Universität Leipzig

Institut für Informatik

Models for Quantitative Distributed Systems and Multi-Valued Logics

Master Thesis

Leipzig, September 2010

Author — B.Sc. Martin Huschenbett
Master Student in Computer Science
Leipzig University

Supervisor — Prof. Dr. Manfred Droste
Institute of Computer Science
Leipzig University

Abstract

We investigate weighted asynchronous cellular automata with weights in valuation monoids. These automata form a distributed extension of weighted finite automata and allow us to model concurrency. Valuation monoids are abstract weight structures that include semirings and (non-distributive) bounded lattices but also offer the possibility to model average behaviors. We prove that weighted asynchronous cellular automata and weighted finite automata which satisfy an I-diamond property are equally expressive. Depending on the properties of the valuation monoid, we characterize this expressiveness by certain syntactically restricted fragments of weighted MSO logics. Finally, we define the quantitative model-checking problem for distributed systems and show how it can be reduced to the corresponding problem for sequential systems.

Contents

1 Introduction

During the last decades, automata, logics, and the connection between them turned out to be very useful tools in the modeling and verification of computer systems. In this setting, an automaton $\mathcal{A}$ is used to describe certain aspects of the system under consideration and the property that shall be verified is expressed by some logical formula φ. The model-checking problem is to decide whether all behaviors of $\mathcal{A}$ satisfy the specification encoded by φ. The earliest result in this field was shown by Büchi and Elgot [5, 22] and states that finite automata and monadic second-order logics (MSO logics) over words are equally expressive. The goal of this thesis is to generalize that approach to verification in two directions. First, we extend it beyond a boolean setting by taking quantitative properties into account. Second, we consider systems which are not necessarily sequential but can also be distributed and concurrent.

For modeling and verifying quantitative systems, automata and logics assign values from rich structures to behaviors. These values can, e.g., be the probability of acceptance, describe the consumption of some resource, or represent a reward, respectively. Those new requirements led to several specialized extensions of classical finite automata, like probabilistic automata [6, 38], timed automata [1], and lattice automata [30].

A more generic approach for modeling quantitative systems is provided by the theory of weighted automata [14, 29]. A weighted automaton is essentially a finite automaton with the additional feature that weights from an arbitrary semiring are assigned to each transition. Such an automaton assigns an element from the semiring to each behavior which is computed using the addition and multiplication of the semiring. There is a well developed theory of weighted automata and the first prominent result was established already in the 1960s by Schützenberger [39]. However, weighted logics were not taken into account for a long time. This gap on the specification side of weighted model checking was closed by Droste and Gastin [11, 12] some years ago. Their result characterizes the expressiveness of weighted automata by means of weighted MSO logics and hence generalizes the results of Büchi and Elgot. Nowadays, semiring weighted logics are still an ongoing subject of research [4, 13, 17, 18].

Due to the very general nature of semirings many quantitative systems can be modeled by semiring weighted automata. However, the interest in quantitative aspects which do not fit into this semiring framework grew for several years. On

the one hand, these are aspects like the average consumption of some resource or longtime highs and lows of some cost or output [7]. On the other hand, automata with weights from – potentially non-distributive – bounded lattices were investigated [19, 20]. Such weight structures are used, e.g., in multi-valued logics [27, 33] and quantum logics [3]. Recently, Droste and Meinecke [15] introduced valuation monoids to capture this variety of possible weight structures within one uniform framework. They studied weighted automata as well as weighted logics over such structures and investigated their relationship, culminating in a result similar to Droste and Gastin's [13].

All the automaton models mentioned so far have in common that they assign values to words. This is well suited for modeling sequential systems but improper for distributed systems. A valuable concept for modeling behaviors in a concurrent setting are traces as introduced by Mazurkiewicz [34]. The corresponding unweighted automaton model are Zielonka's asynchronous cellular automata [41]. Thomas [40] generalized the result of Büchi and Elgot by proving that the class of trace languages acceptable by these automata coincides with those languages definable in MSO logics. Later on, this connection was extended to infinite traces by Ebinger and Muscholl [21]. Both results opened the way for qualitative model-checking of distributed systems.

First steps towards modeling quantitative distributed systems by means of automata were taken by Droste and Gastin [10]. Although their automaton model was not distributed it turned out to be very important for the subsequent research. Weighted asynchronous cellular automata with weights from a commutative semiring were introduced and studied by Kuske [31]. Soon afterwards, Meinecke [35] extended the theorems of Büchi and Elgot, Thomas, and Droste and Gastin to a semiring weighted and concurrent setting. Combining the results of Kuske and Meinecke [23] allows one to reduce quantitative model-checking of distributed systems to that of sequential systems.

The situation described above naturally raises the question whether similar results hold true for quantitative model-checking of distributed systems over valuation monoids. Since this issue has not been considered yet, we will investigate it in this thesis.

The main results are as follows. First, we present weighted asynchronous cellular automata as a model for quantitative distributed systems. Moreover, we define weighted $\mathfrak{I}$-diamond automata for modeling the interleaving behavior of such systems and show that both kinds of weighted automata are equally expressive. Second, we introduce weighted MSO logics as a formalism to write specifications for quantitative distributed systems. The main theorem of this thesis characterizes the expressiveness of weighted asynchronous cellular automata by certain syntactically restricted fragments of this logics. The choice of the fragment depends on

the properties of the valuation monoid. Thus, we provide a joint extension of the results of Droste and Gastin [13], Kuske and Meinecke [23, 31, 35], and Droste and Meinecke [15]. At last, we define the quantitative model-checking problem for distributed systems. Using the fact that all proofs throughout the thesis are effective, we show how it can be reduced to the quantitative model-checking problem for sequential systems.

2 Traces and asynchronous cellular automata

In this chapter we give the necessary background in trace theory needed for this thesis. For a more general overview we refer the reader to [8, 9]. In contrast to the algebraic approach taken there, we introduce traces as labeled partial orders. This view is made explicit in [24, 40]. Moreover, we present asynchronous cellular automata and $\mathfrak{I}$-diamond automata as devices for recognizing trace languages.

2.1 Σ-posets and traces

Let Σ be an alphabet, i.e., a finite, non-empty set of actions. A Σ-*poset* is a triplet $s = (V, \sqsubseteq, \lambda)$ where V some arbitrary set, $\sqsubseteq$ is a partial order on V, and $\lambda\colon V \to \Sigma$ is a labeling function. *We agree that throughout the whole thesis, all considered Σ-posets are finite and not empty. As usual, we will also identify isomorphic structures.*

To model the architecture of a distributed system we further fix an *architecture graph* $(\mathfrak{L}, \mathfrak{D})$ where $\mathfrak{L}$ is a finite, non-empty set of *locations* and $\mathfrak{D}$ is a symmetric and reflexive relation on $\mathfrak{L}$, called *dependence relation*. For some $\ell \in \mathfrak{L}$ we denote with

$$\mathfrak{D}(\ell) = \{\, m \in \mathfrak{L} \mid (\ell, m) \in \mathfrak{D} \,\}$$

the set of all locations depending on ℓ. *For the rest of this thesis, we fix the graph $(\mathfrak{L}, \mathfrak{D})$.*

A *distributed alphabet* is a pair (Σ, lc) consisting of an alphabet Σ and a surjective *location mapping* $\mathrm{lc}\colon \Sigma \to \mathfrak{L}$. For any $\ell \in \mathfrak{L}$ we denote with

$$\Sigma_\ell = \{\, a \in \Sigma \mid \mathrm{lc}(a) = \ell \,\}$$

the set of actions assigned to location ℓ. Since lc is surjective, $(\Sigma_\ell)_{\ell \in \mathfrak{L}}$ is an $\mathfrak{L}$-indexed family of mutually disjoint, non-empty alphabets which completely determines the pair (Σ, lc). Thus, some authors define distributed alphabets using such families, c.f., [23, 31]. *Usually, we will omit the location mapping* lc *if it is clear from the context and denote the distributed alphabet solely with Σ.*

Let $t = (V, \sqsubseteq, \lambda)$ be some Σ-poset. We define a map $\mathrm{lc}_t\colon V \to \mathfrak{L}$ as $\mathrm{lc}_t = \mathrm{lc} \circ \lambda$. We call t a *trace* over Σ if for all $x, y \in V$ the following two conditions are satisfied:

(1) if $(\mathrm{lc}_t(x), \mathrm{lc}_t(y)) \in \mathfrak{D}$, then $x \sqsubseteq y$ or $y \sqsubseteq x$,

(2) if $x \sqsubset y$ and there is no $z \in V$ such that $x \sqsubset z \sqsubset y$, then $(\mathrm{lc}_t(x), \mathrm{lc}_t(y)) \in \mathfrak{D}$.

The set of all traces is denoted with $\mathbb{T}(\Sigma)$ and subsets $L \subseteq \mathbb{T}(\Sigma)$ are called *trace languages*.

To each $w = a_1 \dots a_n \in \Sigma^+$ we assign a trace $\mathrm{trc}(w) = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ as follows:

- $V = \{\, 1, \dots, n \,\}$,
- $\sqsubseteq$ is the reflexive and transitive closure of

$$E = \{\, (i,j) \in V \times V \mid i < j \text{ and } (\mathrm{lc}(a_i), \mathrm{lc}(a_j)) \in \mathfrak{D} \,\},$$

  and
- $\lambda(i) = a_i$.

In this way, we obtain a surjective map $\mathrm{trc}\colon \Sigma^+ \to \mathbb{T}(\Sigma)$. *If we introduce a trace as* $\mathrm{trc}(w) = (V, \sqsubseteq, \lambda)$ *for some* $w \in \Sigma^+$*, for simplicity we will assume that* V*,* $\sqsubseteq$*, and* λ *are constructed exactly like above.*

Two words $u, v \in \Sigma^+$ are called *trace equivalent* if $\mathrm{trc}(u) = \mathrm{trc}(v)$. We conclude this section by characterizing trace equivalence on the word level. The *independence relation* $\mathfrak{I}$ on Σ is defined as

$$\mathfrak{I} = \{\, (a,b) \in \Sigma \times \Sigma \mid (\mathrm{lc}(a), \mathrm{lc}(b)) \notin \mathfrak{D} \,\}.$$

Clearly, $\mathfrak{I}$ is symmetric and irreflexive. Moreover, the reflexive and transitive closure of the relation

$$\{\, (xaby, xbay) \mid x, y \in \Sigma^*, (a,b) \in \mathfrak{I} \,\}.$$

on Σ^+ is denoted with $\sim_{\mathfrak{I}}$. From trace theory it is well known that $u, v \in \Sigma^+$ are trace equivalent if and only if $u \sim_{\mathfrak{I}} v$. Intuitively, two words are trace equivalent iff one can be obtained from the other by repeatedly interchanging adjacent independent letters.

2.2 Asynchronous cellular automata

In this section we give a brief overview of asynchronous cellular automata. They were introduced by Zielonka [41] and can be regarded as a distributed extension of classical (non-deterministic) finite automata. Moreover, they turned out to be a very natural automaton model for traces.

An *asynchronous cellular automaton* over Σ (*ACA* for short) is a tuple $\mathcal{A} = ((Q_\ell)_{\ell \in \mathfrak{L}}, I, (T_\ell)_{\ell \in \mathfrak{L}}, F)$ where

- $(Q_\ell)_{\ell\in\mathfrak{L}}$ is a family of mutually disjoint finite sets Q_ℓ of local states for each $\ell \in \mathfrak{L}$,
- $T_\ell \subseteq \prod_{m\in\mathfrak{D}(\ell)} Q_m \times \Sigma_\ell \times Q_\ell$ is a local transition relation for any $\ell \in \mathfrak{L}$,
- $I, F \subseteq \prod_{\ell\in\mathfrak{L}} Q_\ell$ are sets of global initial and final states, respectively.

The ACA $\mathcal{A}$ is *deterministic* if I is a singleton and T_ℓ is the graph of a function $\prod_{m\in\mathfrak{D}(\ell)} Q_m \times \Sigma_\ell \to Q_\ell$ for each $\ell \in \mathfrak{L}$.

Let $t = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ be a trace. Due to the reflexivity of $\mathfrak{D}$ and condition (1) for traces, for each $\ell \in \mathfrak{L}$ and any $U \subseteq V$ the set $U \cap \mathrm{lc}_t^{-1}(\ell)$ is totally ordered by $\sqsubseteq$. Hence, if this set is not empty, it contains a largest element $\partial_\ell(U)$. Moreover, for $x \in V$ we define

$$\Downarrow x = \{\, y \in V \mid y \sqsubset x \,\}.$$

A *run* of $\mathcal{A}$ on t is a pair $\rho = (\iota, r)$ consisting of a global state $\iota \in \prod_{\ell\in\mathfrak{L}} Q_\ell$ and a mapping $r\colon V \to \bigcup_{\ell\in\mathfrak{L}} Q_\ell$ satisfying $r(x) \in Q_{\mathrm{lc}_t(x)}$ for any $x \in V$. The idea behind is the following: before executing the run ρ, the automaton in each location $m \in \mathfrak{L}$ is in its initial state ι_m. When executing an event $x \in V$, the automaton in location $\mathrm{lc}_t(x)$ reads the current states $\mathrm{read}_m(\rho, x)$ of the automata in locations $m \in \mathfrak{D}(\mathrm{lc}_t(x))$ and moves to its local state $r(x)$. Formally,

$$\mathrm{read}_m(\rho, x) = \begin{cases} r(\partial_m(\Downarrow x)) & \text{if } \partial_m(\Downarrow x) \text{ is defined,} \\ \iota_m & \text{otherwise,} \end{cases}$$

and

$$\mathrm{read}_{\mathfrak{D}(\mathrm{lc}_t(x))}(\rho, x) = \big(\mathrm{read}_m(\rho, x)\big)_{m\in\mathfrak{D}(\mathrm{lc}_t(x))}.$$

At the end of the run, the automaton in location $m \in \mathfrak{L}$ is in state $\mathrm{final}_m(\rho)$ defined as

$$\mathrm{final}_m(\rho) = \begin{cases} r(\partial_m(V)) & \text{if } \partial_m(V) \text{ is defined,} \\ \iota_m & \text{otherwise,} \end{cases}$$

and we put

$$\mathrm{final}(\rho) = \big(\mathrm{final}_m(\rho)\big)_{m\in\mathfrak{L}}.$$

The whole run ρ is said to be *successful* if $\iota \in I$,

$$(\mathrm{read}_{\mathfrak{D}(\mathrm{lc}_t(x))}(\rho, x), \lambda(x), r(x)) \in T_{\mathrm{lc}_t(x)}$$

for all $x \in V$, and $\mathrm{final}(\rho) \in F$. The set of all successful runs of $\mathcal{A}$ on t is denoted with $\mathrm{succ}(\mathcal{A}, t)$. The language *recognized* by $\mathcal{A}$ is the set of all traces which admit a successful run, i.e.,

$$L_{\mathrm{t}}(\mathcal{A}) = \{\, t \in \mathbb{T}(\Sigma) \mid \mathrm{succ}(\mathcal{A}, t) \neq \emptyset \,\}.$$

Finally, a trace language $L \subseteq \mathbb{T}(\Sigma)$ is called *recognizable* if it is recognized by some asynchronous cellular automaton over Σ.

2.3 Finite and $\mathfrak{I}$-diamond automata

Deterministic and non-deterministic finite automata are well known automaton models for word languages. If they satisfy one additional condition – the so called $\mathfrak{I}$-diamond property – they can be used in the field of trace theory as well.

A *finite automaton* over Σ (*FA* for short) is a tuple $\mathcal{A} = (Q, I, T, F)$ where

- Q is a finite set of states,
- $T \subseteq Q \times \Sigma \times Q$ is the transition relation, and
- $I, F \subseteq Q$ are sets of initial and final states, respectively.

