}$ is true:

$$\psi = \bigoplus\nolimits_{\mu}^{\mathrm{match}} \varphi_{\mathrm{unweighted}} \otimes \varphi_{\mathrm{weighted}}$$

Finally, we have $[\![\psi]\!] = \|\mathcal{A}\|$ for the ωWPDA $\mathcal{A}$ of Example 5.6. ▽

The weighted matching ω-logic, $\omega\mathrm{ML}(\Sigma, D)$, contains exactly one predicate μ and exactly one quantification over it. This corresponds to the behavior of pushdown automata where exactly one pushdown tape is used.

In contrast, the pushdown automaton uses the ω-valuation function Val^{ω} only once per run and never recursively. As formulas $\mathrm{Val}_x \mathrm{Val}_y\, \varphi \in \omega\mathrm{MSO}(\Sigma, D)$ are not always translatable into automata, we follow Droste and Gastin (2007), Droste and Meinecke (2012) and Droste and Dück (2017) and define some possible restrictions of our logic.

The set of *almost boolean formulas* is the smallest set of all formulas of $\omega\mathrm{MSO}(\Sigma, D)$ containing all constants $d \in D$ and all boolean formulas which is closed under $\oplus$ and $\otimes$.

Definition 5.21 (Droste and Dück, 2017; Droste and Meinecke, 2012)**.** Let $\varphi \in \omega\mathrm{MSO}(\Sigma, D)$ and let $\mathrm{const}(\psi)$ be the set of all elements of D occurring in ψ. We call φ

1. *strongly-$\otimes$-restricted* if for all subformulas $\mu \otimes \nu$ of φ:
   either μ and ν are almost boolean or μ is boolean or ν is boolean.

2. $\otimes$-*restricted* if for all subformulas $\mu \otimes \nu$ of φ:
   either μ is almost boolean or ν is boolean.
3. *commutatively-*$\otimes$*-restricted* if for all subformulas $\mu \otimes \nu$ of φ:
   either const(μ) and const(ν) commute or μ is almost boolean.
4. *Val-restricted* if for all subformulas $\mathrm{Val}_x\ \mu$ of φ, μ is almost boolean.
5. *syntactically restricted* if it is both Val-restricted and strongly-$\otimes$-restricted. ▼

Let now $\psi = \bigoplus_{\mu}^{\text{match}} \varphi \in \omega\mathrm{ML}(D, \Sigma)$. For $X \in \{\text{strongly-}\otimes, \text{Val}, \text{syntactically}\}$, we also say that ψ is X-restricted if φ is X-restricted.

The following will be the third main result of this chapter. *Regular* ω-pv-monoids will be defined in the next section on page 116 as they depend on nested ω-word automata.

Theorem 5.22. *Let D be a regular ω-pv-monoid and $s\colon \Sigma^\omega \to D$ be a series.*

1. *The following are equivalent:*
   a) *s is ωWPDA-recognizable*
   b) *There is a syntactically restricted $\omega ML(\Sigma, D)$-sentence φ with $[\![\varphi]\!] = s$.*
2. *Let D be left-distributive. The following are equivalent:*
   a) *s is ωWPDA-recognizable*
   b) *There is a Val-restricted and $\otimes$-restricted $\omega ML(\Sigma, D)$-sentence φ with $[\![\varphi]\!] = s$.*
3. *Let D be a cc-ω-valuation semiring. The following are equivalent:*
   a) *s is ωWPDA-recognizable*
   b) *There is a Val-restricted and commutatively-$\otimes$-restricted $\omega ML(\Sigma, D)$-sentence φ with $[\![\varphi]\!] = s$.*

This theorem will be proved in Section 5.6.

5.5 Weighted Nested ω-Word Languages

The $\omega\mathrm{MSO}(\Sigma, D)$ formulas correspond exactly to weighted nested ω-word languages as by Droste and Dück (2017) (cf. Alur and Madhusudan, 2004, for the original unweighted nested words). In fact, without considering the existential quantification over the matching relation $\exists^{\text{match}}\mu$, the matching must explicitly be encoded in the words; the result is a nested word. For the convenience of the reader, weighted nested ω-words/visibly pushdown languages (Droste and Dück, 2017) are recalled in this section.

A *nested ω-word* nw over Σ is a pair $(w, \nu) = (a_0 a_1 a_2 \ldots, \nu)$ where $w \in \Sigma^\omega$ is an ω-word and $\nu \in \mathrm{Match}(\mathbb{N})$ is a matching relation over $\mathbb{N}$. Let $NW^\omega(\Sigma)$ denote the set of all nested ω-words over Σ. For two positions $i, j \in \mathbb{N}$ with $\nu(i, j)$, we call i a *call position* and j a *return position*. If i is neither call nor return, we call it an *internal position*.

Definition 5.23. A *weighted stair Muller nested word automaton* (ωWNWA) over the alphabet Σ is a tuple $M = (Q, T, I, \mathcal{F})$ where

- Q is a finite set of states,
- $T = T_{\text{call}} \cup T_{\text{int}} \cup T_{ret}$ is a set of transitions, with
  - $T_{\text{call}}, T_{\text{int}} \colon Q \times \Sigma \times Q \to D$,
  - $T_{\text{ret}} \colon Q \times Q \times \Sigma \times Q \to D$,
- $I \subseteq Q$ is a set of initial states,
- $\mathcal{F} \subseteq 2^Q$ is a set of (Muller-accepting) sets of final states. ▼

A *run* of the ωWNWA M on the nested ω-word $nw = (a_0 a_1 a_2 \ldots, \nu)$ is an infinite sequence of states $\rho = (q_0, q_1, \ldots)$. We denote by $\text{wt}_M(\rho, nw, i)$ the weight of the transition of ρ used at position $i \in \mathbb{N}$, defined as

$$\text{wt}_M(\rho, nw, i) = \begin{cases} T_{\text{call}}(q_i, a_i, q_{i+1}), & \text{if } \nu(i,j) \text{ for some } j > i, \\ T_{\text{int}}(q_i, a_i, q_{i+1}), & \text{if } i \text{ is internal}, \\ T_{\text{ret}}(q_i, q_j, a_i, q_{i+1}), & \text{if } \nu(j,i) \text{ for some } j < i. \end{cases}$$

Hereby, the matching relation ν decides which transition will be used. On return positions, the transition has additionally access to the state of the automaton before reading the matching call symbol.

