Weighted Tree Generating Regular Systems and Crisp-Determinization of Weighted Tree Automata

Summary of PhD Thesis

Dávid Kószó
Supervisor: Zoltán Fülöp, DSc

Doctoral School of Computer Science
Department of Foundations of Computer Science
Faculty of Science and Informatics
University of Szeged

Szeged
2023

1 Introduction

In computer science, a tree is a widely used abstract data type. In particular, we can use trees to represent or manipulate hierarchical data. For instance, each of the following applications involves a tree-like abstract data type: the directory structure of each file system, the class-hierarchy in object-oriented programming without allowing multiple inheritance, abstract syntax trees for computer languages, parse trees in Natural Language Processing (NLP), Document Object Models ("DOM tree") of XML and HTML documents, *etc.* Interestingly, even JSON and YAML documents can be considered as trees, but they are typically represented in a different way.

In the PhD thesis we deal only with finite trees over ranked alphabets. A *ranked alphabet* Σ is a finite and nonempty set of symbols in which we associate with each symbol a unique rank, *i.e.*, a nonnegative integer. For each nonnegative integer k, we denote the set of all symbols in Σ of rank k by $\Sigma^{(k)}$. Then a *tree over* Σ is a finite, labeled, and ordered tree such that if a node of the tree has k children, then that node is labeled by an element of $\Sigma^{(k)}$. The set of all trees over Σ is denoted by T_Σ. Furthermore, each subset of T_Σ is called a *tree language over* Σ.

The classical model of finite-state tree automata (for short: fta) [45, 50–52] was invented to recognize a tree language over some ranked alphabet. An *fta* A over a ranked alphabet Σ consists of a finite and nonempty set Q (states), a family $\delta = (\delta_k \mid k \text{ is an integer})$ of relations $\delta_k \subseteq Q^k \times \Sigma^{(k)} \times Q$ (k-ary transitions), and a set $F \subseteq Q$ (root states). Then a *tree* ξ *over* Σ *is recognized by* A if we can associate to each node of ξ a state in the following way: (1) if a node is labeled by a symbol $\sigma \in \Sigma^{(k)}$ and the states associated to that node and its k children are q and $q_1, \ldots, q_k$, respectively, then $(q_1, \ldots, q_k, \sigma, q)$ is a k-ary transition in δ_k and (2) the state associated to the root of ξ is a root state. The tree language recognized by A is called a *recognizable tree language*. Moreover, two fta are said to be *equivalent* if they recognize the same tree language. It is well known that with fta qualitative properties of recognizable tree languages can be described, such as emptiness, finiteness, *etc.*. For surveys on the theory of fta we refer to [19, 29, 38].

In parallel and later, further concepts were introduced and proved to be equivalent to fta such as tree generating regular systems (for short: tgrs) [16]; rational tree languages [29, 38, 53]; monadic second-order logic for trees [20, 53]; regular tree grammars [16, 38]; representable tree languages [38]. Moreover, each tree language recognized by an fta is the image of a local tree language under a deterministic tree relabeling [29, 38, 51].

Later the idea came up to describe not only qualitative but also quantitative properties of recognizable tree languages, like degree of ambiguity or costs of acceptance. Clearly, each tree language can be considered as a mapping from the set of input trees to the Boolean semiring $\{0, 1\}$. Moreover, by replacing the Boolean semiring in such a mapping by any other semiring B, and allowing that the mapping associates arbitrary elements of B to the trees, a way was opened to describe also those quantitative properties. More precisely, each quantitative property can be interpreted as a mapping from the set of input trees to some semiring or more generally to the carrier set of some weight structure. Mappings describing quantitative properties of tree languages are called *weighted tree languages* (or *formal power series over trees*). To recognize such weighted tree languages, the model of weighted tree automata (for short: wta) was invented. The concept of *wta* is a natural extension of the concept of fta by adding weights to each transition and to each state. More precisely, a *wta* $\mathcal{A}$ over a ranked alphabet Σ and a weight structure B with carrier set B (for short: (Σ, B)-wta) consists of a finite and nonempty set Q (states), a family $\delta = (\delta_k \mid k \text{ is an integer})$ of mappings $\delta_k : Q^k \times \Sigma^{(k)} \times Q \to B$ (k-ary transition weight mapping), and a mapping $F : Q \to B$ (root weight mapping). Then the operations of B allow to combine the transition weights while processing the input tree.