For a word $w = a_1 \dots a_n \in \Sigma^+$ a *run* of $\mathcal{A}$ on w is a tuple $\sigma = (q_0, \dots, q_n) \in Q^{n+1}$. We call σ *successful* if $q_0 \in I$, $(q_{i-1}, a_i, q_i) \in T$ for all $i = 1, \dots, n$, and $q_n \in F$. The set of all successful runs of $\mathcal{A}$ on w is denoted with $\mathrm{succ}(\mathcal{A}, w)$. The language *recognized* by $\mathcal{A}$ is the set of all words which admit a successful run, i.e.,

$$L_{\mathrm{w}}(\mathcal{A}) = \{\, w \in \Sigma^+ \mid \mathrm{succ}(\mathcal{A}, w) \neq \emptyset \,\}.$$

A word language $L \subseteq \Sigma^+$ is called *recognizable* if it is recognized by some finite automaton over Σ.

An $\mathfrak{I}$-*diamond automaton* over Σ is an FA $\mathcal{A} = (Q, I, T, F)$ over Σ which has the $\mathfrak{I}$-*diamond property*, i.e., for all $p, q, r \in Q$ and $(a, b) \in \mathfrak{I}$ with $(p, a, q), (q, b, r) \in T$ there is some $q' \in Q$ such that $(p, b, q'), (q', a, r) \in T$. In this situation it is well known, that for all $u \sim_{\mathfrak{I}} v$ we have $u \in L_{\mathrm{w}}(\mathcal{A})$ iff $v \in L_{\mathrm{w}}(\mathcal{A})$. Thus, for any trace $t \in \mathbb{T}(\Sigma)$ the automaton $\mathcal{A}$ accepts either all words from $\mathrm{trc}^{-1}(t)$ or none of them. In this way, we can understand an $\mathfrak{I}$-diamond automaton as device for recognizing trace languages.

The connection between both automaton models for traces, ACAs and $\mathfrak{I}$-diamond automata, was established by Zielonka:

Theorem 2.1 (Zielonka [42])**.** *Let $L \subseteq \mathbb{T}(\Sigma)$ be trace language. The following are equivalent:*

(1) L is recognizable,

(2) L is recognized by some deterministic ACA over Σ,

(3) $\mathrm{trc}^{-1}(L)$ is recognized by some $\mathfrak{I}$-diamond automaton over Σ, and

(4) $\mathrm{trc}^{-1}(L)$ is recognizable.

At the end, we want to mention that there are other formalisms to characterize the class of recognizable trace languages, namely algebraic recognizability, c-rational expressions, loop-connected automata, and MSO logics. However, only the the latter two will play a role in this thesis.

3 Weighted asynchronous cellular automata

So far, we can use asynchronous cellular automata to model qualitative properties of distributed systems. In this chapter we want to extend this automaton model in a way that it can express quantitative aspects of such systems. Therefore, we introduce weighted asynchronous cellular automata which essentially are ACAs where a weight is assigned to each transition. They can be regarded as devices that compute a weight for each trace by combining these transition weights in a suitable manner. Hence, the underlying weight structure must essentially provide two operations, one for combining all transition weights of a run and another for resolving non-determinism, i.e., combining the weights of all successful runs a trace admits. Recently, Droste and Meinecke [15] introduced valuation monoids for this purpose.

3.1 Valuation monoids

A *valuation monoid* is an algebraic structure $(\mathcal{D}, +, \mathrm{Val}, \mathbf{0})$ where

- $(\mathcal{D}, +, \mathbf{0})$ is a commutative monoid,
- $\mathrm{Val}\colon \mathcal{D}^+ \to \mathcal{D}$ is a total map called *valuation function*,
- $\mathrm{Val}(d) = d$ for all $d \in \mathcal{D}$, and
- $\mathrm{Val}(d_1, \ldots, d_n) = \mathbf{0}$ whenever $d_i = \mathbf{0}$ for some $i \in \{\, 1, \ldots, n \,\}$.

The intention is that $+$ resolves non-determinism whereas the valuation function Val combines the transition weights of a run.

Example 3.1. The following structures are examples for valuation monoids:

(1) $(\mathbb{Q} \cup \{\, -\infty \,\}, \max, \mathrm{avg}, -\infty)$ and $(\mathbb{Q} \cup \{\, \infty \,\}, \min, \mathrm{avg}, \infty)$ where

$$\mathrm{avg}(d_1, \ldots, d_n) = \frac{d_1 + \cdots + d_n}{n},$$

(2) $(K, +, \Pi, \mathbf{0})$ where $(K, +, \cdot, \mathbf{0}, \mathbf{1})$ is a semiring and

$$\Pi(d_1, \dots, d_n) = d_1 \cdots d_n,$$

(3) $(\mathcal{L}, \vee, \inf, \bot)$ where $(\mathcal{L}, \vee, \wedge, \bot, \top)$ is a bounded lattice and

$$\inf(d_1, \dots, d_n) = d_1 \wedge \cdots \wedge d_n.$$

For more examples we refer the reader to [15].

Since the transitions of an asynchronous cellular automaton are not executed in some specific linear order, valuation monoids need to have an additional property – called order independence – to fit into our distributed framework.

Definition 3.2. A valuation monoid $(\mathcal{D}, +, \mathrm{Val}, \mathbf{0})$ is *order independent* if

$$\mathrm{Val}(d_1, \dots, d_n) = \mathrm{Val}(d'_1, \dots, d'_n)$$

for all $n \geq 1$ and $(d_1, \dots, d_n), (d'_1, \dots, d'_n) \in \mathcal{D}^+$ where the first sequence is a permutation of the second.

Obviously, the valuation monoids from Example 3.1 (1) and (3) are order independent, whereas this is valid for (2) iff K is commutative.

3.2 Weighted asynchronous cellular automata

For the rest of this chapter we fix a distributed alphabet (Σ, lc) *and a valuation monoid* $(\mathcal{D}, +, \mathrm{Val}, \mathbf{0})$*. In this section we also assume that* $\mathcal{D}$ *is order independent.* First, we define weighted asynchronous cellular automata.

Definition 3.3. A *weighted asynchronous cellular automaton* over Σ and $\mathcal{D}$ (*wACA* for short) is a tuple $\mathcal{A} = ((Q_\ell)_{\ell \in \mathfrak{L}}, I, (T_\ell)_{\ell \in \mathfrak{L}}, F, (\gamma_\ell)_{\ell \in \mathfrak{L}})$ where

- $((Q_\ell)_{\ell \in \mathfrak{L}}, I, (T_\ell)_{\ell \in \mathfrak{L}}, F)$ is an ACA over Σ and
- $\gamma_\ell \colon T_\ell \to \mathcal{D}$ is a *transition weight function* for each $\ell \in \mathfrak{L}$.

For a trace $t = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ (successful) runs ρ of $\mathcal{A}$ on t, $\mathrm{read}(\rho, x)$, $\mathrm{final}(\rho)$, and $\mathrm{succ}(\mathcal{A}, t)$ are defined like for (unweighted) ACAs. The *weight* of a successful run $\rho = (\iota, r)$ of $\mathcal{A}$ on t is defined as

$$\gamma(\rho, t) = \mathrm{Val}\Big(\big(\gamma_{\mathrm{lc}_t(x)}(\mathrm{read}_{\mathfrak{D}(\mathrm{lc}_t(x))}(\rho, x), \lambda(x), r(x))\big)_{x \in V}\Big).$$

Notice that the value of $\gamma(\rho, t)$ does not depend on the order in which V is enumerated since $\mathcal{D}$ is order independent.

Definition 3.4. Let $\mathcal{A}$ be a wACA over Σ and $\mathcal{D}$. The *behavior* of $\mathcal{A}$ is the function $\|\mathcal{A}\|_{\mathrm{t}} \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ defined as

$$\|\mathcal{A}\|_{\mathrm{t}}(t) = \sum_{\rho \in \mathrm{succ}(\mathcal{A},t)} \gamma(\rho, t).$$

From the definition above, we see that maps $\mathbb{T}(\Sigma) \to \mathcal{D}$ are subject to our interest. Thus, they get a concise name and are called *trace series* from now on. In particular, we are interested those trace series that can occur as the behavior of some wACA.

Definition 3.5. A trace series $S \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ is called *recognizable* if there exists a wACA $\mathcal{A}$ such that

$$S = \|\mathcal{A}\|_{\mathrm{t}}.$$

A common pattern in the investigation of trace systems is to consider some underlying word structure and to lift statements about word systems to the trace level. In this thesis we take the same approach and extend a recent result of Droste and Meinecke [15].

3.3 Weighted finite and $\mathfrak{I}$-diamond automata

First, we need to reintroduce the weighted automaton model from Droste and Meinecke [15]. A *weighted finite automaton* over Σ and $\mathcal{D}$ (*wFA* for short) is a tuple $\mathcal{A} = (Q, I, T, F, \gamma)$ where

- (Q, I, T, F) is an FA over Σ and
- $\gamma \colon T \to \mathcal{D}$ is the *transition weight function.*

For some word $w = a_1 \dots a_n \in \Sigma^+$ (successful) runs of $\mathcal{A}$ on w and $\mathrm{succ}(\mathcal{A}, w)$ are defined like for (unweighted) FAs. For a successful run $\sigma = (q_0, \dots, q_n)$ of $\mathcal{A}$ on w the *weight* of σ is defined as

$$\gamma(\sigma, w) = \mathrm{Val}\Big(\big(\gamma(q_{i-1}, a_i, q_i)\big)_{i=1,\dots,n} \Big).$$

The *(word) behavior* $\|\mathcal{A}\|_{\mathrm{w}} \colon \Sigma^+ \to \mathcal{D}$ of $\mathcal{A}$ is computed by

$$\|\mathcal{A}\|_{\mathrm{w}}(w) = \sum_{\sigma \in \mathrm{succ}(\mathcal{A},w)} \gamma(\sigma, w).$$

Similar to the situation for traces, a map $\Sigma^+ \to \mathcal{D}$ is called *word series* and a word series $S \colon \Sigma^+ \to \mathcal{D}$ is *recognizable* if there exists a wFA $\mathcal{A}$ such that

$$S = \|\mathcal{A}\|_{\mathrm{w}}.$$

For the rest of this section, we assume that the valuation monoid $\mathcal{D}$ is order independent. As explained earlier, finite automata having the $\mathfrak{I}$-diamond property can be regarded as devices for recognizing trace languages. Thus, weighted finite automata satisfying some $\mathfrak{I}$-diamond property were investigated [10,23,25]. Moreover, Kuske [31] introduced the stronger notion of $\mathfrak{I}$-consistency. Next, we adopt these concepts to our setting with weights in a valuation monoid.

For some wFA $\mathcal{A} = (Q, I, T, F, \gamma)$ over Σ and for all $a, b \in \Sigma$ and $p, r \in Q$ we put

$$Q_{p,r}^{a,b} = \{\, q \in Q \mid (p, a, q) \in T \text{ and } (q, b, r) \in T \,\}.$$

Definition 3.6. Let $\mathcal{A} = (Q, I, T, F, \gamma)$ be a wFA.

- $\mathcal{A}$ is called $\mathfrak{I}$-*consistent* if for all $p, q, r \in Q$ and $(a, b) \in \mathfrak{I}$ the following three conditions are met:
  (1) if $(p, a, q), (q, b, r) \in T$ then there exists some $q' \in Q$ such that (p, b, q'), $(q', a, r) \in T$,
  $$\gamma(p, a, q) = \gamma(q', a, r), \quad \text{and} \quad \gamma(q, b, r) = \gamma(p, b, q'), \tag{3.1}$$
  (2) if $(p, a, q), (q, b, r) \in T$ and $(p, a, q'), (q', b, r) \in T$ for some $q' \in Q$ then $q' = q$, and
  (3) if $(p, a, q), (p, b, r) \in T$ then there is some $s \in Q$ such that $(q, b, s) \in T$.
- $\mathcal{A}$ has the $\mathfrak{I}$-*diamond property* if for all $p, r \in Q$ and $(a, b) \in \mathfrak{I}$ there is an injective mapping $f = f_{p,r}^{a,b} \colon Q_{p,r}^{a,b} \to Q_{p,r}^{b,a}$ such that for any $q \in Q_{p,r}^{a,b}$ we have
  $$\gamma(p, a, q) = \gamma(f(q), a, r) \quad \text{and} \quad \gamma(q, b, r) = \gamma(p, b, f(q)).$$

As already suggested, $\mathfrak{I}$-consistency is a stronger condition than the $\mathfrak{I}$-diamond property.

Lemma 3.7. *Let $\mathcal{A}$ be a wFA. If $\mathcal{A}$ is $\mathfrak{I}$-consistent then $\mathcal{A}$ has the $\mathfrak{I}$-diamond property.*

Proof. Let $\mathcal{A} = (Q, I, T, F, \gamma)$ be an $\mathfrak{I}$-consistent wFA. Consider $p, r \in Q$ and $(a, b) \in \mathfrak{I}$. Due to condition (2) for $\mathfrak{I}$-consistency the set $Q_{p,r}^{a,b}$ is either empty or a singleton. In the first case there is nothing to prove. For the latter case let $Q_{p,r}^{a,b} = \{\, q \,\}$. Condition (1) implies the existence of some $q' \in Q_{p,r}^{b,a}$ satisfying Eq. (3.1). Thus, the choice $f_{p,r}^{a,b}(q) = q'$ shows the claim. □

The following lemma explains how the $\mathfrak{I}$-diamond property relates to the study of trace series.

Lemma 3.8. *Let $\mathcal{A}$ be a wFA having the $\mathfrak{I}$-diamond property. Then*

$$\|\mathcal{A}\|_{\mathrm{w}}(u) = \|\mathcal{A}\|_{\mathrm{w}}(v)$$

holds true for all $u, v \in \Sigma^+$ satisfying $u \sim_{\mathfrak{I}} v$.

Proof. First, recall that $\sim_{\mathfrak{I}}$ is the reflexive and transitive closure of the relation

$$R = \{\, (xaby, xbay) \mid x, y \in \Sigma^*, (a, b) \in \mathfrak{I} \,\}$$

on Σ^+. Thus, it suffices to show $\|\mathcal{A}\|_{\mathrm{w}}(u) = \|\mathcal{A}\|_{\mathrm{w}}(v)$ for all $(u, v) \in R$. Therefore, let $u = xaby$ and $v = xbay$ where $x, y \in \Sigma^*$ and $(a, b) \in \mathfrak{I}$. Moreover, let $m = |x|$ and $n = |y|$. For every successful run $\sigma = (q_0, \ldots, q_{m+n+2})$ of $\mathcal{A}$ on u we define a run

$$f(\sigma) = \big(q_0, \ldots, q_m, f^{a,b}_{q_m,q_{m+2}}(q_{m+1}), q_{m+2}, \ldots, q_{m+n+2}\big)$$

of $\mathcal{A}$ on v. Due to the choice of $f^{a,b}_{q_m,q_{m+2}}$ the run $f(\sigma)$ is also successful and from the order independence of $\mathcal{D}$ we conclude $\gamma(f(\sigma), v) = \gamma(\sigma, u)$. Hence, f is a mapping $f\colon \operatorname{succ}(\mathcal{A}, u) \to \operatorname{succ}(\mathcal{A}, v)$.