The weight $\text{wt}_M(\rho, nw)$ of the run ρ on nw is defined by

$$\text{wt}_M(\rho, nw) = \text{Val}^\omega((\text{wt}_M(\rho, nw, i))_{i \in \mathbb{N}}) .$$

We call q_i for $i \in \mathbb{N}$ *top-level* if there exist no positions $j, k \in \mathbb{N}$ with $j < i < k$ and $\nu(j,k)$. Let

$$Q^{\text{top}}(\rho) = \rho \restriction_{\text{top-level}}$$

be the subsequence of all top-level positions in the run $\rho = (q_i)_{i \in \mathbb{N}}$. Note that $Q^{\text{top}}(\rho)$ is still an infinite sequence.

A run $\rho = (q_i)_{i \in \mathbb{N}}$ is *successful* if $q_0 \in I$ and $\text{Inf}(Q^{\text{top}}(\rho)) \in \mathcal{F}$. The *behavior* of M is the function $\|M\| \colon NW^\omega(\Sigma) \to D$ defined by

$$\|M\|(nw) = \sum \left(\text{wt}_M(\rho, nw) \mid \rho \text{ successful}\right) .$$

Every function $s \colon NW^\omega(\Sigma) \to D$ is called a *nested ω-word series* (nw-series). Every nw-series s which is the behavior of some ωWNWA over D is called *ωWNWA-recognizable*.

Note that the definition here differs slightly from the definitions given by Droste and Dück (2017) as we do not allow pending calls and pending returns. In Droste and Dück (2017), it is possible that ν matches $i \in \mathbb{N}$ as a pending return like $\nu(-\infty, i)$ or as a pending call like $\nu(i, \infty)$. We do not consider these possibilities here. Instead, we will later assume a push in the ωWPDA without a succeeding pop not to be matched.

We will now discuss how ωMSO is an equivalent logic to ωWNWAs. Note that $\omega\text{MSO}(\Sigma, D)$ formulas may contain the free variable μ. Given a nested word $nw = (w, \nu)$, we let $\sigma(\mu) = \nu$ and make no difference between $(w, \sigma) \in \Sigma^\omega \times (\{\mu\} \to$

Match($\mathbb{N}$)) and the nested word nw. We extend the semantics definitions as follows. Let $\varphi \in \omega\mathrm{MSO}(\Sigma, D)$ and $\mathrm{Free}(\varphi) \subseteq \{\mu\}$, then we define $[\![\varphi]\!]_{\mathrm{nw}} \colon NW^\omega(\Sigma) \to D$ by letting

$$[\![\varphi]\!]_{\mathrm{nw}}(w, \nu) = [\![\varphi]\!](w, \sigma) \qquad \text{for } \sigma(\mu) = \nu\,.$$

Let $d \in D$ denote the *constant series* with value d, i.e., $d(nw) = d$ for each $nw \in NW^\omega(\Sigma)$.

An ω-pv-monoid D is called *regular* if all constant series of D are ωWNWA-recognizable. In other words, D is regular if for any alphabet Σ, we have: For each $d \in D$, there exists an ωWNWA A_d with $\|A_d\| = d$.

Note that here, regularity of ω-pv-monoids is defined by the means of ωWNWAs. In the proof of Theorem 5.24, this is used in the structural induction as a logical formula $\varphi = d$, for a weight d, can otherwise not necessarily be translated into an automaton.

Sufficient properties for an ω-pv-monoid to be regular are shown by Droste and Meinecke (2012). Especially left-multiplicative and left-Val$^\omega$-distributive ω-pv-monoids are regular because we can easily construct ωWNWAs (and even ωWFAs) for every constant series. All ω-pv-monoids in Example 5.2 are regular.

Theorem 5.24 (Droste and Dück, 2017)**.** *Let D be a regular ω-pv-monoid and consider the nw-series $s \colon NW^\omega(\Sigma) \to D$.*

1. *The following are equivalent:*
   a) *s is ωWNWA-recognizable*
   b) *There is a syntactically restricted ωMSO(Σ, D)-formula φ with $\mathrm{Free}(\varphi) \subseteq \{\mu\}$ and $[\![\varphi]\!]_{nw} = s$.*
2. *Let D be left-distributive. The following are equivalent:*
   a) *s is ωWNWA-recognizable*
   b) *There is a Val-restricted and $\otimes$-restricted ωMSO(Σ, D)-formula φ with $\mathrm{Free}(\varphi) \subseteq \{\mu\}$ and $[\![\varphi]\!]_{nw} = s$.*
3. *Let D be a cc-ω-valuation semiring. The following are equivalent:*
   a) *s is ωWNWA-recognizable*
   b) *There is a Val-restricted and commutatively-$\otimes$-restricted ωMSO(Σ, D)-formula φ with $\mathrm{Free}(\varphi) \subseteq \{\mu\}$ and $[\![\varphi]\!]_{nw} = s$.*

The mapping $\pi \colon NW^\omega(\Sigma) \to \Sigma^\omega$ removes the nesting relation from the nested word, i.e., for $nw = (w, \nu)$, we define $\pi(nw) = w$. This can be extended to nw-series $s \colon NW^\omega(\Sigma) \to D$ by setting $\pi(s)(w) = \sum_{nw \in \pi^{-1}(w)} s(nw)$ which equals $\pi(s)(w) = \sum_{M \in \mathrm{Match}(\mathbb{N})} s(w, \varnothing[\mu/M])$.