The first such wta over a complete distributive lattice was introduced in [41] (also see [32]) under the name fuzzy tree automata. Over the years, several other weight algebras were used to enrich the expressive power of wta: *e.g.*, fields [9], commutative semirings [7], multioperator monoids [34, 35, 42, 43], strong bimonoids [1, 3, 47], and tree-valuation monoids [24]. In the thesis we consider the model of wta over strong bimonoids. A *strong bimonoid* $\mathsf{B} = (B, \oplus, \otimes, \mathbb{0}, \mathbb{1})$ [18, 26, 28, 47] is basically a semiring in which $\otimes$ does not necessarily distribute over $\oplus$.

For a wta $\mathcal{A}$ over a strong bimonoid B, two semantics can be defined: the initial algebra semantics and the run semantics [3, 36, 37, 47]. In general, the two kinds of semantics may differ [26]; however, if B is a semiring or $\mathcal{A}$ is bottom-up deterministic[1], then they coincide, see, *e.g.*, [12, Lm. 4.1.13] and [47, Thm. 4.1] and [3, Thm. 3.10]. In the thesis we deal only with the run semantics.

A *run of $\mathcal{A}$ on a tree ξ* is a mapping ρ from the nodes of ξ to Q. Moreover, a run ρ is said to be a *q-run* if it associates to the root of ξ the state $q \in Q$. The *weight of a run ρ of $\mathcal{A}$ for ξ*, denoted by $\mathrm{wt}(\xi, \rho)$, is the element in B defined by induction as follows: if $\xi = \sigma(\xi_1, \ldots, \xi_k)$ for some natural number k, $\sigma \in \Sigma^{(k)}$, and $\xi_1, \ldots, \xi_k \in \mathrm{T}_\Sigma$, then

$$\mathrm{wt}(\xi, \rho) = \Big(\bigotimes_{i=1}^{k} \mathrm{wt}_{\mathcal{A}}(\xi_i, \rho|_i) \Big) \otimes \delta_k\big(\rho(1)\cdots\rho(k), \sigma, \rho(\varepsilon)\big) \quad , \tag{1}$$

where $\rho|_i$ denotes a restriction of ρ such that $\rho|_i$ is a run of $\mathcal{A}$ on ξ_i for each $i \in \{1, \ldots, k\}$. Then the *(run) semantics of $\mathcal{A}$*, denoted by $[\![\mathcal{A}]\!]$, is the mapping $[\![\mathcal{A}]\!] : \mathrm{T}_\Sigma \to B$ defined, for each $\xi \in \mathrm{T}_\Sigma$, by

$$[\![\mathcal{A}]\!](\xi) = \bigoplus_{q \in Q} \quad \bigoplus_{\rho \text{ is a } q\text{-run of } \mathcal{A} \text{ on } \xi} \quad \mathrm{wt}(\xi, \rho) \otimes F(q) \quad .$$

Furthermore, two wta $\mathcal{A}$ and $\mathcal{A}'$ are said to be *equivalent* if we have $[\![\mathcal{A}]\!] = [\![\mathcal{A}']\!]$.