Similarly, we define a map $g\colon \operatorname{succ}(\mathcal{A}, v) \to \operatorname{succ}(\mathcal{A}, u)$ by

$$g(q_0, \ldots, q_{m+n+2}) = \big(q_0, \ldots, q_m, f^{b,a}_{q_m,q_{m+2}}(q_{m+1}), q_{m+2}, \ldots, q_{m+n+2}\big).$$

Since for all $p, r \in Q$ the sets $Q^{a,b}_{p,r}$ and $Q^{b,a}_{p,r}$ are finite and the maps $f^{a,b}_{p,r}\colon Q^{a,b}_{p,r} \to Q^{b,a}_{p,r}$ and $f^{b,a}_{p,r}\colon Q^{b,a}_{p,r} \to Q^{a,b}_{p,r}$ are injective, they form a pair of inverse bijections. This carries over to f and g, i.e., f and g are bijections which are inverse to each other. Thus,

$$\begin{aligned}\|\mathcal{A}\|_{\mathrm{w}}(u) &= \sum_{\sigma \in \operatorname{succ}(\mathcal{A},u)} \gamma(\sigma, u) = \sum_{\sigma \in \operatorname{succ}(\mathcal{A},u)} \gamma(f(\sigma), v) \\ &= \sum_{\sigma' \in \operatorname{succ}(\mathcal{A},v)} \gamma(\sigma', v) = \|\mathcal{A}\|_{\mathrm{w}}(v).\end{aligned}$$

□

The assertion of the proposition above ensures that in the following definition the behavior is independent from the choice of $w \in \operatorname{trc}^{-1}(t)$:

Definition 3.9. A *weighted $\mathfrak{I}$-diamond automaton* over Σ over $\mathcal{D}$ is a wFA $\mathcal{A}$ which has the $\mathfrak{I}$-diamond property. The *trace behavior* of $\mathcal{A}$ is the trace series $\|\mathcal{A}\|_{\mathrm{t}}\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ defined by

$$\|\mathcal{A}\|_{\mathrm{t}}(t) = \|\mathcal{A}\|_{\mathrm{w}}(w),$$

where $w \in \operatorname{trc}^{-1}(t)$. A weighted $\mathfrak{I}$-diamond automaton is called *consistent* if it is $\mathfrak{I}$-consistent.

In order to formulate our first new result on recognizable trace series, we need to assign a word series to each trace series. Thus, for every $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ we define a word series $\mathrm{trc}^{-1}(S)\colon \Sigma^+ \to \mathcal{D}$ by

$$\mathrm{trc}^{-1}(S)(w) = S(\mathrm{trc}(w)).$$

Using this notation we obtain for every weighted $\mathfrak{I}$-diamond automaton $\mathcal{A}$ the relation

$$\|\mathcal{A}\|_{\mathrm{w}} = \mathrm{trc}^{-1}\big(\|\mathcal{A}\|_{\mathrm{t}}\big). \tag{3.2}$$

Theorem 3.10. *Let $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series. The following are equivalent:*

(1) S is recognizable,

(2) there exists some consistent weighted $\mathfrak{I}$-diamond automaton $\mathcal{A}$ such that $S = \|\mathcal{A}\|_{\mathrm{t}}$,

(3) there exists some weighted $\mathfrak{I}$-diamond automaton $\mathcal{A}$ such that $S = \|\mathcal{A}\|_{\mathrm{t}}$, and

(4) $\mathrm{trc}^{-1}(S)$ is recognizable.

The implication from (2) to (3) is trivial and the one from (3) to (4) follows immediately from Eq. (3.2). The rest of this chapter is devoted to the proofs of the missing implications from (1) to (2) (Section 3.4) and from (4) to (1) (Section 3.7).

3.4 From trace automata to word automata

For the rest of this chapter, we assume that the valuation monoid $\mathcal{D}$ is order independent. The objective of this section is to prove the proposition below which restates the implication from (1) to (2) of Theorem 3.10.

Proposition 3.11. *Let $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a recognizable trace series. Then there exists a consistent weighted $\mathfrak{I}$-diamond automaton $\mathcal{A}$ such that $S = \|\mathcal{A}\|_{\mathrm{t}}$.*

The main idea of the proof is to construct for every wACA $\mathcal{A}$ an $\mathfrak{I}$-consistent wFA $\mathcal{A}^{\mathrm{w}}$ whose behavior is $\mathrm{trc}^{-1}(\|\mathcal{A}\|_{\mathrm{t}})$. Therefore, let $\mathcal{A} = ((Q_\ell)_{\ell\in\mathfrak{L}}, I, (T_\ell)_{\ell\in\mathfrak{L}}, F, (\gamma_\ell)_{\ell\in\mathfrak{L}})$ be some wACA. We define the wFA $\mathcal{A}^{\mathrm{w}} = (Q, I, T, F, \gamma)$ by letting

$$Q = \prod_{\ell\in\mathfrak{L}} Q_\ell,$$

$$T = \left\{ (p, a, q) \;\middle|\; (p_{\mathfrak{D}(\mathrm{lc}(a))}, a, q_{\mathrm{lc}(a)}) \in T_{\mathrm{lc}(a)} \text{ and } q_\ell = p_\ell \text{ for all } \ell \in \mathfrak{L} \setminus \{\, \mathrm{lc}(a) \,\} \right\},$$

and

$$\gamma(p, a, q) = \gamma_{\mathrm{lc}(a)}(p_{\mathfrak{D}(\mathrm{lc}(a))}, a, q_{\mathrm{lc}(a)}),$$

where $q_X = (q_\ell)_{\ell \in X}$ for all $q \in Q$ and $X \subseteq \mathfrak{L}$.

The rest of this section is devoted to investigating this wFA $\mathcal{A}^{\mathrm{w}}$, culminating in the proof of Proposition 3.11.

Lemma 3.12. *For each wACA $\mathcal{A}$ the wFA $\mathcal{A}^{\mathrm{w}}$ is $\mathfrak{I}$-consistent.*

Proof. Consider $p, q, r \in Q$ and $(a, b) \in \mathfrak{I}$. First, let us assume that (p, a, q), $(q, b, r) \in T$. From the choice of T we conclude $q_\ell = p_\ell$ for all $\ell \neq \mathrm{lc}(a)$ and $r_\ell = q_\ell$ for each $\ell \neq \mathrm{lc}(b)$. Since $\mathrm{lc}(a) \neq \mathrm{lc}(b)$, this implies that q is uniquely determined by p, r, a, and b. Thus, requirement (2) for $\mathfrak{I}$-consistency is met.

In order to verify condition (1) we must construct some $q' \in Q$ such that $(p, b, q'), (q', a, r) \in T$ and Eq. (3.1) is satisfied. We do this by putting

$$q'_m = \begin{cases} r_m & \text{if } m = \mathrm{lc}(b), \\ p_m & \text{otherwise.} \end{cases}$$

First, $\mathrm{lc}(a) \notin \mathfrak{D}(\mathrm{lc}(b))$ and $(q, b, r) \in T$ imply

$$(p_{\mathfrak{D}(\mathrm{lc}(b))}, b, q'_{\mathrm{lc}(b)}) = (q_{\mathfrak{D}(\mathrm{lc}(b))}, b, r_{\mathrm{lc}(b)}) \in T_{\mathrm{lc}(b)},$$

i.e., $(p, b, q') \in T$. Second, $\mathrm{lc}(b) \notin \mathfrak{D}(\mathrm{lc}(a))$ and $(p, a, q) \in T$ analogously imply

$$(q'_{\mathfrak{D}(\mathrm{lc}(a))}, a, r_{\mathrm{lc}(a)}) = (p_{\mathfrak{D}(\mathrm{lc}(a))} a, q_{\mathrm{lc}(a)}) \in T_{\mathrm{lc}(a)},$$

i.e., $(q', a, r) \in T$. The previous two equalities and the definition of γ immediately show Eq. (3.1).

Finally, assume $(p, a, q), (p, b, r) \in T$. To prove that condition (3) is met we must find some $s \in Q$ with $(q, b, s) \in T$. We show that

$$s_m = \begin{cases} r_m & \text{if } m = \mathrm{lc}(b), \\ q_m & \text{otherwise,} \end{cases}$$

is a suitable choice. Since $\mathrm{lc}(a) \notin \mathfrak{D}(\mathrm{lc}(b))$ and $(p, b, r) \in T$ we have

$$(q_{\mathfrak{D}(\mathrm{lc}(b))}, b, s_{\mathrm{lc}(b)}) = (p_{\mathfrak{D}(\mathrm{lc}(b))}, b, r_{\mathrm{lc}(b)}) \in T_{\mathrm{lc}(b)},$$

i.e., $(q, b, s) \in T$. □

Lemma 3.13. *Let $\mathcal{A}$ be a wACA over Σ. Then for each $w \in \Sigma^+$ we have*

$$\|\mathcal{A}^{\mathrm{w}}\|_{\mathrm{w}}(w) = \|\mathcal{A}\|_{\mathrm{t}}(\mathrm{trc}(w)).$$

Proof. We fix some $w = a_1 \dots a_n \in \Sigma$ and let $t = \mathrm{trc}(w) = (V, \sqsubseteq, \lambda)$. First, we observe that

$$\Downarrow i \cap \mathrm{lc}_t^{-1}(\ell) = \{\, j \in V \mid i < j \text{ and } \mathrm{lc}_t(j) = \ell \,\}$$

for all $i \in V$ and $\ell \in \mathfrak{L}$. Moreover, for each run ρ of $\mathcal{A}$ on t and any $i \in V$ we put

$$\mathrm{read}(\rho, i) = (\mathrm{read}_\ell(\rho, i))_{\ell \in \mathfrak{L}}.$$

Now, consider a successful run $\sigma = (q_0, \dots, q_n)$ of $\mathcal{A}^{\mathrm{w}}$ on w. Then $\rho = (q_0, r)$, where $r(i) = (q_i)_{\mathrm{lc}_t(i)}$, is a run of $\mathcal{A}$ on t. Using induction on $i = 1, \dots, n$ we show

$$\mathrm{read}(\rho, i) = q_{i-1}. \tag{3.3}$$

Thus, take some $i \in V$ and $\ell \in \mathfrak{L}$. If $i = 1$ we have $\Downarrow i \cap \mathrm{lc}_t^{-1}(\ell) = \emptyset$ and hence $\mathrm{read}_\ell(\rho, i) = (q_0)_\ell$. For $i > 1$ there are two cases. First, suppose $\mathrm{lc}_t(i-1) = \ell$. Then we have $\partial_\ell(\Downarrow i) = i - 1$ and $\mathrm{read}_\ell(\rho, i) = r(i-1) = (q_{i-1})_\ell$. Second, assume $\mathrm{lc}_t(i-1) \neq \ell$. We obtain

$$\Downarrow i \cap \mathrm{lc}_t^{-1}(\ell) = \Downarrow (i-1) \cap \mathrm{lc}_t^{-1}(\ell)$$

and hence

$$\mathrm{read}_\ell(\rho, i) = \mathrm{read}_\ell(\rho, i-1).$$

By induction we have $\mathrm{read}_\ell(\rho, i-1) = (q_{i-2})_\ell$. Moreover, since $\mathrm{lc}(a_{i-1}) \neq \ell$ we can conclude $(q_{i-2})_\ell = (q_{i-1})_\ell$ from $(q_{i-2}, a_{i-1}, q_{i-1}) \in T$. Combining the last three insights we get $\mathrm{read}_\ell(\rho, i) = (q_{i-1})_\ell$. Thus, we have shown Eq. (3.3). However, we want to exploit these arguments once more to conclude

$$\mathrm{final}(\rho) = q_n. \tag{3.4}$$

Therefore, we introduce an artificial node $n+1$ which shall satisfy $\Downarrow (n+1) = V$. In this way, we get $\mathrm{final}(\rho) = \mathrm{read}(\rho, n+1)$ and

$$\Downarrow (n+1) \cap \mathrm{lc}_t^{-1}(\ell) = V \cap \mathrm{lc}_t^{-1}(\ell) = \{\, j \in V \mid i < n+1 \text{ and } \mathrm{lc}_t(j) = \ell \,\}.$$

This equation justifies a use of the induction step for $i = n+1$ and the claim of Eq. (3.4) follows. Finally, Eqs. (3.3) and (3.4) imply that the run ρ is also successful and satisfies $\gamma(\rho, t) = \gamma(\sigma, w)$. Thus, putting $f(\sigma) = \rho$ defines a map

$$f\colon \mathrm{succ}(\mathcal{A}^{\mathrm{w}}, w) \to \mathrm{succ}(\mathcal{A}, t).$$

Next, we show that f is a bijection. In order to verify surjectivity, consider a successful run $\rho = (\iota, r)$ of $\mathcal{A}$ on t. Using an induction argument very similar to the one above we can easily show that

$$\sigma = (\mathrm{read}(\rho, 1), \dots, \mathrm{read}(\rho, n), \mathrm{final}(\rho))$$

is a successful run of $\mathcal{A}^{\mathrm{w}}$ on w such that $f(\sigma) = \rho$. Concerning injectivity, let $\sigma = (p_0, \dots, p_n)$ and $\sigma' = (q_0, \dots, q_n)$ be two successful runs of $\mathcal{A}^{\mathrm{w}}$ on w such that $f(\sigma) = f(\sigma')$. By induction we prove that $p_i = q_i$ holds true for all $i = 0, \dots, n$. For $i = 0$ this follows immediately from the definition of f and $f(\sigma) = f(\sigma')$. For $i > 0$ we have to show $(p_i)_\ell = (q_i)_\ell$ for any $\ell \in \mathfrak{L}$. For $\ell = \mathrm{lc}_t(i)$ this is implied by $f(\sigma) = f(\sigma')$ as well. For all other values of ℓ it follow from the definition of T and the fact that both runs σ and σ' are successful. Thus, we proved f to be bijective.

In the end, we conclude

$$\begin{aligned}\|\mathcal{A}^{\mathrm{w}}\|_{\mathrm{w}}(w) &= \sum_{\sigma \in \mathrm{succ}(\mathcal{A}^{\mathrm{w}}, w)} \gamma(\sigma, w) = \sum_{\sigma \in \mathrm{succ}(\mathcal{A}^{\mathrm{w}}, w)} \gamma(f(\sigma), t) \\ &= \sum_{\rho \in \mathrm{succ}(\mathcal{A}, t)} \gamma(\rho, t) = \|\mathcal{A}\|_{\mathrm{t}}(t). \qquad \square\end{aligned}$$

Now, we are prepared to prove Proposition 3.11.

Proof of Proposition 3.11. Let $\mathcal{A}$ be a wACA such that $\|\mathcal{A}\|_{\mathrm{t}} = S$. By Lemma 3.12 the wFA $\mathcal{A}^{\mathrm{w}}$ is $\mathfrak{I}$-consistent and hence a consistent weighted $\mathfrak{I}$-diamond automaton. Moreover, by Lemma 3.13 for each $t \in \mathbb{T}(\Sigma)$ and any $w \in \mathrm{trc}^{-1}(t)$ we have

$$\|\mathcal{A}^{\mathrm{w}}\|_{\mathrm{t}}(t) = \|\mathcal{A}^{\mathrm{w}}\|_{\mathrm{w}}(w) = \|\mathcal{A}\|_{\mathrm{t}}(t) = S(t). \qquad \square$$

3.5 Closure properties of recognizable trace series

In this section we define two operations on trace series, namely restriction and projection, and investigate sufficient conditions under which they preserve recognizability. Both operations will play an important role in Section 3.7.