The following is crucial for the rest of the chapter.

Lemma 5.25. *Let $s \colon NW^\omega(\Sigma) \to D$ be an ωWNWA-recognizable nw-series. Then the series $\pi(s) : \Sigma^\omega \to D$ is ωWPDA-recognizable.*

For unweighted languages, there is a similar proof by Blass and Gurevich (2006) and Droste, Dziadek and Kuich (2020a). Here, the proof is more complicated because

the acceptance conditions differ. We have to construct a Büchi-accepting pushdown automaton from a stair Muller nested-word automaton.

Proof. Let $M = (Q, T, I, \mathcal{F})$ be an ωWNWA with $\|M\| = s$. By Theorem 5.9, it suffices to construct a Muller-accepting ωWPDA $M' = (Q', \Gamma', T', I', \mathcal{F}', \mathrm{wt}')$ with $\|M'\| = \pi(s)$.

Let $T = T_{\mathrm{call}} \cup T_{\mathrm{int}} \cup T_{\mathrm{ret}}$. Then $Q' = Q \times \{0,1\}$, $\Gamma' = Q'$ and $I' = I \times \{1\}$ and

$$\begin{aligned} T' = &\Big\{ \big((p,i), a, (q,0), (\downarrow, (p,i))\big) \mid T_{\mathrm{call}}(p,a,q) \neq \mathbb{0}, i \in \{0,1\} \Big\} \cup \\ &\Big\{ \big((p,i), a, (q,i), \#\big) \mid T_{\mathrm{int}}(p,a,q) \neq \mathbb{0}, i \in \{0,1\} \Big\} \cup \\ &\Big\{ \big((p,0), a, (q,i), (\uparrow, (p',i))\big) \mid T_{\mathrm{ret}}(p,p',a,q) \neq \mathbb{0}, i \in \{0,1\} \Big\}. \end{aligned}$$

The new automaton M' pushes states onto the stack. This allows M' to simulate M as the nested-word automaton has access to the states of matched calls.

Additionally, a state $(q,1)$ is top-level and any state $(q,0)$ is not. When pushing something onto the stack, the next state is no longer top-level. Internal transitions do not change if the automaton is top-level or not. Finally, popping a state means that the new state is top-level exactly if the popped state was top-level before. Note that popping from top-level state $(p,1)$ is never possible.

The weights are directly transferred by setting

$$\begin{aligned} &\mathrm{wt}'\big((p,i), a, (q,0), (\downarrow, (p,i))\big) = T_{\mathrm{call}}(p,a,q), \\ &\mathrm{wt}'\big((p,i), a, (q,i), \#\big) = T_{\mathrm{int}}(p,a,q), \\ &\mathrm{wt}'\big((p,0), a, (q,i), (\uparrow, (p',i))\big) = T_{\mathrm{ret}}(p,p',a,q), \qquad \text{for } i \in \{0,1\}. \end{aligned}$$

Finally, we set

$$\mathcal{F}' = \big\{ (F \times \{1\}) \cup (S \times \{0\}) \mid F \in \mathcal{F}, S \subseteq Q \big\}.$$

This means, that the original states in the sets $F \in \mathcal{F}$ are accepting if they are top-level and in between top-level positions, the automaton can visit arbitrary subsets of states that are not top-level. □

5.6 Equivalence of Logic and Automata

This section proves the equivalence of $\omega\mathrm{ML}(\Sigma, D)$ and weighted simple ω-pushdown automata.

Let $a = (a_1, \ldots, a_n) \in A_1 \times \ldots \times A_n$ be a tuple. Then we define as in Section 2.6 for $1 \leq i \leq n$ the i^{th} *projection* of a by $\mathrm{pr}_i(a) = a_i$.

Lemma 5.26. *Let D be a regular ω-pv-monoid and $s \colon \Sigma^\omega \to D$ be an ωWPDA-recognizable series. Then s is ωML-definable by a syntactically restricted $\omega\mathrm{ML}(D, \Sigma)$-sentence.*

Proof. By assumption, there exists an ωWPDA $M = (Q, \Gamma, T, I, F, \mathrm{wt})$ over Σ and D with $\|M\| = s$. Let $T = \{t_1, \ldots, t_m\}$ be an enumeration of the transitions. We define a syntactically restricted $\omega\mathrm{ML}(\Sigma, D)$-sentence θ such that $[\![\theta]\!] = s$ as follows. We proceed similarly to the lines of Droste and Gastin (2007), Vogler, Droste and Herrmann (2016) and Droste and Dück (2017).

We defined our logic in layers and the first layer, the boolean $\omega\mathrm{MSO}(\Sigma, D)$ formulas are equivalent to unweighted $\omega\mathrm{MSO}(\Sigma)$ formulas (see note and definition on page 111). We also defined ωWPDA and ωPDA similar in a way such that we can reuse the $\omega\mathrm{MSO}(\Sigma)$ formulas that we used in the proof of Lemma 2.31 to prove our BET-Theorem of unweighted ω-context-free languages.