The theory of wta has a huge literature. Several questions have been studied throughout the years, *e.g.*, the pumping lemma for wta [11] and the determinization problem for wta [14, 17, 36]. Furthermore, similarly to the unweighted case, additional concepts were invented and shown to be equivalent to wta, see, *e.g.*, weighted regular tree grammars [7] and the Kleene theorem for wta [7, 25]; monadic second-order logic and the Büchi-Elgot-Trakhtenbrot's theorem for recognizable weighted tree languages [27, 35] (*cf.* [21–23] for the string case); and weighted representable tree languages [39, 40]. It is also known that each weighted tree language recognized by a wta is the image of a local weighted tree language under a deterministic tree relabeling [33]. For a survey on the theory of wta we refer to [30, 36, 37].

In the thesis we deal with two research topics. The first topic is the equivalence of wta and weighted tree generating regular systems (for short: wtgrs) over semirings. In [4] the concept of wtgrs over a strong bimonoid was introduced as a natural extension of

[1] A (Σ, B)-wta $\mathcal{A}$ is said to be *bottom-up deterministic* if, for every natural number k, $w \in Q^k$, and $\sigma \in \Sigma^{(k)}$, there exists at most one $q \in Q$ such that $\delta_k(w, \sigma, q) \neq \mathbb{0}$.

the concept of tree generating regular system (for short: tgrs) [16] to the weighted case, but, despite the expectations, the semantics of wtgrs was not defined as a straightforward generalization of the original semantics of tgrs. More precisely, in [4] an alternative, but essentially equivalent semantics was introduced for tgrs, of which the generalization to the weighted case opens a way to prove the equivalence of tgrs and Boolean wtgrs, and the desired equivalence of wta and wtgrs (like the equivalence of fta and tgrs in [16]). In the thesis, the main results of [4] are recalled as Theorems 4.2.8 and 4.3.4 and 4.4.5.

The second topic is the crisp-determinization problem. The *determinization problem* shows up if we wish to specify a problem (*e.g.*, a tree language) in a nondeterministic way and to calculate its solution (*e.g.*, membership) in a deterministic way. More precisely, the determinization problem asks the following: for a given nondeterministic device $\mathcal{A}$ of a given type (or class), does there exist a bottom-up deterministic device $\mathcal{A}'$ of the same type which is equivalent to $\mathcal{A}'$?

It is well known that the determinization problem is solved positively for the class of all fta (*cf.*, *e.g.*, [53, Thm. 1], [29, Thm. 3.8] , and [38, Thm. 2.2.6]), *i.e.*, for each fta A, there is an equivalent bottom-up deterministic fta A'. The construction of A' from A is called powerset construction. However, the situation changes drastically if we consider the class of all wta. More precisely, there exists a wta to which there does not exist an equivalent bottom-up deterministic wta [7, 9, 31, 42]. On the other side, there are subclasses of the class of all wta for which the determinization problem can be solved positively [14, Cor. 4.9 and Thm. 4.24], [36, Thm. 3.17], and [17, Thm. 5.2].

A special case of determinization of wta over strong bimonoids is when we require that the resulting deterministic wta is crisp-deterministic. We call a wta *crisp-deterministic* if it is total[2] and bottom-up deterministic, and each of its transitions carries either the additive unit $\mathbb{0}$ or the multiplicative unit $\mathbb{1}$ of the underlying strong bimonoid B; weights different from these units may only appear at the root of the given input tree. Then the *crisp-determinization problem* (of wta over strong bimonoids) deals with the following question: for a given wta $\mathcal{A}$, does there exist a crisp-deterministic wta $\mathcal{A}'$ such that $\mathcal{A}'$ is equivalent to $\mathcal{A}$? If the answer to this question is "yes", *i.e.*, such a wta $\mathcal{A}'$ exists, then we say that the wta $\mathcal{A}$ is *crisp-determinizable*.

It is clear that the notion of crisp-deterministic wta is quite restrictive. However, in spite of this fact, it is worth to investigate crisp-deterministic wta as they have a strong relationship with fuzzy questions (*cf.* [37, Ch. 19]).