First, let $S \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series and $L \subseteq \mathbb{T}(\Sigma)$ be a trace language. We define a trace series $S{\restriction_L} \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ by

$$S{\restriction_L}(t) = \begin{cases} S(t) & \text{if } t \in L, \\ 0 & \text{otherwise.} \end{cases}$$

Proposition 3.14. *Let $S \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a recognizable trace series and $L \subseteq \mathbb{T}(\Sigma)$ be recognizable trace language. Then, $S{\restriction_L}$ is also recognizable.*

Proof. Let $\mathcal{A}' = ((Q'_\ell)_{\ell \in \mathfrak{L}}, I', (T'_\ell)_{\ell \in \mathfrak{L}}, F', (\gamma'_\ell)_{\ell \in \mathfrak{L}})$ be a wACA with $\|\mathcal{A}'\| = S$ and $\mathcal{A}'' = ((Q''_\ell)_{\ell \in \mathfrak{L}}, I'', (T''_\ell)_{\ell \in \mathfrak{L}}, F'')$ be a deterministic ACA satisfying $L_{\mathrm{t}}(\mathcal{A}'') = L$. We construct a wACA $\mathcal{A} = ((Q_\ell)_{\ell \in \mathfrak{L}}, I, (T_\ell)_{\ell \in \mathfrak{L}}, F, (\gamma_\ell)_{\ell \in \mathfrak{L}})$ such that $\|\mathcal{A}\| = S{\restriction_L}$. Therefore, let $Q_\ell = Q'_\ell \times Q''_\ell$ for each $\ell \in \mathfrak{L}$,

$$I = \{\, (q'_\ell, q''_\ell)_{\ell \in \mathfrak{L}} \mid q' \in I', q'' \in I'' \,\}, \quad \text{and} \quad F = \{\, (q'_\ell, q''_\ell)_{\ell \in \mathfrak{L}} \mid q' \in F', q'' \in F'' \,\}.$$

Moreover, we define T_ℓ by letting $\big((p'_m, p''_m)_{m\in\mathfrak{D}(\ell)}, a, (q'_\ell, q''_\ell)\big) \in T_\ell$ iff

$$\big((p'_m)_{m\in\mathfrak{D}(\ell)}, a, q'\big) \in T' \text{ and } \big((p''_m)_{m\in\mathfrak{D}(\ell)}, a, q''\big)\big) \in T''$$

and put

$$\gamma_\ell\big((p'_m, p''_m)_{m\in\mathfrak{D}(\ell)}, a, (q'_\ell, q''_\ell)\big) = \gamma'_\ell\big((p'_m)_{m\in\mathfrak{D}(\ell)}, a, q'_\ell\big).$$

This completes the construction of $\mathcal{A}$.

It remains to show that $\mathcal{A}$ has the desired behavior. Therefore, we consider some trace $t = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ and a run $\rho = (\iota, r)$ of $\mathcal{A}$ on t. By letting $(\iota'_\ell, \iota''_\ell) = \iota_\ell$ for each $\ell \in \mathfrak{L}$ and $(r'(x), r''(x)) = r(x)$ for any $x \in V$ we obtain runs $\rho' = (\iota', r')$ and $\rho'' = (\iota'', r'')$ of $\mathcal{A}'$ respectively $\mathcal{A}''$ on t. For all $m \in \mathfrak{L}$ and $x \in V$ we obtain

$$\mathrm{read}_m(\rho, x) = (\mathrm{read}_m(\rho', x), \mathrm{read}_m(\rho'', x))$$

and

$$\mathrm{final}_m(\rho) = (\mathrm{final}_m(\rho'), \mathrm{final}_m(\rho'')).$$

Thus, ρ is successful iff both runs ρ' and ρ'' are successful and this case we have $\gamma(\rho, t) = \gamma'(\rho', t)$. Hence, if $t \notin L$ there exists no successful run of $\mathcal{A}$ on t and we obtain $\|\mathcal{A}\|_{\mathrm{t}}(t) = 0$.

Now, let us assume $t \in L$. By letting $f(\rho) = \rho'$ we define a map

$$f\colon \mathrm{succ}(\mathcal{A}, t) \to \mathrm{succ}(\mathcal{A}', t).$$

As already mentioned, we have $\gamma(\rho, t) = \gamma'(f(\rho), t)$ for all $\rho \in \mathrm{succ}(\mathcal{A}, t)$. Since $t \in L$ and $\mathcal{A}''$ is deterministic there is exactly one successful run ρ'' of $\mathcal{A}''$ on t and therefore f is a bijection. Finally, we obtain

$$\begin{aligned}\|\mathcal{A}\|(t) &= \sum_{\rho\in\mathrm{succ}(\mathcal{A},t)} \gamma(\rho, t) = \sum_{\rho\in\mathrm{succ}(\mathcal{A},t)} \gamma'(f(\rho), t) \\ &= \sum_{\rho'\in\mathrm{succ}(\mathcal{A}',t)} \gamma'(\rho', t) = \|\mathcal{A}'\|(t) = S(t).\end{aligned}$$ □

Next, we define and study a projection operation on trace series. Thus, for the rest of this chapter we fix a second distributed alphabet (Γ, lc'). Moreover, let $\pi\colon \Gamma \to \Sigma$ be a *location preserving* mapping, i.e., for all $\tau \in \Gamma$ we have

$$\mathrm{lc}(\pi(\tau)) = \mathrm{lc}'(\tau).$$

For any trace $u = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Gamma)$ over Γ we can consider the Σ-poset $(V, \sqsubseteq, \pi \circ \lambda)$. Since π is location preserving this is a trace over Σ, which we denote with $\pi(u)$. In this way, we extend π to a map $\pi\colon \mathbb{T}(\Gamma) \to \mathbb{T}(\Sigma)$. From the finiteness of $\pi^{-1}(a)$ for

every $a \in \Sigma$ we can conclude that $\pi^{-1}(t)$ is finite for each $t \in \mathbb{T}(\Sigma)$. Thus, for any trace series $S\colon \mathbb{T}(\Gamma) \to \mathcal{D}$ we can define another trace series $\pi(S)\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ by

$$\pi(S)(t) = \sum_{u \in \pi^{-1}(t)} S(u).$$

Proposition 3.15. *Let $\pi\colon \Gamma \to \Sigma$ be a location preserving map and $S\colon \mathbb{T}(\Gamma) \to \mathcal{D}$ be a recognizable trace series. Then $\pi(S)$ is also a recognizable trace series.*

Proof. Let $\mathcal{A} = ((Q_\ell)_{\ell\in\mathfrak{L}}, I, (T_\ell)_{\ell\in\mathfrak{L}}, F, (\gamma_\ell)_{\ell\in\mathfrak{L}})$ be a wACA over Γ with $\|\mathcal{A}\|_{\mathrm{t}} = S$. We construct a wACA $\mathcal{A}' = ((Q'_\ell)_{\ell\in\mathfrak{L}}, I', (T'_\ell)_{\ell\in\mathfrak{L}}, F', (\gamma'_\ell)_{\ell\in\mathfrak{L}})$ over Σ such that $\|\mathcal{A}'\|_{\mathrm{t}} = \pi(S)$. First, we fix some arbitrary $\upsilon^0_\ell \in \Gamma_\ell$ for each $\ell \in \mathfrak{L}$. Moreover, we let $Q'_\ell = Q_\ell \times \Gamma_\ell$ for any $\ell \in \mathfrak{L}$,

$$I' = \left\{ (q_\ell, \upsilon^0_\ell)_{\ell\in\mathfrak{L}} \;\middle|\; q \in I \right\}, \quad \text{and} \quad F' = \left\{ (q_\ell, \tau_\ell)_{\ell\in\mathfrak{L}} \mid q \in F \right\}.$$

We define T'_ℓ by letting $\big((p_m, \upsilon_m)_{m\in\mathfrak{D}(\ell)}, a, (q_\ell, \tau_\ell)\big) \in T'_\ell$ iff

$$\big((p_m)_{m\in\mathfrak{D}(\ell)}, \tau_\ell, q_\ell\big) \in T_\ell \text{ and } \pi(\tau_\ell) = a$$

and put

$$\gamma'_\ell\big((p_m, \upsilon_m)_{m\in\mathfrak{D}(\ell)}, a, (q_\ell, \tau_\ell)\big) = \gamma_\ell\big((p_m)_{m\in\mathfrak{D}(\ell)}, \tau_\ell, q_\ell\big).$$

This completes the construction of $\mathcal{A}'$.

It remains to show that $\mathcal{A}'$ has the desired behavior. Therefore, we consider a trace $t = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ and a successful run $\rho' = (\iota', r')$ of $\mathcal{A}'$ on t. Let $(\iota_\ell, \tau_\ell) = \iota'_\ell$ for all $\ell \in \mathfrak{L}$ and $(r(x), \mu(x)) = r'(x)$ for each $x \in V$. Then $u = (V, \sqsubseteq, \mu)$ is a trace over Γ and $\rho = (\iota, r)$ is a successful run of $\mathcal{A}$ on u with $\gamma(\rho, u) = \gamma'(\rho', t)$. Moreover, we have $\tau_\ell = \upsilon^0_\ell$ for all $\ell \in \mathfrak{L}$ and $\pi(\mu(x)) = \lambda(x)$ for any $x \in V$. Obviously, the last condition is equivalent to $\pi(u) = t$. Hence, by putting $f(\rho') = (\rho, u)$ we obtain a mapping

$$f\colon \operatorname{succ}(\mathcal{A}', t) \to \bigcup_{u\in\pi^{-1}(t)} \operatorname{succ}(\mathcal{A}, u) \times \{ u \}.$$

Conversely, let $u = (V, \sqsubseteq, \mu) \in \pi^{-1}(t)$ and $\rho = (\iota, r)$ be a successful run of $\mathcal{A}$ on u. By letting $\iota'_\ell = (\iota_\ell, \upsilon^0_\ell)$ for $\ell \in \mathfrak{L}$ and $r'(x) = (r(x), \mu(x))$ for $x \in V$ we obtain a successful run $\rho' = (\iota', r')$ of $\mathcal{A}'$ on t. Thus, $g(\rho, u) = \rho'$ defines map

$$g\colon \bigcup_{u\in\pi^{-1}(t)} \operatorname{succ}(\mathcal{A}, u) \times \{ u \} \to \operatorname{succ}(\mathcal{A}', t).$$

Furthermore, g is inverse to f and hence f is a bijection. Finally, we conclude

$$\begin{aligned}\|\mathcal{A}'\|_{\mathrm{t}}(t) &= \sum_{\rho'\in\operatorname{succ}(\mathcal{A}',t)} \gamma'(\rho', t) = \sum_{\rho'\in\operatorname{succ}(\mathcal{A}',t)} \gamma(f(\rho')) = \sum_{\substack{u\in\pi^{-1}(t)\\ \rho\in\operatorname{succ}(\mathcal{A},u)}} \gamma(\rho, u) \\ &= \sum_{u\in\pi^{-1}(t)} \|\mathcal{A}\|_{\mathrm{t}}(u) = \sum_{u\in\pi^{-1}(t)} S(u) = \pi(S)(t).\end{aligned}$$

□

3.6 Lexicographic normal form

In order to prove the implication from (4) to (1) of Theorem 3.10 we have to consider a wFA $\mathcal{A}$ satisfying $\|\mathcal{A}\|_{\mathrm{w}}(u) = \|\mathcal{A}\|_{\mathrm{w}}(v)$ for all $u \sim_{\mathfrak{I}} v$ and to construct a wACA $\mathcal{B}$, such that $\|\mathcal{B}\|_{\mathrm{t}}(\mathrm{trc}(w)) = \|\mathcal{A}\|_{\mathrm{w}}(w)$ for any $w \in \Sigma^+$. The strategy will be as follows: for a given trace $t \in \mathbb{T}(\Sigma)$ the wACA $\mathcal{B}$ uniformly "chooses" some $w \in \mathrm{trc}^{-1}(t)$ and simulates all runs of $\mathcal{A}$ on w. In order to accomplish this uniform choice we use the lexicographic normal form.

First, we fix a total order $\preceq$ on $\mathfrak{L}$ for the rest of this chapter. This order induces the lexicographic order on $\mathfrak{L}^+$ which we denote with $\preceq$, too. Moreover, to each word $w = a_1 \dots a_n \in \Sigma^+$ we assign a sequence $\mathrm{lc}(w) \in \mathfrak{L}^+$ defined as

$$\mathrm{lc}(w) = \mathrm{lc}(a_1) \dots \mathrm{lc}(a_n).$$

A word $w \in \Sigma^+$ is in *lexicographic normal form* if for all $u \in \Sigma^+$ with $w \sim_{\mathfrak{I}} u$ we have $\mathrm{lc}(w) \preceq \mathrm{lc}(u)$. We let LNF be the set of all words in lexicographic normal form, i.e.,

$$\mathrm{LNF} = \{\, w \in \Sigma^+ \mid \forall u \in \Sigma^+ \colon w \sim_{\mathfrak{I}} u \to \mathrm{lc}(w) \preceq \mathrm{lc}(u) \,\}.$$

The following lemma shows that for each trace $t \in \mathbb{T}(\Sigma)$ there is exactly one $w \in \mathrm{trc}^{-1}(t)$ which is in lexicographic normal form. This word w will be denoted with $\mathrm{lnf}(t)$.

Lemma 3.16. *Let $u, v \in \Sigma^+$. If $u \sim_{\mathfrak{I}} v$ and $\mathrm{lc}(u) = \mathrm{lc}(v)$, then $u = v$.*

Proof. Let $\mathrm{trc}(u) = (V_1, \sqsubseteq_1, \lambda_1)$ and $\mathrm{trc}(w) = (V_2, \sqsubseteq_2, \lambda_2)$. By convention we have $V_1 = V_2 = \{\, 1, \dots, n \,\}$. Since $\mathrm{trc}(u) = \mathrm{trc}(v)$ there is a bijection $f \colon V_1 \to V_2$ such that $i \sqsubseteq_1 j$ iff $f(i) \sqsubseteq_2 f(j)$ for all all $i, j \in V_1$ and $\lambda_1(i) = \lambda_2(f(i))$, i.e., $a_i = b_{f(i)}$, for each $i \in V_1$. Thus, it suffices to show that f is the identity. Indirectly, let us assume there is an $i \in V_1$ such that $f(i) \neq i$ and consider the smallest such i w.r.t. $\leq$. Let $k = f(i)$. Moreover, there exists a $j \in V_1$ such that $f(j) = i$. Due to the minimality of i we have $j, k > i$. Furthermore, we know that $a_i = b_k$ and $a_j = b_i$. Since $\mathrm{lc}(a_i) = \mathrm{lc}(b_i)$ we obtain $(\mathrm{lc}(a_i), \mathrm{lc}(a_j)) \in \mathfrak{D}$ and $(\mathrm{lc}(b_i), \mathrm{lc}(b_k)) \in \mathfrak{D}$. This implies $i \sqsubseteq_1 j$ and $i \sqsubseteq_2 k$. From the first equation we can can conclude $k = f(i) \sqsubseteq_2 f(j) = i$ and thus, $i = k$. However, this contradicts the choice of i. □

3.7 From word automata to trace automata

This section is devoted to the proof of the missing implication from (4) to (1) of Theorem 3.10 which is restated as a proposition below.

Proposition 3.17. *Let $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series. If $\mathrm{trc}^{-1}(S)$ is a recognizable word series, then S is also recognizable.*

In order to accomplish this proof we need to introduce loop-connected finite automata. Let $\mathcal{A} = (Q, I, T, F)$ be a FA over Σ. A word $w = a_1 \dots a_n \in \Sigma^+$ is a *loop-label* of $\mathcal{A}$ if there are $q_0, \dots, q_n \in Q$ with $q_0 = q_n$ and $(q_{i-1}, a_i, q_i) \in T$ for all $i = 1, \dots, n$. Moreover, w is *connected* if the subgraph of $(\mathfrak{L}, \mathfrak{D})$ induced by $\{ \mathrm{lc}(a_1), \dots, \mathrm{lc}(a_n) \}$ is connected. Notice that the connectedness of w solely depends on $\mathrm{lc}(w)$. Finally, $\mathcal{A}$ is *loop-connected* if all loop-labels of $\mathcal{A}$ are connected. The following lemma establishes a relationship between lexicographic normal forms and loop-connected automata.

Lemma 3.18 (Kuske [31])**.** *The set* LNF $\subseteq \Sigma^+$ *can be recognized by some loop-connected FA.*

The subsequent two lemmas show that the class of word languages recognizable by loop-connected FAs is closed under inverse projection and intersection with recognizable languages. For the first lemma, we consider once a more another distributed alphabet (Γ, lc') and a location preserving map $\pi\colon \Gamma \to \Sigma$. By letting $\pi(\tau_1 \dots \tau_n) = \pi(\tau_1) \dots \pi(\tau_n)$ we extend π to a map $\pi\colon \Gamma^+ \to \Sigma^+$.