Thus, let $\psi_{\text{successful}}$ be the $\omega\mathrm{MSO}(\Sigma, D)$ formula defined in Equation (2.8) on page 32 that is true only for successful runs of M. This formula internally uses the set of second-order variables $\mathcal{V} = \{X_t \mid t \in T\}$ to partition the positions in the input word corresponding to the transition used at that position.

The second step is to add weights. First, we will define an auxiliary formula

$$((x \in X) \to d) = (x \notin X) \oplus ((x \in X) \otimes d)\,,$$

which is $\mathbb{1}$ if $x \notin X$ and d if $x \in X$.

We use the weight of one run

$$\theta_{\text{weight}} = \mathrm{Val}_x\big(\bigotimes_{t \in T} (x \in X_t \to \mathrm{wt}(t))\big)$$

to define

$$\theta_{\text{wt\_run}} = \psi_{\text{successful}} \otimes \theta_{\text{weight}}\,,$$

which is either $\mathbb{0}$ for non-successful runs or $\mathrm{Val}^\omega(\mathrm{wt}(\rho))$ for successful runs ρ.

Finally, let $\theta \in \omega\mathrm{ML}(\Sigma, D)$ be the sentence defined as

$$\theta = \bigoplus_{\mu}^{\text{match}} \bigoplus_{X_{t_1}} \bigoplus_{X_{t_2}} \cdots \bigoplus_{X_{t_m}} \theta_{\text{wt\_run}}\,.$$

The formula θ sums up all possible runs. It tries all possible ways to use the stack (by matching different positions) and tries all possible transitions.

Then,

$$\begin{aligned}
&[\![\theta]\!](w)\\
&= \sum_{M \in \mathrm{Match}(\mathbb{N})} \sum_{I_1 \subseteq \mathbb{N}} \sum_{I_2 \subseteq \mathbb{N}} \cdots \sum_{I_m \subseteq \mathbb{N}} [\![\theta_{\text{wt_run}}]\!](w, \varnothing[\mu/M, X_{t_1}/I_1, X_{t_2}/I_2, \ldots, X_{t_m}/I_m])\\
&= \sum \big([\![\theta_{\text{wt_run}}]\!](w, \sigma_\rho) \mid \sigma_\rho \text{ assignment for run } \rho \text{ of } M \text{ on } w\big)\\
&= \sum \big([\![\theta_{\text{weight}}]\!](w, \sigma_\rho) \mid \sigma_\rho \text{ assignment for successful run } \rho \text{ of } M \text{ on } w\big)\\
&= \sum (\mathrm{Val}^\omega(\mathrm{wt}(\rho)) \mid \rho \text{ successful run of } M \text{ on } w)\\
&= \|M\|(w)\,.
\end{aligned}$$

The formula θ_{weight} is the only subformula containing Val_x and its argument is almost boolean. The formula $\theta_{\text{wt\_run}}$ is strongly-$\otimes$-restricted as $\psi_{\text{successful}}$ is boolean. It follows that θ is syntactically restricted. □

Lemma 5.27. *Let D be a regular ω-pv-monoid.*

1. *Let ψ be a syntactically restricted $\omega ML(\Sigma, D)$-sentence. Then $[\![\psi]\!]\colon \Sigma^\omega \to D$ is ωWPDA-recognizable.*
2. *Let D be left-distributive and ψ be a Val-restricted and $\otimes$-restricted $\omega ML(D, \Sigma)$-sentence. Then $[\![\psi]\!]$ is ωWPDA-recognizable.*
3. *Let D be a cc-ω-valuation semiring and ψ be a Val-restricted and commutatively-$\otimes$-restricted $\omega ML(\Sigma, D)$-sentence. Then $[\![\psi]\!]$ is ωWPDA-recognizable.*

Proof. In all three cases, let $\psi = \bigoplus_{\mu}^{\mathrm{Match}} \varphi$ for $\varphi \in \omega\mathrm{MSO}(\Sigma, D)$. Apply Theorem 5.24 to infer that $[\![\varphi]\!]_{\mathrm{nw}}$ is ωWNWA-recognizable. Now, we use the projection $\pi\colon NW^\omega(\Sigma) \to \Sigma^\omega$ of Section 5.5 to get

$$\pi([\![\varphi]\!]_{\mathrm{nw}})(w) = \sum_{M \in \mathrm{Match}(\mathbb{N})} ([\![\varphi]\!](w, \varnothing[\mu / M])) = [\![\psi]\!](w)\,.$$

By Lemma 5.25, $[\![\psi]\!] = \pi([\![\varphi]\!]_{\mathrm{nw}})$ is ωWPDA-recognizable. □

Proof of Theorem 5.22. This is immediate by Lemmas 5.26 and 5.27. □

CHAPTER 6

Conclusion and Outlook

We first provided a BET-Theorem (Büchi, 1960; Elgot, 1961; Trakhtenbrot, 1961) for unweighted ω-context-free languages. The ω-Kleene closure enabled us to construct ω-context-free grammars in Greibach normal form. Then we used those grammars in Greibach normal form to construct a new normal form for pushdown automata, the simple ω-pushdown automata. Simple ω-pushdown automata do not use ϵ-transitions; in the literature, this is also called a *realtime* pushdown automaton. Realtime pushdown automata read a symbol of the input word in every transition — exactly like context-free grammars in Greibach normal form generate a letter in every derivation step. Additionally, each derivation step of context-free grammars in (quadratic) Greibach normal form increases the number of non-terminals in the sentential form by at most one. We showed that for realtime pushdown automata it suffices to handle at most one stack symbol per transition. Here the Greibach normal form provides exactly the properties needed to construct simple ω-pushdown automata.