Lemma 3.19. *Let $L \subseteq \Sigma^+$ be a word language which is recognized by some loop-connected FA. Then there exists a loop-connected FA recognizing $\pi^{-1}(L)$.*

Proof. Let $\mathcal{A} = (Q, I, T, F)$ be a loop-connected FA recognizing L. From automata theory it is well known that the FA $\mathcal{A}' = (Q, I, T', F)$ with

$$T' = \{ (p, \tau, q) \mid (p, \pi(\tau), q) \in T \}$$

recognizes $\pi^{-1}(L)$. It remains to show that $\mathcal{A}'$ is loop-connected. Let $u \in \Gamma^+$ be a loop-label of $\mathcal{A}'$. Then $\pi(u)$ is a loop-label of $\mathcal{A}$ and hence connected. Since $\mathrm{lc}'(u) = \mathrm{lc}(\pi(u))$ this implies the connectedness of u. □

Lemma 3.20. *Let $L \subseteq \Sigma^+$ be a word language which is recognized by some loop-connected FA and $L' \subseteq \Sigma^+$ be a recognizable word language. Then there exists a loop-connected FA recognizing $L \cap L'$.*

Proof. Let $\mathcal{A} = (Q, I, T, F)$ be a loop-connected FA recognizing L and $\mathcal{A}' = (Q', I', T', F')$ be an FA recognizing L'. Again, from automata theory it is known that the FA $\mathcal{A}'' = (Q \times Q', I \times I', T'', F \times F')$ with

$$T'' = \left\{ \big((p, p'), a, (q, q')\big) \;\middle|\; (p, a, q) \in T, (p', a, q') \in T' \right\}$$

recognizes $L \cap L'$. Thus, it remains to show that $\mathcal{A}''$ is loop-connected. However, every loop-label of $\mathcal{A}''$ is also a loop-label of $\mathcal{A}$ and hence connected. □

As a last step before proving Proposition 3.17, we need to establish a connection between loop-connected FAs and $\mathfrak{I}$-diamond automata. For a language $L \subseteq \Sigma^+$ we denote with $\overline{L}$ the *trace closure* of L, i.e.,

$$\overline{L} = \mathrm{trc}^{-1}(\mathrm{trc}(L)).$$

Lemma 3.21 (Kuske [31])**.** *Let $L \subseteq \Sigma^+$ be a language which is recognized by some loop-connected FA. Then there exists an $\mathfrak{I}$-diamond automaton recognizing $\overline{L}$.*

Now, we are prepared to prove Proposition 3.17.

Proof of Proposition 3.17. Let $\mathcal{A} = (Q, I, T, F, \gamma)$ be a wFA with $\|\mathcal{A}\|_{\mathrm{w}} = \mathrm{trc}^{-1}(S)$. We put $\Gamma = Q \times \Sigma \times Q$ and let L be the set of words $(p_1, a_1, q_1) \dots (p_n, a_n, q_n) \in \Gamma^+$ satisfying the following three conditions:

(1) $a_1 \dots a_n \in \mathrm{LNF}$,

(2) $q_i = p_{i+1}$ for all $i = 1, \dots, n-1$, and

(3) $p_1 \in I$, $(p_i, a_i, q_i) \in T$ for each $i = 1, \dots, n$, and $q_n \in F$.

Obviously, such a word encodes a successful run of $\mathcal{A}$ on $a_1 \dots a_n$.

Now, we turn Γ into a distributed alphabet $(\Gamma, \mathrm{lc}_\Gamma)$ by letting $\mathrm{lc}_\Gamma(p, a, q) = \mathrm{lc}(a)$. Then $\pi(p, a, q) = a$ defines location preserving map $\pi\colon \Gamma \to \Sigma$. We will denote both extensions of π to maps $\Gamma^+ \to \Sigma^+$ and $\mathbb{T}(\Gamma) \to \mathbb{T}(\Sigma)$ with π as well, since it will be clear from the context which one is meant.

Furthermore, the FA $\mathcal{B} = (Q, I, T', F)$ over Γ with

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

recognizes the set of all words from Γ^+ satisfying conditions (2) and (3) from above. Since

$$L = \pi^{-1}(\mathrm{LNF}) \cap L_{\mathrm{w}}(\mathcal{B}),$$

the language L is recognized by some loop-connected FA due to Lemmas 3.18 to 3.20. In order to avoid ambiguities we denote the map $\mathrm{trc}\colon \Gamma^+ \to \mathbb{T}(\Gamma)$ in the following with trc_Γ. By Lemma 3.21 the set

$$\overline{L} = \mathrm{trc}_\Gamma^{-1}(\mathrm{trc}_\Gamma(L))$$

is recognizable by some $\mathfrak{I}$-diamond automaton and from Zielonka's theorem (Theorem 2.1) we conclude that $\mathrm{trc}_\Gamma(L) \subseteq \mathbb{T}(\Gamma)$ is a recognizable trace language.

Next, consider the wACA $\mathcal{B} = ((Z_\ell)_{\ell \in \mathfrak{L}}, \{\, z \,\}, (\Delta_\ell)_{\ell \in \mathfrak{L}}, \{\, z \,\}, (\delta_\ell)_{\ell \in \mathfrak{L}})$ over Γ and $\mathcal{D}$ where Z_ℓ is a singleton for each $\ell \in \mathfrak{L}$, $z \in \prod_{\ell \in \mathfrak{L}} Z_\ell$ is the unique global state,

$$\Delta_\ell = \prod_{m \in \mathfrak{D}(\ell)} Z_m \times (\Gamma_\ell \cap T) \times Z_\ell,$$

and

$$\delta_\ell((p_m)_{m\in\mathfrak{D}(\ell)}, \tau, q_\ell) = \gamma(\tau).$$

By Propositions 3.14 and 3.15 it suffices to show $S = \pi(\|\mathcal{B}\|\restriction_{\mathrm{trc}(L)})$ in order to prove that S is recognizable.

For the rest of this proof fix some $t \in \mathbb{T}(\Sigma)$ and let $w = \mathrm{lnf}(t)$ be the lexicographic normal form of t. We observe that

$$\pi(\|\mathcal{B}\|\restriction_{\mathrm{trc}_\Gamma(L)})(t) = \sum_{u\in\pi^{-1}(t)} \|\mathcal{B}\|\restriction_{\mathrm{trc}_\Gamma(L)}(u) = \sum_{u\in\pi^{-1}(t)\cap\mathrm{trc}_\Gamma(L)} \|\mathcal{B}\|(u)$$

and

$$S(t) = \mathrm{trc}^{-1}(S)(w) = \|\mathcal{A}\|(w) = \sum_{\sigma\in\mathrm{succ}(\mathcal{A},w)} \gamma(\sigma, w).$$

Thus, it suffices to give a bijection $f\colon \mathrm{succ}(\mathcal{A}, w) \to \pi^{-1}(t) \cap \mathrm{trc}_\Gamma(L)$ such that $\gamma(\sigma, w) = \|\mathcal{B}\|_{\mathrm{t}}(f(\sigma))$ for all $\sigma \in \mathrm{succ}(\mathcal{A}, w)$. Therefore let $a_1, \ldots, a_n \in \Sigma$ such that $w = a_1 \ldots a_n$. We show that the following definition has the desired properties

$$f\big((q_0, \ldots, q_n)\big) = \mathrm{trc}_\Gamma\big((q_0, a_1, q_1) \ldots (q_{n-1}, a_n, q_n)\big).$$

Let $\sigma = (q_0, \ldots, q_n) \in \mathrm{succ}(\mathcal{A}, w)$. Clearly, $\pi(f(\sigma)) = \mathrm{trc}(a_1 \ldots a_n) = t$. Since σ is a successful run of $\mathcal{A}$ on w and $w \in \mathrm{LNF}$ we have $(q_0, a_1, q_1) \ldots (q_{n-1}, a_n, q_n) \in L$ and hence, f is well-defined w.r.t. its image. Since $(q_{i-1}, a_i, q_i) \in T$ for all $i = 1, \ldots, n$, from the definition of $\mathcal{B}$ we conclude

$$\|\mathcal{B}\|(f(\sigma)) = \mathrm{Val}\big(\gamma(q_0, a_1, q_1), \ldots, \gamma(q_{n-1}, a_n, q_n)\big) = \gamma(\sigma, w).$$

The injectivity of f follows directly from Lemma 3.16 and the fact that $\mathrm{lc}_\Gamma(\tau)$ only depends on a for each $\tau = (p, a, q) \in \Gamma$.

Finally, we have to show that f is surjective. Therefore, consider some trace $u \in \pi^{-1}(t) \cap \mathrm{trc}_\Gamma(L)$. By definition there is some $\tau_1 \ldots \tau_n \in L$ such that $u = \mathrm{trc}_\Gamma(\tau_1 \ldots \tau_n)$. For $i = 1, \ldots, n$ let $\tau_i = (p_i, b_i, q_i)$ and put $v = b_1 \ldots b_n$. Then $t = \pi(u) = \mathrm{trc}(v)$ and since $v \in \mathrm{LNF}$ we obtain $v = w$. Putting $q_0 = p_1$ we obtain a successful run $\sigma = (q_0, \ldots, q_n)$ of $\mathcal{A}$ on w which satisfies $f(\sigma) = u$. □

4 Weighted MSO logics

Up to now, we have presented weighted asynchronous cellular automata as an abstract model for quantitative distributed systems. In this chapter we introduce weighted monadic second-order logics for traces as a formalism to write specifications for such systems. Since wACAs assign weights to traces, the interpretation of a weighted MSO-formula has to do the same. Afterwards, we compare the expressiveness of weighted MSO logics to that of wACAs. The highlight of this chapter – and the whole thesis in general – is a theorem stating sufficient conditions for the coincidence of both expressive powers.

In order to allow this comparison, valuation monoids have to be taken as the underlying weight structure. Since logical disjunction and existential quantification are means to express non-determinism, they are interpreted by addition. Accordingly, the zero from the valuation monoid is interpreted as the truth value "false". Moreover, the semantics of universal quantification can be defined using the valuation function. However, valuation monoids do neither provide a way to interpret logical conjunction nor do they offer a good candidate for the truth value "true". For this reason, Droste and Meinecke [15] extended their valuation monoids to product valuation monoids.

4.1 Product valuation monoids

A *product valuation monoid* is an algebraic structure $(\mathcal{D}, +, \mathrm{Val}, \diamond, \mathbf{0}, \mathbf{1})$ where

- $(\mathcal{D}, +, \mathrm{Val}, \mathbf{0})$ is a valuation monoid,
- $\mathbf{1} \in \mathcal{D}$ is a constant,
- $\diamond\colon \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ is a binary operation on $\mathcal{D}$ such that $\mathbf{0} \diamond d = d \diamond \mathbf{0} = \mathbf{0}$ and $\mathbf{1} \diamond d = d \diamond \mathbf{1} = d$ for all $d \in \mathcal{D}$, and
- $\mathrm{Val}(d_1, \ldots, d_n) = \mathbf{1}$ whenever $d_i = \mathbf{1}$ for all $i = 1, \ldots, n$.

The product valuation monoid is *order independent* if the underlying valuation monoid $(\mathcal{D}, +, \mathrm{Val}, \mathbf{0})$ has this property.

In order to fix the shortcomings mentioned above, valuation monoids were extended in two ways. On the one hand, the operation $\diamond$ was introduced for defining

the semantics of logical conjunction. On the other hand, the constant $\mathbf{1}$ shall serve as truth value "true". Besides, notice that the operation $\diamond$ needs to be neither associative nor commutative. However, restricted to $\{\mathbf{0}, \mathbf{1}\}$ is has both properties and models classical boolean conjunction.

Having in mind that the operations of a product valuation monoid will be used to interpret logical operations and the latter are supposed to interact in several ways, we define a couple of properties reflecting these connections. A product valuation monoid $(\mathcal{D}, +, \mathrm{Val}, \diamond, \mathbf{0}, \mathbf{1})$ is called *left-distributive* if $\diamond$ distributes over $+$ from the left and one of the following two properties is satisfied:

(1) *left-multiplicativity*, i.e., for all $n \geq 1$ and $d, d_1, \ldots, d_n \in \mathcal{D}$ we have

$$d \diamond \mathrm{Val}(d_1, \ldots, d_n) = \mathrm{Val}(d \diamond d_1, d_2, \ldots, d_n),$$

or

(2) *left*-Val-*distributivity*, i.e., for all $n \geq 1$ and $d, d_1, \ldots, d_n \in \mathcal{D}$ we have

$$d \diamond \mathrm{Val}(d_1, \ldots, d_n) = \mathrm{Val}(d \diamond d_1, \ldots, d \diamond d_n).$$

Moreover, $\mathcal{D}$ is *conditionally commutative* if

$$\mathrm{Val}(d_1, \ldots, d_n) \diamond \mathrm{Val}(d'_1, \ldots, d'_n) = \mathrm{Val}(d_1 \diamond d'_1, \ldots, d_n \diamond d'_n).$$

for all $n \geq 1$ and $d_1, \ldots, d_n, d'_1, \ldots, d'_n \in \mathcal{D}$ satisfying $d_i \diamond d'_j = d'_j \diamond d_i$ whenever $i > j$. If $\diamond$ is associative and distributes over $+$ from both sides, then $(\mathcal{D}, +, \diamond, \mathbf{0}, \mathbf{1})$ is a semiring and we call $(\mathcal{D}, +, \mathrm{Val}, \diamond, \mathbf{0}, \mathbf{1})$ a *valuation semiring*. A *cc-valuation semiring* is a valuation semiring which is left-distributive and conditionally commutative. Finally, $\mathcal{D}$ is *regular* if for every alphabet Σ and any $d \in \mathcal{D}$ the word series $\Sigma^+ \to \mathcal{D}, w \mapsto d$ is recognizable. Since it is not pleasant that regularity is defined using wFAs while the important automaton model in this thesis are wACAs, the following lemma corrects this aesthetic mistake.

Lemma 4.1. *Let $\mathcal{D}$ be a (product) valuation monoid and $d \in \mathcal{D}$. The following are equivalent:*

(1) For every alphabet Σ the trace series $\Sigma^+ \to \mathcal{D}, w \mapsto d$ is recognizable.

(2) There exists an alphabet Σ such that the word series $\Sigma^+ \to \mathcal{D}, w \mapsto d$ is recognizable.

If $\mathcal{D}$ is order independent both conditions are also equivalent to:

(3) For every distributed alphabet Σ the trace series $\mathbb{T}(\Sigma) \to \mathcal{D}, t \mapsto d$ is recognizable.

(4) There exists a distributed alphabet Σ such that the trace series $\mathbb{T}(\Sigma) \to \mathcal{D}$, $t \mapsto d$ is recognizable.

Proof. The implications from (1) to (2) and from (3) to (4) are trivial. Those from (1) to (3) and from (4) to (2) follow directly from Theorem 3.10. It remains to show that (2) implies (1). Thus, let Σ be some alphabet such that the word series $\Sigma^+ \to \mathcal{D}, w \mapsto d$ is recognizable and Γ be another alphabet. By assumption, there exists a wFA $\mathcal{A} = (Q, I, T, F, \gamma)$ over Σ such that $\|\mathcal{A}\|_{\mathrm{w}}(w) = d$ for all $w \in \Sigma^+$. We fix some $a_0 \in \Sigma$ and define a wFA $\mathcal{A}' = (Q, I, T', F, \gamma')$ over Γ where

$$T' = \{\, (p, \tau, q) \mid \tau \in \Gamma, (p, a_0, q) \in T \,\} \quad \text{and} \quad \gamma'(p, \tau, q) = \gamma(p, a_0, q).$$

We show that $\|\mathcal{A}'\|_{\mathrm{w}}(u) = d$ for all $u \in \Gamma^+$ and the claim follows.

Therefore, consider a word $u \in \Gamma^+$. Let $w = a_0 a_0 \dots a_0 \in \Sigma^+$ be the unique word consisting entirely of a_0's with $|w| = |u|$. From the definition of $\mathcal{A}'$ we conclude $\mathrm{succ}(\mathcal{A}', u) = \mathrm{succ}(\mathcal{A}, w)$ and $\gamma'(\sigma) = \gamma(\sigma)$ for all $\sigma \in \mathrm{succ}(\mathcal{A}', u)$. Hence,

$$\|\mathcal{A}'\|_{\mathrm{w}}(u) = \|\mathcal{A}\|_{\mathrm{w}}(w) = d. \qquad \square$$

Moreover, Droste and Meinecke mentioned as a sufficient condition for regularity that any left-distributive product valuation monoid is regular. However, they neither gave a proof nor an argument for that. Thus, we want to do this here.

Lemma 4.2. *Every left-distributive product valuation monoid is regular.*

Proof. Let $\mathcal{D}$ be a left-distributive product valuation monoid, Σ an alphabet, and $d \in \mathcal{D}$. First, let us assume that $\mathcal{D}$ is left-multiplicative. Consider the wFA $\mathcal{A} = (Q, I, T, F, \gamma)$ over Σ where $Q = \{\, p, q \,\}$, $I = \{\, p \,\}$, $F = \{\, q \,\}$, $T = Q \times \Sigma \times \{\, q \,\}$, $\gamma(p, a, q) = d$, and $\gamma(q, a, q) = \mathbf{1}$ for any $a \in \Sigma$. For every $w \in \Sigma^+$ there is exactly one successful run σ of $\mathcal{A}$ on w, namely $\sigma = (p, q, \dots, q)$, and we have

$$\|\mathcal{A}\|_{\mathrm{w}}(w) = \gamma(\sigma, w) = \mathrm{Val}(d, \mathbf{1}, \dots, \mathbf{1}) = d \diamond \mathrm{Val}(\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}) = d.$$

Now, suppose that $\mathcal{D}$ is left-Val-distributive. Let $\mathcal{B} = (Q, I, T, F, \gamma)$ be the wFA over Σ with $Q = I = F = \{\, q \,\}$, $T = Q \times \Sigma \times Q$, and $\gamma(q, a, q) = d$ for each $a \in \Sigma$. Again, there is exactly on successful run of $\mathcal{B}$ on each word $w \in \Sigma^+$, namely $\sigma = (q, \dots, q)$, and we obtain

$$\|\mathcal{B}\|_{\mathrm{w}}(w) = \gamma(\sigma, w) = \mathrm{Val}(d, \dots, d) = \mathrm{Val}(d \diamond \mathbf{1}, \dots, d \diamond \mathbf{1}) = d \diamond \mathrm{Val}(\mathbf{1}, \dots, \mathbf{1}) = d.$$

Thus, in both cases the word series $\Sigma^+ \to \mathcal{D}, w \mapsto d$ is recognizable. $\square$

We conclude this section by extending the valuation monoids from Example 3.1 to product valuation monoids.

Example 4.3 (Continues Example 3.1)**.** The following structures are product valuation monoids:

(1) $(\mathbb{Q} \cup \{-\infty\}, \max, \text{avg}, +, -\infty, 0)$ and $(\mathbb{Q} \cup \{\infty\}, \min, \text{avg}, +, \infty, 0)$ are both order independent cc-valuation semirings.

(2) $(K, +, \Pi, \cdot, \mathbf{0}, \mathbf{1})$ is a cc-valuation semiring for every semiring $(K, +, \cdot, \mathbf{0}, \mathbf{1})$. It is order independent iff K is commutative.

(3) $(\mathcal{L}, \vee, \inf, \wedge, \bot, \top)$ is a left-multiplicative and left-Val-distributive order independent product valuation monoid for every bounded lattice $(\mathcal{L}, \vee, \wedge, \bot, \top)$. From the proof of Lemma 4.1 we can see that it is also regular.

More examples for product valuation monoids can be found in [15].

4.2 Weighted MSO logics for Σ-posets

For the rest of this chapter we fix a distributed alphabet (Σ, lc) *and an order independent product valuation monoid* $(\mathcal{D}, +, \text{Val}, \diamond, \mathbf{0}, \mathbf{1})$. Moreover, we provide two disjoint countable sets $\mathcal{V}_1$ and $\mathcal{V}_2$ of first and second order variables, respectively. As a convention, elements of $\mathcal{V}_1$ (respectively $\mathcal{V}_2$) will be denoted with small (respectively capital) letters. The syntax of *weighted MSO logics* over Σ and $\mathcal{D}$ (*wMSO logics* for short) is given by the following BNF grammar [15]:

$$\beta ::= P_a(x) \mid x \leq y \mid x \in X \mid \neg\beta \mid \beta \wedge \beta \mid \forall x\, \beta \mid \forall X\, \beta$$
$$\varphi ::= d \mid \beta \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \exists x\, \varphi \mid \forall x\, \varphi \mid \exists X\, \varphi$$

where $d \in \mathcal{D}$, $a \in \Sigma$, $x, y \in \mathcal{V}_1$, and $X \in \mathcal{V}_2$. The formulas β are called *boolean formulas* whereas φ describes the *weighted MSO-formulas* (*wMSO-formulas* for short). For a given wMSO-formula φ the set $\text{free}(\varphi) \subseteq \mathcal{V}_1 \cup \mathcal{V}_2$ of free variables, i.e., variables not in the range of any quantifier, is defined as usual. We call φ *closed* whenever $\text{free}(\varphi) = \emptyset$.

Recall that we agreed to consider only finite, non-empty Σ*-posets throughout this thesis.* For a Σ-poset $s = (V, \sqsubseteq, \lambda)$ an *s-assignment* is a pair $\alpha = (\alpha_1, \alpha_2)$ consisting of two maps $\alpha_1 \colon \mathcal{V}_1 \to V$ and $\alpha_2 \colon \mathcal{V}_2 \to 2^V$. For $x \in \mathcal{V}_1$ and $v \in V$ the *update* $\alpha[x \mapsto v]$ is defined as $\alpha[x \mapsto v] = (\alpha_1[x \mapsto v], \alpha_2)$ where

$$\alpha_1[x \mapsto v](y) = \begin{cases} v & \text{if } y = x, \\ \alpha_1(y) & \text{otherwise.} \end{cases}$$

Similarly, for $X \in \mathcal{V}_2$ and $U \subseteq V$ the *update* $\alpha[X \mapsto U]$ is defined as $\alpha[X \mapsto U] = (\alpha_1, \alpha_2[X \mapsto U])$ where

$$\alpha_2[X \mapsto U](Y) = \begin{cases} U & \text{if } Y = X, \\ \alpha_2(Y) & \text{otherwise.} \end{cases}$$

Clearly, $\alpha[x \mapsto v]$ and $\alpha[X \mapsto U]$ are s-assignments as well.

The *semantics* $[\![\varphi]\!]$ of a wMSO-formula φ assigns an element of $\mathcal{D}$ to each pair (s, α) consisting of a Σ-poset $s = (V, \sqsubseteq, \lambda)$ and an s-assignment α. This semantics $[\![\varphi]\!]$ is defined inductively on the structure of φ as follows:

$$[\![P_a(x)]\!](s,\alpha) = \begin{cases} \mathbf{1} & \text{if } \lambda(\alpha_1(x)) = a \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad [\![x \leq y]\!](s,\alpha) = \begin{cases} \mathbf{1} & \text{if } \alpha_1(x) \sqsubseteq \alpha_1(y) \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$[\![x \in X]\!](s,\alpha) = \begin{cases} \mathbf{1} & \text{if } \alpha_1(x) \in \alpha_2(X) \\ \mathbf{0} & \text{otherwise} \end{cases} \qquad [\![\neg\beta]\!](s,\alpha) = \begin{cases} \mathbf{1} & \text{if } [\![\beta]\!](s,\alpha) = \mathbf{0} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

$$\begin{aligned} [\![d]\!](s,\alpha) &= d \\ [\![\varphi_1 \vee \varphi_2]\!](s,\alpha) &= [\![\varphi_1]\!](s,\alpha) + [\![\varphi_2]\!](s,\alpha) \\ [\![\varphi_1 \wedge \varphi_2]\!](s,\alpha) &= [\![\varphi_1]\!](s,\alpha) \diamond [\![\varphi_2]\!](s,\alpha) \\ [\![\exists x\, \varphi]\!](s,\alpha) &= \sum_{v \in V} [\![\varphi]\!](s, \alpha[x \mapsto v]) \\ [\![\exists X\, \varphi]\!](s,\alpha) &= \sum_{U \subseteq V} [\![\varphi]\!](s, \alpha[X \mapsto U]) \\ [\![\forall x\, \varphi]\!](s,\alpha) &= \mathrm{Val}\Big(\big([\![\varphi]\!](s, \alpha[x \mapsto v])\big)_{v \in V} \Big) \\ [\![\forall X\, \varphi]\!](s,\alpha) &= \mathrm{Val}\Big(\big([\![\varphi]\!](s, \alpha[X \mapsto U])\big)_{U \subseteq V} \Big) \end{aligned}$$

In the last two equations the values of the right-hand sides do not depend on the orders in which V and 2^V are enumerated since $\mathcal{D}$ is order independent. Hence, we leave these orders unspecified. Moreover, by induction we can easily show that $[\![\beta]\!](s, \alpha) \in \{ \mathbf{0}, \mathbf{1} \}$ for each boolean formula β.

Notice that the boolean formulas include neither a test for equality of first order variables nor disjunction nor existential quantification. However, all of them can be expressed using the abbreviations from Table 4.1, where $x, y \in \mathcal{V}_1$, $X \in \mathcal{V}_2$, and β, β_1, β_2 are boolean formulas.

abbr.	long form φ	semantics, i.e., $[\![\varphi]\!](s,\alpha) = \mathbf{1}$ iff
$x = y$	$x \leq y \wedge y \leq x$	$\alpha_1(x) = \alpha_1(y)$
$\beta_1 \veebar \beta_2$	$\neg(\neg\beta_1 \wedge \neg\beta_2)$	$[\![\beta_1]\!](s,\alpha) = \mathbf{1}$ or $[\![\beta_2]\!](s,\alpha) = \mathbf{1}$
$\dot\exists x\, \beta$	$\neg\forall x(\neg\beta)$	there is some $v \in V$ such that $[\![\beta]\!](s, \alpha[x \mapsto v]) = \mathbf{1}$
$\dot\exists X\, \beta$	$\neg\forall X(\neg\beta)$	there is some $U \subseteq V$ such that $[\![\beta]\!](s, \alpha[X \mapsto U]) = \mathbf{1}$

Table 4.1: Abbreviations for missing boolean formulas

Finally, from the definition of the semantics it is easy to see that $[\![\varphi]\!](s, \alpha)$ does not depend on the values of α for arguments outside $\mathrm{free}(\varphi)$. In particular, if φ is closed then $[\![\varphi]\!](s, \alpha)$ does not depend on α at all and we write $[\![\varphi]\!](s)$ instead.

4.3 Weighted MSO logics for traces and the main theorem

Since traces are Σ-posets satisfying some additional conditions, the logics introduced in the previous sections can directly be used for them. Though, to avoid ambiguities we want to introduce a more explicit notation.

Definition 4.4. Let φ be a closed wMSO-formula. The *trace semantics* of φ is the trace series$[\![\varphi]\!]_{\mathrm{t}} \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ defined by

$$[\![\varphi]\!]_{\mathrm{t}}(t) = [\![\varphi]\!](t).$$

As mentioned at the beginning of this chapter, we want to compare the expressive powers of wACAs and wMSO logics. However, already in some very special cases the full logics is expressively stronger than wACAs [15, 23]. Thus, we define several suitable fragments of wMSO logics and compare their expressiveness to the concept of recognizability.

A wMSO-formula φ is *almost boolean* if it belongs to the smallest class of wMSO-formulas containing all boolean formulas and all constants $d \in \mathcal{D}$ which is closed under conjunction and disjunction. Furthermore, φ is $\forall$*-restricted* if for all of its subformulas of the shape $\forall x\, \psi$ the formula ψ is almost boolean. For a wMSO-formula φ we denote the set of all constants $d \in \mathcal{D}$ appearing in φ with $\mathrm{const}(\varphi)$. Finally, we define three fragments of wMSO logics according to Table 4.2. A formula φ belongs to a fragment if it is $\forall$-restricted and satisfies the given condition for every sub-formula $\psi_1 \wedge \psi_2$ of φ. Clearly, each srMSO-formula is also an rMSO-formula and every formula from rMSO belongs to crMSO as well.

Definition 4.5. Let $X \in \{\, \mathrm{wMSO}, \mathrm{srMSO}, \mathrm{rMSO}, \mathrm{crMSO} \,\}$ be some fragment of wMSO. A trace series $S \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ is X*-definable* if there exists some closed X-formula φ such that

$$S = [\![\varphi]\!]_{\mathrm{t}}.$$

The following theorem is the main result of this thesis. Depending on the properties of the product valuation monoid, it gives a characterization of the class of recognizable trace series by certain fragments of weighted MSO logics.

Theorem 4.6. *Let $S \colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series.*

Fragment of wMSO	Condition on every sub-formula $\psi_1 \wedge \psi_2$
strongly restricted wMSO (srMSO)	• ψ_1 and ψ_2 are almost boolean, or • ψ_1 is boolean, or • ψ_2 is boolean
restricted wMSO (rMSO)	• ψ_1 is almost boolean, or • ψ_2 is boolean
commutatively restricted wMSO (crMSO)	• $\mathrm{const}(\psi_1)$ and $\mathrm{const}(\psi_2)$ commute w.r.t. $\diamond$, or • ψ_1 is almost boolean

Table 4.2: Fragments of wMSO

(1) Let $\mathcal{D}$ be regular. Then S is recognizable if and only if S is srMSO-definable.

(2) Let $\mathcal{D}$ be left-distributive. Then S is recognizable if and only if S is rMSO-definable.

(3) Let $\mathcal{D}$ be a cc-valuation semiring. Then S is recognizable if and only if S is crMSO-definable.

The rest of this chapter is devoted to the proof of this theorem. For this, we adopt a technique which was introduced by Ebinger and Muscholl [21] and extended to a weighted setting by Meinecke [35]. The main idea is to use a translation of formulas and to reduce to the corresponding characterization for word series [15].

4.4 Weighted MSO logics for words

In order to accomplish the mentioned reduction, we need to define word semantics for weighted MSO logics. Thus, we assign to each word $w = a_1 \dots a_n \in \Sigma^+$ a Σ-poset $\mathrm{lto}(w) = (V, \leq, \lambda)$ where $V = \{1, \dots, n\}$, $\leq$ is the natural order on V, and $\lambda(i) = a_i$ for all $i \in V$. Notice that $\mathrm{lto}(w)$ actually is a Σ-labeled total order, hence the name lto. *When introducing a Σ-poset as $\mathrm{lto}(w) = (V, \leq, \lambda)$ we assume that V and λ are constructed exactly in this way.* Clearly, each $\mathrm{lto}(w)$-assignment is also a $\mathrm{trc}(w)$-assignment and vice versa. For each closed wMSO-formula φ the *word semantics* of φ is the word series $[\![\varphi]\!]_{\mathrm{w}} \colon \Sigma^+ \to \mathcal{D}$ defined by

$$[\![\varphi]\!]_{\mathrm{w}}(w) = [\![\varphi]\!](\mathrm{lto}(w)).$$

It is easy to see that this definition of the word semantics coincides with the one from [15]. For $X \in \{\mathrm{wMSO}, \mathrm{srMSO}, \mathrm{rMSO}, \mathrm{crMSO}\}$ the notion X-definability of word series is analogue to Definition 4.5. The connection between recognizability

and definability of word series was already investigated in a more general setting where order independence was not assumed.

Theorem 4.7 (Droste and Meinecke [15]). *Let $S\colon \Sigma^+ \to \mathcal{D}$ be a word series.*

(1) Let $\mathcal{D}$ be regular. Then S is recognizable if and only if S is srMSO-definable.

(2) Let $\mathcal{D}$ be left-distributive. Then S is recognizable if and only if S is rMSO-definable.

(3) Let $\mathcal{D}$ be a cc-valuation semiring. Then S is recognizable if and only if S is crMSO-definable.

Obviously, Theorem 4.6 follows from Theorems 3.10 and 4.7 in combination with the following result.