Then, we introduced the ω-matching logic. A subset of our logic is similar to a logic by Alur and Madhusudan (2009) and therefore, we can apply their BET-Theorem for regular nested ω-word languages and prove the expressive equivalence of our logic and a projection of nested ω-word automata, i.e., together with our results on simple pushdown automata, we have that our logic is expressively equivalent to ω-context-free languages.

Note that we chose to use simple pushdown automata because they ease the proof of our BET-Theorem significantly, maybe they even allow the proof in the first place. The reason for this lies in something we could call the "essence of context-free languages". Our logic uses matching relations that resemble the brackets in Dyck languages. Dyck languages were introduced as the basis of the famous Chomsky-Schützenberger Theorem (Chomsky and Schützenberger, 1963). The correlation described above shows us that the same essentials can already be found in simple pushdown automata and gained from the Greibach normal form of context-free grammars.

In the next part of this thesis, we took similar steps to the above to heavily generalize our BET-Theorem but also our simple pushdown automata to weighted languages.

Thus, we extended the characterization of ω-algebraic series so that we can use the ω-Kleene closure to transfer the property of Greibach normal form from algebraic systems to mixed ω-algebraic systems and to ω-algebraic systems. This generalizes a fundamental property of context-free languages.

We believe that the same technique can be used to transfer other properties of algebraic systems to ω-algebraic systems. For instance, Cohen and Gold (1977) use

this technique also for the elimination of chain rules, for the Chomsky normal form and for effective decision methods of emptiness, finiteness and infiniteness.

Furthermore, we applied the new Greibach normal form for the construction of weighted simple reset pushdown automata. We extended our knowledge about pushdown matrices by several properties. Then we proved that our construction of simple reset pushdown automata is correct and every algebraic series is the behavior of a simple reset pushdown automaton.

We investigated infinite applications of simple reset pushdown matrices. The construction that we used for finite words had to be extended to exactly model the canonical solutions of ω-algebraic systems. With this new construction, we were able to prove that every ω-algebraic series is the behavior of a simple ω-reset pushdown automaton.

The model of simple pushdown automata seems to be a very natural model in itself. Simple pushdown automata do not use ϵ-transitions and only change one symbol of the stack per transition. For our BET-Theorem, these were exactly the properties needed for our proof. We think that simple pushdown automata, thought of as a normal form for pushdown automata, might help also in other proofs. Additionally, simple pushdown automata seem to have desirable practical benefits as they provide a very predictable model. Namely, they need exactly one transition per input symbol, the stack size is bounded by the length of the input read so far and inversely, to empty a stack of size n, at least n more input symbols must be read.

Simple pushdown automata occur when applying general homomorphisms to nested-word automata of finite words (Blass and Gurevich, 2006), to nested-word automata of infinite words (Chapter 2) and to weighted nested ω-word automata (Chapter 5). They have been used for a Büchi-type logical characterization of timed pushdown languages of finite words (Droste and Perevoshchikov, 2015a) and also in the weighted setting, simple pushdown automata of finite words have been used by Droste and Perevoshchikov (2015b).

As indicated in the introduction, we only describe one inclusion for the relation between ω-algebraic series and simple ω-reset pushdown automata over commutative complete semirings: all ω-algebraic series can be represented as the behavior of a simple ω-reset pushdown automaton. It is currently open if the behavior of every simple ω-reset pushdown automaton is an ω-algebraic series (cf. Figure 1.4 in Chapter 1). Although this seems obvious, for this direction, we still need a stronger version of the results by Droste, Ésik and Kuich (2017) and a generalization to infinite words of a result by Kuich and Salomaa (1986) that relates reset pushdown automata to pushdown automata. Droste, Ésik and Kuich (2017) show that for a weighted ω-pushdown automaton $\mathcal{P}$, there exists a mixed ω-algebraic system such that the behavior $\|\mathcal{P}\|$ of $\mathcal{P}$ is a component of a solution of this system. However, for our purposes here, we would need that $\|\mathcal{P}\|$ is a component of a *canonical* solution of this system. This is however still open.

The class of weighted ω-context-free languages, the ω-algebraic series, are a com-

paratively young class of weighted languages. Several traditional formal language problems have already been solved for this class, but others are still untouched. It seems worthwhile to investigate this class even further.

Finally, we defined ω-valuation monoids and ω-pushdown automata with weights from ω-valuation monoids. We generalized a fundamental result of unweighted automata theory: Büchi acceptance and Muller acceptance are expressively equivalent; we can show that this remains the case for weighted simple pushdown automata of infinite words.

For the class of languages recognized by our automata, we proved several closure properties and a Nivat-like decomposition theorem. It states that the weighted languages in our class are induced by an unweighted context-free language and a very simple weighted part; the two components can be intersected and a projection of this intersection gives us the original language.

The last main result is a logic that is expressively equivalent to weighted simple ω-pushdown automata. This logic has three layers. The first layer basically describes nested ω-word-languages. The first two layers together describe weighted nested ω-word-languages. The third layer existentially quantifies the matching variable and corresponds to a projection from nested words to context-free languages. In this way, we can apply the BET-Theorem for weighted regular nested ω-word-languages to obtain our equivalence result.

The present result raises the question of how weighted simple ω-pushdown automata over ω-valuation monoids and therefore also our weighted matching ω-logic relate to a corresponding notion of weighted context-free ω-languages, the ω-algebraic series; we described this for weighted simple ω-pushdown automata over commutative complete star-omega semirings in Chapter 4. It remains open however for ω-valuation monoids.