Theorem 4.8. *Let $X \in \{ wMSO, srMSO, rMSO, crMSO \}$ be a fragment of wMSO and $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series. Then S is X-definable if and only if $\mathrm{trc}^{-1}(S)$ is X-definable.*

Therefore, the rest of this chapter is dedicated to proving this theorem. As already mentioned, we will do this by using the translation technique introduced in [21].

4.5 From trace logics to word logics

For the rest of this chapter we fix a fragment $X \in \{ wMSO, srMSO, rMSO, crMSO \}$ of wMSO. The goal of this section is to show the "only if"-part of Theorem 4.8 which is restated by the following proposition.

Proposition 4.9. *Let $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be an X-definable trace series. Then $\mathrm{trc}^{-1}(S)$ is an X-definable word series.*

The main idea is to take a closed X-formula φ with $S = [\![\varphi]\!]_{\mathrm{t}}$ and to transform it into a closed X-formula φ^{w} satisfying $\mathrm{trc}^{-1}(S) = [\![\varphi^{\mathrm{w}}]\!]_{\mathrm{w}}$. The first step is to establish a connection between the orders of $\mathrm{trc}(w)$ and $\mathrm{lto}(w)$ for any $w \in \Sigma^+$. Therefore, we denote the number of different locations with N, i.e., $N = |\mathfrak{L}|$.

Lemma 4.10. *Let $w \in \Sigma^+$ and $t = \mathrm{trc}(w) = (V, \sqsubseteq, \lambda)$. Then for all $i, j \in V$ we have $i \sqsubseteq j$ if and only if there are $k_0, \ldots, k_N \in V$ such that $k_0 = i$, $k_N = j$, and $k_{r-1} \leq k_r$ and $\big(\mathrm{lc}_t(k_{r-1}), \mathrm{lc}_t(k_r)\big) \in \mathfrak{D}$ for all $r = 1, \ldots, N$.*

Proof. First, recall that $\sqsubseteq$ was defined to be the transitive and reflexive closure of the set

$$E = \{\, (i,j) \in V \times V \mid i < j \text{ and } (\mathrm{lc}_t(i), \mathrm{lc}_t(j)) \in \mathfrak{D} \,\}.$$

Thus, the "if"-part follows immediately.

For the converse implication consider $i \sqsubseteq j$. We call a sequence $(k_0, \ldots, k_n) \in V^+$ *good* if $k_0 = i$, $k_n = j$, and $(k_{r-1}, k_r) \in E$ for all $r = 1, \ldots, n$. By definition of $\sqsubseteq$ there is at least one good sequence $(k_0, \ldots, k_n)$. Next, we show that there is also a good sequence with $n \leq N$. Therefore, assume $n > N$. According to the pigeonhole principle there are $r, s \in \{\, 1, \ldots, n \,\}$ with $r < s$ such that $\mathrm{lc}_t(k_r) = \mathrm{lc}_t(k_s)$. Since $(k_r, k_s) \in E^+$ we also have $k_r < k_s$. Moreover, from $(\mathrm{lc}_t(k_{r-1}), \mathrm{lc}_t(k_r)) \in \mathfrak{D}$ and $\mathrm{lc}_t(k_r) = \mathrm{lc}_t(k_s)$ we conclude $(\mathrm{lc}_t(k_{r-1}), \mathrm{lc}_t(k_s)) \in \mathfrak{D}$. Thus, $(k_{r-1}, k_s) \in E$ and $(k_0, \ldots, k_{r-1}, k_s, \ldots, k_n)$ is a good sequence which is shorter than the one we started with. By repeatedly applying this argument we get a good sequence $(k_0, \ldots, k_n)$ with $n \leq N$. Finally, we put $k_{n+1} = \cdots = k_N = j$ and obtain the desired $k_0, \ldots, k_N \in V$. □

This allows us to give the desired translation. For some arbitrary X-formula φ we denote with φ^{w} the formula which is obtained from φ by replacing every sub-formula of the shape $x \leq y$ with

$$\exists z_0\, \exists z_1 \cdots \exists z_N \Bigg(z_0 = x \wedge z_N = y \wedge \bigwedge_{1 \leq i \leq N} \big(z_{i-1} \leq z_i \wedge (\mathrm{lc}(z_{i-1}), \mathrm{lc}(z_i)) \in \mathfrak{D} \big) \Bigg) \tag{4.1}$$

where $(\mathrm{lc}(z), \mathrm{lc}(z')) \in \mathfrak{D}$ is short hand for

$$\bigvee_{\substack{a,b \in \Sigma \\ (\mathrm{lc}(a), \mathrm{lc}(b)) \in \mathfrak{D}}} P_a(z) \wedge P_b(z').$$

Notice that the formula in Eq. (4.1) is boolean and hence φ^{w} is an X-formula. Moreover, since formula (4.1) has exactly the same free variables as $x \leq y$, namely x and y, the formula φ^{w} is closed iff φ is closed. The following lemma and its corollary state the desired property of this translation.

Lemma 4.11. *Let φ be an X-formula, $w \in \Sigma^+$ and α be an $\mathrm{lto}(w)$-assignment. Then*

$$[\![\varphi^{\mathrm{w}}]\!](\mathrm{lto}(w), \alpha) = [\![\varphi]\!](\mathrm{trc}(w), \alpha).$$

Corollary 4.12. *Let φ be a closed X-formula. Then for each $w \in \Sigma^+$ we have*

$$[\![\varphi^{\mathrm{w}}]\!]_{\mathrm{w}}(w) = [\![\varphi]\!]_{\mathrm{t}}(\mathrm{trc}(w)).$$

Proof of Lemma 4.11. The claim is proved by induction on the structure of φ and the only non-trivial case is $\varphi = (x \leq y)$. However, this situation was subject of Lemma 4.10 and formula (4.1) reflects exactly the results there. □

Now, we are prepared to give the missing proof of Proposition 4.9.

Proof of Proposition 4.9. Let φ be a closed X-formula such that $[\![\varphi]\!]_{\mathrm{t}} = S$. Then φ^{w} is also a closed X-formula. Moreover, for any $w \in \Sigma^+$ we obtain

$$[\![\varphi^{\mathrm{w}}]\!]_{\mathrm{w}}(w) = [\![\varphi]\!]_{\mathrm{t}}(\mathrm{trc}(t)) = S(\mathrm{trc}(w)) = \mathrm{trc}^{-1}(S)(w),$$

i.e., $\mathrm{trc}^{-1}(S) = [\![\varphi^{\mathrm{w}}]\!]_{\mathrm{w}}$. Thus, $\mathrm{trc}^{-1}(S)$ is X-definable. □

4.6 From word logics to trace logics

This section is devoted to the missing proof of the "if"-part of Theorem 4.8:

Proposition 4.13. *Let $S\colon \mathbb{T}(\Sigma) \to \mathcal{D}$ be a trace series. If $\mathrm{trc}^{-1}(S)$ is X-definable, then S is also X-definable.*

The key idea combines that from the previous section with the one described at the beginning of Section 3.6. There, we introduced the lexicographic normal form for the purpose of uniformly "choosing" an element from $\mathrm{trc}^{-1}(t)$ for each $t \in \mathbb{T}(\Sigma)$. Here, we fix again a total order $\preceq$ on Σ and reuse all other notations from Section 3.6. For any $w \in \mathrm{LNF}$ we will investigate a relationship between the orders of $\mathrm{lto}(w)$ and $\mathrm{trc}(w)$. Then, we exploit this relation for translating an X-formula φ for $\mathrm{trc}^{-1}(S)$ into an X-formula φ^{t} for S. Again, we let $N = |\mathfrak{L}|$.

Let $t = (V, \sqsubseteq, \lambda) \in \mathbb{T}(\Sigma)$ be some trace. We inductively construct a sequence $R_t^{(1)}, \ldots, R_t^{(2N)} \subseteq V \times V$ of relations on V. First, we put

$$R_t^{(1)} = \{\, (i,j) \mid i \sqsubseteq j \,\}.$$

Thereafter, for $2 \leq n \leq 2N$ we let $R_t^{(n)}$ be the set of all pairs $(i,j) \in V \times V$ which satisfy one of the following three conditions:

(1) $i \sqsubseteq j$, or

(2) $\mathrm{lc}_t(i) \prec \mathrm{lc}_t(j)$ and $(j,i) \notin R_t^{(n-1)}$, or

(3) $\mathrm{lc}_t(i) \succ \mathrm{lc}_t(j)$ and there exists a $k \in V$ such that $\mathrm{lc}_t(i) \prec \mathrm{lc}_t(k)$, $(i,k) \in R_t^{(n-1)}$, and $k \sqsubseteq j$.

The following lemma establishes a relationship between $R_t^{(2N)}$ and the lexicographic normal form of t.

Lemma 4.14. *Let $w \in \mathrm{LNF}$, $t = \mathrm{trc}(w) = (V, \sqsubseteq, \lambda)$, and $i, j \in V$. Then we have $(i,j) \in R_t^{(2N)}$ if and only if $i \leq j$.*

Proof. First, let $\ell_1 \succ \cdots \succ \ell_N$ be the descending enumeration of $\mathfrak{L}$ w.r.t $\preceq$. For $i \in V$ we let $\mathrm{ord}(i)$ denote the unique $n \in \{1, \ldots, N\}$ with $\mathrm{lc}_t(i) = \ell_n$. Moreover, we define a map $f \colon V \times V \to \{1, \ldots, 2N\}$ by

$$f(i,j) = \begin{cases} \mathrm{ord}(i) + \mathrm{ord}(j) & \text{if } \mathrm{lc}_t(i) \prec \mathrm{lc}_t(j), \\ \mathrm{ord}(i) + \mathrm{ord}(j) - 1 & \text{otherwise.} \end{cases}$$

Using induction on $n \in \{1, \ldots, 2N\}$ we show that for all $i, j \in V$ with $f(i,j) \leq n$ we have $(i,j) \in R_t^{(n)}$ iff $i \leq j$. For $n = 2N$ this implies the claim.

First, we consider $n = 1$ and $i, j \in V$ with $f(i,j) \leq 1$. The "only if"-part obviously follows from $i \sqsubseteq j$. Conversely, from $f(i,j) \leq 1$ we conclude $\mathrm{lc}_t(i) = \mathrm{lc}_t(j) = \ell_1$ and hence $(\mathrm{lc}_t(i), \mathrm{lc}_t(j)) \in \mathfrak{D}$. Together with $i \leq j$ this implies $i \sqsubseteq j$.

Now, assume $n > 1$ and consider $i, j \in V$ with $f(i,j) \leq n$. For the "only if"-part presume $(i,j) \in R_t^{(n)}$. Then one of the three requirements from the definition of $R_t^{(n)}$ is met. Condition (1) obviously implies $i \leq j$. If (2) holds, we have $f(j,i) = f(i,j) - 1 < n$ and by induction obtain $j \not\leq i$, i.e., $i < j$. Finally, assume (3) is satisfied. From $\mathrm{lc}_t(k) \succ \mathrm{lc}_t(i) \succ \mathrm{lc}_t(j)$ we conclude $\mathrm{ord}(k) < \mathrm{ord}(j) - 1$ and hence $f(i,k) < f(i,j) \leq n$. By induction we get $i \leq k$. Together with $k \leq j$ we obtain $i \leq j$.

In order to show the "if"-part, assume that $i \leq j$ holds true but conditions (1) and (2) do not. We have to show that (3) is met. First, we indirectly show $\mathrm{lc}_t(i) \succ \mathrm{lc}_t(j)$. Thus, conversely suppose that $\mathrm{lc}_t(i) \preceq \mathrm{lc}_t(j)$. From $i \leq j$ and the violation of (1) we conclude $i \not\sqsupseteq j$ and hence $(\mathrm{lc}_t(i), \mathrm{lc}_t(j)) \notin \mathfrak{D}$. In particular, $\mathrm{lc}_t(i) \neq \mathrm{lc}_t(j)$ and therefore $\mathrm{lc}_t(i) \prec \mathrm{lc}_t(j)$. However, since (2) does not hold we have $(j,i) \in R_t^{(n-1)}$. Moreover, from $f(j,i) = f(i,j) - 1 < n$ we get by induction $j \leq i$. Together with $i \leq j$ this implies $i = j$ and hence condition (1) is satisfied. Since this is a contradiction we have shown $\mathrm{lc}_t(i) \succ \mathrm{lc}_t(j)$.

Now, consider the smallest $k \in V$ w.r.t. $\leq$ such that $i \leq k$ and $k \sqsubseteq j$. Such a k exists since $k = j$ satisfies both conditions. Due to the violation of (1) we have $i < k$. For all $k' \in \{i, \ldots, k-1\}$ we have $(k', k) \notin E$ since the opposite would imply $k \sqsubseteq j$, contradicting the minimality of k. Therefore, from $k' < k$ we conclude $(\mathrm{lc}_t(k'), \mathrm{lc}_t(k)) \notin \mathfrak{D}$. Thus, for

$$u = a_1 \ldots a_{i-1} a_k a_i \ldots a_{k-1} a_{k+1} \ldots a_n$$

we have $w \sim_{\mathfrak{I}} u$. Since $w \in \mathrm{LNF}$ we get $\mathrm{lc}(w) \preceq \mathrm{lc}(u)$ and hence $\mathrm{lc}_t(i) \preceq \mathrm{lc}_t(k)$. Furthermore, $(\mathrm{lc}_t(i), \mathrm{lc}_t(k)) \notin \mathfrak{D}$ implies $\mathrm{lc}_t(i) \prec \mathrm{lc}_t(k)$. Again, from $\mathrm{lc}_t(k) \succ \mathrm{lc}_t(i) \succ \mathrm{lc}_t(j)$ we obtain $f(i,k) < f(i,j) \leq n$ and by induction get $(i,k) \in R_t^{(n-1)}$. □

This lemma allows us to give the desired translation of formulas. We let

$$\mathrm{lex}_1(x,y) = (x \leq y)$$

and for $n > 1$ we inductively define

$$\mathrm{lex}_n(x,y) = \Big(x \leq y \veebar \big(\mathrm{lc}(x) \prec \mathrm{lc}(y) \wedge \neg\, \mathrm{lex}_{n-1}(y,x)\big) \veebar \Big(\mathrm{lc}(y) \prec \mathrm{lc}(x) \wedge \exists z \big(\mathrm{lc}(x) \prec \mathrm{lc}(z) \wedge \mathrm{lex}_{n-1}(x,z) \wedge z \leq y\big)\Big)\Big),$$

where $\mathrm{lc}(x') \prec \mathrm{lc}(y')$ is an abbreviation for

$$\bigvee_{\substack{a,b \in \Sigma \\ \mathrm{lc}(a) \prec \mathrm{lc}(b)}} P_a(x') \wedge P_b(y').$$

Finally, put $\mathrm{lex}(x,y) = \mathrm{lex}_{2N}(x,y)$. For some X-formula φ we denote with φ^{t} the formula which is obtained from φ by replacing every sub-formula of the shape $x \leq y$ with $\mathrm{lex}(x,y)$. Since $\mathrm{lex}(x,y)$ is a boolean formula, φ^{t} also belongs to the X-fragment of wMSO. Moreover, $\mathrm{lex}(x,y)$ has exactly the same free variables as $x \leq y$ and hence φ^{t} is closed iff φ is closed. The lemma below and its corollary establish a semantic relation between φ and φ^{t}.

Lemma 4.15. *Let φ be a X-formula, $w \in \mathrm{LNF}$, and α be a $\mathrm{trc}(w)$-assignment. Then*

$$[\![\varphi^{\mathrm{t}}]\!](\mathrm{trc}(w), \alpha) = [\![\varphi]\!](\mathrm{lto}(w), \alpha).$$

Corollary 4.16. *Let φ a closed X-formula. Then for all $t \in \mathbb{T}(\Sigma)$ we have*

$$[\![\varphi^{\mathrm{t}}]\!]_{\mathrm{t}}(t) = [\![\varphi]\!]_{\mathrm{w}}(\mathrm{lnf}(t)).$$

Proof of Lemma 4.15. Again, the claim is shown by induction on the structure of φ and the only non-trivial case is $\varphi = (x \leq y)$. However, this situation was considered in Lemma 4.14 and the formula $\mathrm{lex}(x,y)$ is translation of the results there. □

Now, we are in a position to give the last of proof this chapter.