The weighted logic has to be restricted to gain the desired equivalence result. We borrow three different restrictions from the corresponding result on weighted nested ω-word automata and accordingly obtain inclusions between the restricted subclasses. It is open however whether these inclusions remain strict after applying the projection π from nested words to words.

For our weighted BET-Theorem, we also borrow the restriction of ω-valuation monoids to be regular. Regularity is defined by the means of ωWNWA. In our weighted BET-Theorem, Theorem 5.22, it would be desirable to generalize the notion of regular ω-pv-monoids to only require ωWPDA instead of ωWNWA. The classical inductive proof method in the used BET-Theorem of weighted nested ω-word automata (Theorem 5.24) no longer works in this case. However, it seems that ω-pv-monoids where constant series are ωWPDA-recognizable but not ωWNWA-recognizable are very artificial.

Bibliography

A. V. Aho, R. Sethi, J. D. Ullman. *Compilers: Principles, Techniques, and Tools*. Addison-Wesley, 1986. ISBN 0-201-10088-6.

R. Alur, K. Etessami, P. Madhusudan. A temporal logic of nested calls and returns. In: *Tools and Algorithms for the Construction and Analysis of Systems* (*TACAS 2004*), LNCS, vol. 2988, pp. 467–481. Springer, 2004. doi:10.1007/978-3-540-24730-2_35.

R. Alur, P. Madhusudan. Visibly pushdown languages. In: *ACM Symposium on Theory of Computing* (*STOC 2004*), pp. 202–211, 2004. doi:10.1145/1007352.1007390.

R. Alur, P. Madhusudan. Adding nesting structure to words. *Journal of the ACM*, vol. 56(3), pp. 16:1–16:43, 2009. doi:10.1145/1516512.1516518.

J.-M. Autebert, J. Berstel, L. Boasson. Context-free languages and pushdown automata. In: *Handbook of Formal Languages*, vol. 1: Word, Language, Grammar, chap. 3, pp. 111–174. Springer, 1997. doi:10.1007/978-3-642-59136-5_3.

P. Babari, M. Droste. A Nivat theorem for weighted picture automata and weighted MSO logics. *J. Comput. Syst. Sci.*, vol. 104, pp. 41–57, 2019. doi:10.1016/j.jcss.2017.02.009.

C. Baier, J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008. ISBN 9780262026499.

J. Berstel. *Transductions and Context-Free Languages*. B. G. Teubner, 1979. doi:10.1007/978-3-663-09367-1.

A. Blass, Y. Gurevich. A note on nested words. *Microsoft Research*, 2006. URL `https://www.microsoft.com/en-us/research/publication/180-a-note-on-nested-words/`

S. L. Bloom, Z. Ésik. *Iteration Theories*. EATCS Monographs on Theoretical Computer Science. Springer, 1993. doi:10.1007/978-3-642-78034-9.

P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, J. Srba. Infinite runs in weighted timed automata with energy constraints. In: *Formal Modeling and Analysis of Timed Systems* (*FORMATS 2008*), pp. 33–47. Springer, 2008. doi:10.1007/978-3-540-85778-5_4.

J. R. Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, vol. 6, pp. 66–92, 1960. doi:10.1002/malq.19600060105.

J. R. Büchi. Symposium on decision problems: On a decision method in restricted second order arithmetic. In: *Logic, Methodology and Philosophy of Science*, Studies in Logic and the Foundations of Mathematics, vol. 44, pp. 1–11. Elsevier, 1966. doi:10.1016/S0049-237X(09)70564-6.

K. Chatterjee, L. Doyen, T. A. Henzinger. Quantitative languages. In: *Computer Science Logic (CSL 2008)*, pp. 385–400. Springer, 2008. doi:10.1007/978-3-540-87531-4_28.

N. Chomsky. On certain formal properties of grammars. *Information and Control*, vol. 2(2), pp. 137–167, 1959. doi:10.1016/S0019-9958(59)90362-6.

N. Chomsky, M. P. Schützenberger. The algebraic theory of context-free languages. In: *Studies in Logic and the Foundations of Mathematics*, vol. 35: Computer Programming and Formal Systems, pp. 118–161. Elsevier, 1963. doi:10.1016/S0049-237X(08)72023-8.

E. M. Clarke, T. A. Henzinger, H. Veith, R. P. Bloem. *Handbook of Model Checking*. Springer, 2016. doi:10.1007/978-3-319-10575-8.

R. S. Cohen, A. Y. Gold. Theory of ω-languages I: Characterizations of ω-context-free languages. *Journal of Computer and System Sciences*, vol. 15(2), pp. 169–184, 1977. doi:10.1016/S0022-0000(77)80004-4.

J. H. Conway. *Regular Algebra and Finite Machines*. Chapman and Hall, 1971. ISBN 9780486485836.

M. Droste, S. Dück. Weighted automata and logics for infinite nested words. *Information and Computation*, vol. 253, pp. 448–466, 2017. doi:10.1016/j.ic.2016.06.010.

M. Droste, S. Dziadek, W. Kuich. Weighted simple reset pushdown automata. *Theoretical Computer Science*, vol. 777, pp. 252–259, 2019a. doi:10.1016/j.tcs.2019.01.016.

M. Droste, S. Dziadek, W. Kuich. Greibach normal form for ω-algebraic systems and weighted simple ω-pushdown automata. In: *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, LIPIcs, vol. 150, pp. 38:1–38:14, 2019b. doi:10.4230/LIPIcs.FSTTCS.2019.38.

M. Droste, S. Dziadek, W. Kuich. Logic for ω-pushdown automata. *Information and Computation*, 2020a. Special issue on "Weighted Automata", Accepted for publication.

M. Droste, S. Dziadek, W. Kuich. Nivat-theorem and logic for weighted pushdown automata on infinite words. In: *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, LIPIcs, vol. 182, pp. 44:1–44:14, 2020b. doi:10.4230/LIPIcs.FSTTCS.2020.44.

M. Droste, S. Dziadek, W. Kuich. Greibach normal form for ω-algebraic systems and weighted simple ω-pushdown automata, 2020c. Submitted, arXiv:2007.08866.

M. Droste, Z. Ésik, W. Kuich. The triple-pair construction for weighted ω-pushdown automata. In: *Conference on Automata and Formal Languages* (*AFL 2017*), Electronic Proceedings in Theoretical Computer Science, vol. 252, pp. 101–113, 2017. doi:10.4204/EPTCS.252.12.

M. Droste, P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, vol. 380(1-2), pp. 69–86, 2007. doi:10.1016/j.tcs.2007.02.055.

M. Droste, P. Gastin. Weighted automata and weighted logics. In: *Handbook of Weighted Automata* (eds. M. Droste, W. Kuich, H. Vogler), EATCS Monographs in Theoretical Computer Science, chap. 5, pp. 175–211. Springer, 2009. doi:10.1007/978-3-642-01492-5_5.

M. Droste, W. Kuich. A Kleene theorem for weighted ω-pushdown automata. *Acta Cybernetica*, vol. 23, pp. 43–59, 2017. doi:10.14232/actacyb.23.1.2017.4.

M. Droste, W. Kuich, H. Vogler (eds.). *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009. doi:10.1007/978-3-642-01492-5.

M. Droste, D. Kuske. Weighted automata. In: *Handbook of Automata Theory* (ed. J.-E. Pin), chap. 4. European Mathematical Society, to appear.

M. Droste, I. Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Information and Computation*, vol. 220, pp. 44–59, 2012. doi:10.1016/j.ic.2012.10.001.

M. Droste, V. Perevoshchikov. A logical characterization of timed pushdown languages. In: *Computer Science Symposium in Russia* (*CSR 2015*), LNCS, vol. 9139, pp. 189–203. Springer, 2015a. doi:10.1007/978-3-319-20297-6_13.

M. Droste, V. Perevoshchikov. Logics for weighted timed pushdown automata. In: *Fields of Logic and Computation II*, pp. 153–173. Springer, 2015b. doi:10.1007/978-3-319-23534-9_9.

M. Droste, G. Rahonis. Weighted automata and weighted logics on infinite words. In: *Developments in Language Theory* (*DLT 2006*), vol. 54, pp. 49–58. Springer, 2006. doi:10.1007/11779148_6.

M. Droste, H. Vogler. Kleene and Büchi theorems for weighted automata and multi-valued logics over arbitrary bounded lattices. In: *Developments in Language Theory* (*DLT 2010*), LNCS, vol. 6224, pp. 160–172. Springer, 2010. doi:10.1007/978-3-642-14455-4_16.

M. Droste, H. Vogler. The Chomsky-Schützenberger theorem for quantitative context-free languages. *International Journal of Foundations of Computer Science*, vol. 25(08), pp. 955–969, 2014. doi:10.1142/S0129054114400176.

S. Eilenberg. *Automata, Languages, and Machines*, Pure and Applied Mathematics, vol. 59, Part A. Elsevier, 1974. doi:10.1016/S0079-8169(08)60880-6.

C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, vol. 98, pp. 21–51, 1961. doi:10.2307/2270940.

C. C. Elgot. Matricial theories. *Journal of Algebra*, vol. 42(2), pp. 391–421, 1976. doi:10.1016/0021-8693(76)90106-X.

E. A. Emerson. Temporal and modal logic. In: *Handbook of Theoretical Computer Science*, vol. B: Formal Models and Semantics, chap. 16, pp. 995–1072. Elsevier, 1990. doi:10.1016/b978-0-444-88074-1.50021-4.

Z. Ésik, S. Iván. Büchi context-free languages. *Theoretical Computer Science*, vol. 412(8), pp. 805–821, 2011. doi:10.1016/j.tcs.2010.11.026.

Z. Ésik, W. Kuich. A semiring-semimodule generalization of ω-context-free languages. In: *Theory Is Forever*, LNCS, vol. 3113, pp. 68–80. Springer, 2004. doi:10.1007/978-3-540-27812-2_7.

Z. Ésik, W. Kuich. A semiring-semimodule generalization of ω-regular languages I. *Journal of Automata, Languages and Combinatorics*, vol. 10(2–3), pp. 203–242, 2005a. doi:10.25596/jalc-2005-203.

Z. Ésik, W. Kuich. A semiring-semimodule generalization of ω-regular languages II. *Journal of Automata, Languages and Combinatorics*, vol. 10(2–3), pp. 243–264, 2005b. doi:10.25596/jalc-2005-243.

Z. Ésik, W. Kuich. *Modern Automata Theory*, 2007a.
URL `http://www.dmg.tuwien.ac.at/kuich`

Z. Ésik, W. Kuich. On iteration semiring-semimodule pairs. *Semigroup Forum*, vol. 75(1), pp. 129–159, 2007b. doi:10.1007/s00233-007-0709-7.

J. Esparza, D. Hansel, P. Rossmanith, S. Schwoon. Efficient algorithms for model checking pushdown systems. In: *Computer Aided Verification (CAV 2000)*, LNCS, vol. 1855, pp. 232–247. Springer, 2000. doi:10.1007/10722167_20.