Proof of Proposition 4.13. Let φ be a closed X-formula such that $[\![\varphi]\!]_{\mathrm{w}} = \mathrm{trc}^{-1}(S)$. Then φ^{t} is also a closed X-formula. Moreover, for any $t \in \mathbb{T}(\Sigma)$ we have

$$[\![\varphi^{\mathrm{t}}]\!]_{\mathrm{t}}(t) = [\![\varphi]\!]_{\mathrm{w}}(\mathrm{lnf}(t)) = \mathrm{trc}^{-1}(S)(\mathrm{lnf}(t)) = S(t),$$

i.e., $S = [\![\varphi^{\mathrm{t}}]\!]_{\mathrm{t}}$. Thus, S is X-definable. □

5 Quantitative model-checking of distributed systems

So far, we defined and studied weighted asynchronous automata and weighted MSO logics for traces. The former can be used to model quantitative distributed systems whereas the latter are a formalism to write specifications for such systems. Naturally, the question arises how we can check whether the abstract model "matches" a given specification. This problem is commonly known as *quantitative model-checking*. In this chapter we introduce two quantitative model-checking problems, one for distributed systems and another for sequential systems. Afterwards we demonstrate how the former can be reduced to the latter.

Before we can delve into the details we have to agree on the meaning of the word "matches". Usually, this depends on the context. For example, if the automaton models the consumption of some resource whereas the specification describes upper bounds for the usage of this resource, then "matches" means "is at most". However, if the quantities represent some gain, we probably want the opposite. Finally, in the setting of multi-valued logics we are also interested in equality. Due to these various possibilities we do not specify the exact meaning here but consider an arbitrary relation on the weight structure.

For the rest of this chapter we fix a product valuation monoid $\mathcal{D}$ and a fragment $X \in \{ wMSO, srMSO, rMSO, crMSO \}$ of wMSO. Moreover, we fix some relation $R \subseteq \mathcal{D} \times \mathcal{D}$ on $\mathcal{D}$ which shall be interpreted as the meaning of "matches". The mentioned quantitative model-checking problems are formally defined as follows.

Definition 5.1. Let $\mathcal{D}$ be order independent. The *(quantitative) distributed model-checking problem* for $\mathcal{D}$, R, and X (*DMCP* for short) is the following decision problem:

Input: $(\mathfrak{L}, \mathfrak{D})$ – an architecture graph,
(Σ, lc) – a distributed alphabet over $(\mathfrak{L}, \mathfrak{D})$,
$\mathcal{A}$ – a wACA over Σ and $\mathcal{D}$, and
φ – a closed X-formula over Σ and $\mathcal{D}$.

Question: Does

$$(\|\mathcal{A}\|_{\mathrm{t}}(t), [\![\varphi]\!]_{\mathrm{t}}(t)) \in R$$

hold true for all $t \in \mathbb{T}(\Sigma)$?

Definition 5.2. The *(quantitative) sequential model-checking problem* for $\mathcal{D}$, R, and X (*SMCP* for short) is the following decision problem:

Input: Σ – an alphabet,
$\mathcal{A}$ – a wFA over Σ and $\mathcal{D}$, and
φ – a closed X-formula over Σ and $\mathcal{D}$.

Question: Does

$$(\|\mathcal{A}\|_{\mathrm{w}}(w), [\![\varphi]\!]_{\mathrm{w}}(w)) \in R$$

hold true for all $w \in \Sigma^+$?

Next, we establish a connection between the decidability of both problems.

Theorem 5.3. *Let $\mathcal{D}$ be order independent. Then the DMCP is decidable if and only if the SMCP is decidable.*

Proof. First, we show the "if"-part. Thus, consider an instance of the DMCP consisting of an architecture graph $(\mathfrak{L}, \mathfrak{D})$, a distributed alphabet (Σ, lc) over $(\mathfrak{L}, \mathfrak{D})$, a wACA $\mathcal{A}$ over Σ, and a closed X-formula φ over Σ. Then $\mathcal{A}^{\mathrm{w}}$ is a wFA over Σ and φ^{w} is a closed X-formula over Σ. From Lemma 3.13 and Corollary 4.12 we can conclude that

$$(\|\mathcal{A}\|_{\mathrm{t}}(t), [\![\varphi]\!]_{\mathrm{t}}(t)) \in R$$

for all $t \in \mathbb{T}(\Sigma)$ if and only if

$$(\|\mathcal{A}^{\mathrm{w}}\|_{\mathrm{w}}(w), [\![\varphi^{\mathrm{w}}]\!]_{\mathrm{w}}(w)) \in R.$$

The last question is an instance of the SMCP and hence decidable by assumption. Since the constructions of $\mathcal{A}^{\mathrm{w}}$ and φ^{w} from $\mathcal{A}$ and φ are effective, we can decide the given instance of the DMCP as well.

For the "only if"-part, consider the case $|\mathfrak{L}| = 1$. Then the notions of traces and words, wACAs and wFAs, and trace and word semantics of X-formulas coincide. Hence, the SMCP is a special case of the DMCP. □

Remark 5.4. Since computational complexity matters in the field of model-checking as well, we will take this issue into account for a moment. Obviously, the reduction does not change the size of the alphabet. If the input wACA $\mathcal{A}$ has at most n local states for each location, then the wFA $\mathcal{A}^{\mathrm{w}}$ has at most $n^{|\mathfrak{L}|}$ states. Let the size of an X-formula be measured by the size of its syntax tree. Then the size of the formula in Eq. (4.1) belongs to $O(|\mathfrak{L}| \cdot |\Sigma|^2)$. Finally, if the input formula φ has size m then the size of φ^{w} is in $O(m \cdot |\mathfrak{L}| \cdot |\Sigma|^2)$.

We want to conclude this section with a short discussion about the decidability of the SMCP. Thus, let us assume that the product valuation monoid $\mathcal{D}$ and the fragment X of wMSO are compatible in the sense of Theorems 4.6 and 4.7. Since

all proofs in [15] are constructive, for each X-formula φ we can effectively obtain a wFA such that

$$\|\mathcal{A}\|_{\mathrm{w}} = [\![\varphi]\!]_{\mathrm{w}}.$$

Hence, we can reduce the SMCP to the following *wFA relationship problem*:

Input: Σ – an alphabet,
$\mathcal{A}$ and $\mathcal{B}$ – two wFAs over Σ and $\mathcal{D}$.
Question: Does

$$(\|\mathcal{A}\|_{\mathrm{w}}(w), \|\mathcal{B}\|_{\mathrm{w}}(w)) \in R$$

hold true for all $w \in \Sigma^+$?

It is well known, that the decidability of this problem depends on the choice of $\mathcal{D}$ and R. Recall that we explained how to regard semirings and bounded lattices as product valuation monoids in Example 4.3. If we construct $\mathcal{D}$ from the Boolean semiring the problem is decidable for any choice of R. For computable fields it is at least decidable when R is the equality relation [2, 12]. However, in case of the tropical semiring the wFA relationship problem is undecidable for $=$ and $\leq$, respectively [28]. Furthermore, this implies undecidability for the weight structures from Example 4.3 (1) and the relations $=$ and $\leq$. Finally, from [19, 20] we can conclude decidability for arbitrary bounded lattices and locally finite semirings as soon as R is decidable.

6 Conclusions

In this thesis we introduced weighted asynchronous cellular automata as an abstract model for quantitative distributed systems. In order to get well defined semantics we had to assume that the underlying valuation monoid is order independent. Afterwards, we established a connection between wACAs and weighted $\mathfrak{I}$-diamond automata, providing a weighted version of Zielonka's theorem [41] and generalizing a result of Kuske [31]. Furthermore, we presented weighted MSO logics for traces as a formalism to write specifications for quantitative distributed systems. We showed a close relationship of this logics to weighted MSO logics for words. Combining these two results with a recent one of Droste and Meinecke [15], we obtained our main result. It gives a characterization of recognizable trace series by means of weighted MSO logics. Depending on the properties of the weight structure we had to choose different fragments of the logics. This characterization is an extension of the results of Droste and Gastin [13], Kuske and Meinecke [23, 31, 35], and Droste and Meinecke [15].

Since all proofs in the thesis were effective, we opened the way for quantitative model-checking of distributed systems. This issue was discussed in the last chapter where we reduced the model-checking problem for distributed systems to that for sequential systems. Finally, we showed how the latter can be further reduced to the wFA relationship problem and sketched the state of knowledge concerning the decidability of this problem.

Model-checking is a field where complexity plays an important role beyond the coarse separation into decidable and undecidable problems. In particular, the computational complexity of decidable problems is of great interest. However, we put no focus on this issue because it has not been studied yet how hard it is to reduce the SMCP to the wFA relationship problem. Moreover, the complexity of the latter problem has been analyzed only in some special cases [7, 30]. Hence, both problems need further investigation in future work.

Pointing in another direction, it would be interesting to consider other kinds of logics in our quantitative framework. In particular, weighted temporal logics attracted some attention [4, 30]. Moreover, Kleene's theorem [26] inspired characterizations of the expressiveness of weighted finite automata by means of weighted rational expressions for several weight structures [16, 20, 39]. Analogously, Droste and Gastin [10, 32] generalized Ochmański's theorem [36, 37] by characterizing the expressive power of semiring weighted $\mathfrak{I}$-diamond automata by two classes of weighted

rational expressions. Recently, we have shown that a similar result holds true when we replace distributivity by some local finiteness condition [25]. Naturally, the question arises whether we can also obtain such a characterization if the weights are taken from a valuation monoid.

Acknowledgements

I want to thank Prof. Manfred Droste for proposing the topic and giving helpful suggestions which improved the thesis. I am also grateful to Dietrich Kuske who unawarely encouraged me to study asynchronous cellular automata by one tiny but teasing comment. Finally, I want to thank Ingmar Meinecke for patiently answering myriads of my questions and providing lots of valuable hints during our stay in Novi Sad.

7 Bibliography

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[2] J. Berstel and C. Reutenauer. *Rational Series and their Languages*, volume 12 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1988.

[3] G. Birkhoff and J. von Neumann. The logic of quantum mechanics. *Annals of Mathematics*, 37:823–843, 1936.

[4] B. Bollig and P. Gastin. Weighted versus probabilistic logics. In *Proceedings of DLT 2009*, volume 5583 of *LNCS*, pages 18–38, 2009.

[5] J. R. Büchi. Weak second-order arithmetic and finite automata. *Zeitschr. f. math. Logik und Grundlagen d. Math.*, 6:66–92, 1960.

[6] R. G. Bukharaev. Probabilistic automata. *Journal of Mathematical Sciences*, 13:359–386, 1980.

[7] K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *Proceedings of CSL 2008*, volume 5213 of *LNCS*, pages 385–400, 2008.

[8] V. Diekert and Y. Métivier. Partial commutation and traces. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 457–533. Springer-Verlag, 1997.

[9] V. Diekert and G. Rozenberg, editors. *The Book of Traces*. World Scienific Publishing, 1995.

[10] M. Droste and P. Gastin. The Kleene-Schützenberger theorem for formal power series in partially commuting variables. *Information and Computation*, 153(1):47–80, 1999.

[11] M. Droste and P. Gastin. Weighted automata and weighted logics. In *Proceedings of ICALP 2005*, volume 3580 of *LNCS*, pages 513–525, 2005.

[12] M. Droste and P. Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2):69–86, 2007.

[13] M. Droste and P. Gastin. Weighted automata and weighted logics. In Droste et al. [14], chapter 5, pages 175–211.

[14] M. Droste, W. Kuich, and H. Vogler, editors. *Handbook of Weighted Automata.* Springer-Verlag, 2009.

[15] M. Droste and I. Meinecke. Describing average- and longtime-behavior by weighted MSO logics. In *Proceedings of MFCS 2010*, volume 6281 of *LNCS*, pages 537–548, 2010.

[16] M. Droste and I. Meinecke. Regular expressions on average and in the long run. In *Proceedings of CIAA 2010*, LNCS, 2010, to appear.

[17] M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting. *Theoretical Computer Science*, 410:3481–3494, 2009.

[18] M. Droste and G. Rahonis. Weighted automata and weighted logics on infine words. *Izvestiya VUZ. Matematika*, 54:26–45, 2010.

[19] M. Droste, T. Stüber, and H. Vogler. Weighted finite automata over strong bimonoids. *Information Sciences*, 180:156–166, 2010.

[20] M. Droste and H. Vogler. Kleene and Büchi theorems for weighted automata and multi-valued logics over arbitrary bounded lattices. In *Proceedings of DLT 2010*, volume 6224 of *LNCS*, pages 162–172, 2010.

[21] W. Ebinger and A. Muscholl. Logical definability on infinite traces. *Theoretical Computer Science*, 154:67–84, 1996.

[22] C. C. Elgot. Decision problems of finite automata design and related arithmetics. *Trans. Amer. Math. Soc.*, 98:21–52, 1961.

[23] I. Fichtner, D. Kuske, and I. Meinecke. Traces, series-parallel posets, and pictures: A weighted study. In Droste et al. [14], chapter 10, pages 397–441.

[24] H. J. Hoogenboom and G. Rozenberg. Dependence graphs. In Diekert and Rozenberg [9], chapter 2, pages 43–67.

[25] M. Huschenbett. A Kleene-Schützenberger theorem for trace series over bounded lattices. In *Proceedings of NCMA 2010*, volume 263 of *books@acg.at*, pages 99–111, 2010.

[26] S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. Shannon and J. McCarthy, editors, *Automata Studies*, chapter 1, pages 3–42. Princeton University Press, 1956.

[27] V. Kreinovich. Towards more realistic (e.g., non-associative) "and"- and "or"-operations in fuzzy logic. *Soft Computing*, 8(4):274–280, 2004.

[28] D. Krob. The equality problem for rational series with multiplicities in the tropical semiring is undecidable. *International Journal of Algebra and Computation*, 4:405–425, 1994.

[29] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*, volume 5 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1986.

[30] O. Kupferman and Y. Lustig. Lattice automata. In *Proceedings of VMCAI 2007*, volume 4349 of *LNCS*, pages 199–213, 2007.

[31] D. Kuske. Weighted asynchronous cellular automata. *Theoretical Computer Science*, 374:127–148, 2006.

[32] D. Kuske. Weighted and unweighted trace automata. *Acta Cybernetica*, 19:393–410, 2009.

[33] A. Mallya. Deductive multi-valued model checking. In *Proceedings of ICLP 2005*, volume 3668 of *LNCS*, pages 297–310, 2005.

[34] A. Mazurkiewicz. Concurrent program schemes and their interpretation. Technical Report DIAMA PB 78, Arhus University, 1977.

[35] I. Meinecke. Weighted logics for traces. In *Proceedings of CSR 2006*, volume 3976 of *LNCS*, pages 235–246, 2006.

[36] E. Ochmańsky. Regular behaviour of concurrent systems. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 27:56–67, 1985.

[37] E. Ochmańsky. Recognizable trace languages. In Diekert and Rozenberg [9], chapter 6, pages 167–204.

[38] M. O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.

[39] M. P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4:245–270, 1961.

[40] W. Thomas. On logical definability of trace languages. In V. Diekert, editor, *Proceeding of the ASMICS-Workshop "Free Partially Commutative Monoids"*, Rep. TUM-I9002, pages 172–182. TU München, 1990.

[41] W. Zielonka. Notes on finite asynchronus automata. *RAIRO - Informatique Théoretique et Applications*, 21:99–135, 1987.

[42] W. Zielonka. Asynchronous automata. In Diekert and Rozenberg [9], chapter 7, pages 205–247.

Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort, Datum	Unterschrift