D. Giammarresi, A. Restivo, S. Seibert, W. Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Information and Computation*, vol. 125(1), pp. 32–45, 1996. doi:10.1006/inco.1996.0018.

S. A. Greibach. A new normal-form theorem for context-free phrase structure grammars. *Journal of the ACM*, vol. 12(1), pp. 42–52, 1965. doi:10.1145/321250.321254.

M. A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, 1978. ISBN 0201029553.

D. Kozen, J. Tiuryn. Logics of programs. In: *Handbook of Theoretical Computer Science*, vol. B: Formal Models and Semantics, chap. 14, pp. 789–840. Elsevier, 1990. doi:10.1016/b978-0-444-88074-1.50019-6.

W. Kuich. Semirings and formal power series: Their relevance to formal languages and automata. In: *Handbook of Formal Languages*, vol. 1: Word, Language, Grammar, chap. 9, pp. 609–677. Springer, 1997. doi:10.1007/978-3-642-59136-5_9.

W. Kuich, A. Salomaa. *Semirings, Automata, Languages*, EATCS Monographs on Theoretical Computer Science, vol. 5. Springer, 1986. doi:10.1007/978-3-642-69959-7.

C. Lautemann, T. Schwentick, D. Thérien. Logics for context-free languages. In: *Computer Science Logic (CSL 1994)*, LNCS, vol. 933, pp. 205–216. Springer, 1994. doi:10.1007/BFb0022257.

Y. Lei, F. Song, W. Liu, M. Zhang. On the complexity of ω-pushdown automata. *Science China Information Sciences*, vol. 60(11), p. 112102, 2017. doi:10.1007/s11432-016-9026-x.

Ch. Mathissen. Weighted logics for nested words and algebraic formal power series. In: *International Colloquium on Automata, Languages, and Programming (ICALP 2008)*, LNCS, vol. 5126, pp. 221–232. Springer, 2008. doi:10.1007/978-3-540-70583-3_19.

K. L. McMillan. *Symbolic Model Checking*. Kluwer Academic Publishers, 1993. doi:10.1007/978-1-4615-3190-6.

R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, vol. 9(5), pp. 521–530, 1966. doi:10.1016/S0019-9958(66)80013-X.

D. E. Muller. Infinite sequences and finite machines. In: *Symposium on Switching Circuit Theory and Logical Design (SWCT 1963)*, pp. 3–16. IEEE, 1963. doi:10.1109/SWCT.1963.8.

M. Nivat. Transductions des langages de Chomsky. *Annales de l'Institut Fourier*, vol. 18(1), pp. 339–455, 1968. doi:10.5802/aif.287.

M. O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, vol. 141, pp. 1–35, 1969. doi:10.2307/1995086.

A. Salomaa, M. Soittola. *Automata-Theoretic Aspects of Formal Power Series*. Texts and Monographs in Computer Science. Springer, 1978. doi:10.1007/978-1-4612-6264-0.

J. W. Thatcher, J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, vol. 2(1), pp. 57–81, 1968. doi:10.1007/BF01691346.

W. Thomas. Automata on infinite objects. In: *Handbook of Theoretical Computer Science*, vol. B: Formal Models and Semantics, chap. 4, pp. 133–191. Elsevier, 1990. doi:10.1016/B978-0-444-88074-1.50009-3.

B. A. Trakhtenbrot. Finite automata and the logic of single-place predicates. *Doklady Akademii Nauk*, vol. 140(2), pp. 326–329, 1961. In Russian.
URL `http://mi.mathnet.ru/dan25511`

H. Vogler, M. Droste, L. Herrmann. A weighted MSO logic with storage behaviour and its Büchi-Elgot-Trakhtenbrot theorem. In: *Language and Automata Theory and Applications* (*LATA 2016*), LNCS, vol. 9618, pp. 127–139. Springer, 2016. doi:10.1007/978-3-319-30000-9_10.

W. M. Waite, G. Goos. *Compiler Construction*. Texts and Monographs in Computer Science. Springer, 1984. doi:10.1007/978-1-4612-5192-7.

N. Wirth. *Compiler Construction*. International computer science series. Addison-Wesley, 1996. ISBN 978-0-201-40353-4.

Bibliographical Description

This thesis is based on the following publications.

M. Droste, S. Dziadek, W. Kuich. Logic for ω-pushdown automata. *Information and Computation*, 2020a. Special issue on "Weighted Automata", Accepted for publication.

M. Droste, S. Dziadek, W. Kuich. Weighted simple reset pushdown automata. *Theoretical Computer Science*, vol. 777, pp. 252–259, 2019a. doi:10.1016/j.tcs.2019.01.016.

M. Droste, S. Dziadek, W. Kuich. Greibach normal form for ω-algebraic systems and weighted simple ω-pushdown automata. In: *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, LIPIcs, vol. 150, pp. 38:1–38:14, 2019b. doi:10.4230/LIPIcs.FSTTCS.2019.38.

M. Droste, S. Dziadek, W. Kuich. Nivat-theorem and logic for weighted pushdown automata on infinite words. In: *Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, LIPIcs, vol. 182, pp. 44:1–44:14, 2020b. doi:10.4230/LIPIcs.FSTTCS.2020.44.

M. Droste, S. Dziadek, W. Kuich. Greibach normal form for ω-algebraic systems and weighted simple ω-pushdown automata, 2020c. Submitted, arXiv:2007.08866.

Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbständig und ohne unzulässige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinngemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden, und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialen oder erbrachten Dienstleistungen als solche gekennzeichnet.

(Ort, Datum)

(Unterschrift)