Recent results in connection with the crisp-determinization problem are published in [1–3]. In [3] a sufficient condition for crisp-determinization was given. In the thesis we recall that result as Theorem 5.2.8 (also *cf.* Theorem 5.2.12). Moreover, in [3], regarding the crisp-determinization problem, undecidability results were given. Inspired by those undecidability results, in the thesis we prove two undecidability results (*cf.* Theorems 5.3.7 and 5.3.14). Finally, in [1], also regarding the crisp-determinization problem, a decidability result was given. In the thesis we recall that decidability result as Theorem 5.4.15. To prove the decidability result, in [1, 2] two new pumping lemmas for wta were presented. In the thesis we recall those pumping lemmas as Theorems 3.2.3 and 3.2.4.

[2] A (Σ, B)-wta $\mathcal{A}$ is said to be *total* if, for every natural number k, $w \in Q^k$, and $\sigma \in \Sigma^{(k)}$, there exists at least one $q \in Q$ such that $\delta_k(w, \sigma, q) \neq \mathbb{0}$.

2 Pumping lemmas

In the thesis we first deal with pumping lemmas for runs of wta. We use them to prove our decidability result (*cf.* Theorem 5.4.15). With a pumping lemma one can achieve structural implications on small or particular large trees (*cf.* [38, Lm. 2.10.1] and [11, Lm. 5.5]). Such pumping lemmas already exist for wta (*cf.* [11, Sect. 5]). However, Borchardt's setting in [11] deals with bottom-up deterministic wta over semirings and employs initial algebra semantics, whereas in our setting we deal with (arbitrary) wta over strong bimonoids and employ run semantics. Nevertheless, if we consider the class of all bottom-up deterministic wta over semirings, then the two settings coincide.

Next we explain our pumping lemma. Essentially, we follow the classical approach for fta with an analysis of Equality (1). More precisely, let $\mathcal{A} = (Q, \delta, F)$ be a (Σ, B)-wta, $\xi \in \mathrm{T}_\Sigma$ with $\mathrm{height}(\xi) \geq |Q|$, and ρ be a run of $\mathcal{A}$ on ξ. As for fta, we choose a path, *i.e.*, a linearly ordered subset of positions, in ξ of which the length equals $\mathrm{height}(\xi)$. Clearly, since $\mathrm{height}(\xi) \geq |Q|$, there exist distinct positions u and v in this path with $\rho(u) = \rho(v)$ in Q. Assume that u is above v, *i.e.*, there exists a position w such that $v = uw$. Now we consider the subtree $\xi|_u$ (respectively, $\xi|_v$) of ξ at u (respectively, v) comprising all positions of ξ which are equal to or below u (respectively, v) (*cf.* Figure 1).

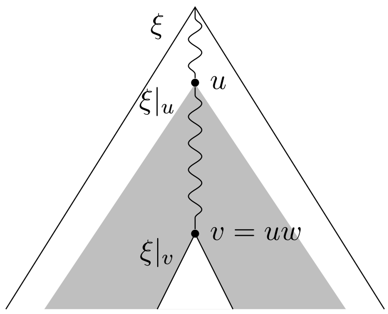

Figure 1: *Illustration of the tree ξ, the positions u and v, and the subtrees $\xi|_u$ and $\xi|_v$.*

In Figure 1 the shaded part is called a context. Evidently, since $\rho(u) = \rho(v)$, we can cut out that context from ξ, *i.e.*, we can replace the subtree $\xi|_u$ by $\xi|_v$, and thus, we can obtain a smaller tree for which a restriction of the run ρ leads to the same state as ρ. But, we can also substitute a copy of that context at position v and copy the corresponding part of the run ρ, and hence, obtain a larger tree ξ' and a run ρ' of $\mathcal{A}$ on ξ' leading again to the same root state as ρ.

A careful extension and analysis of Equality (1) shows that the product of weights of a run θ on a context c can be split into two factors, a 'left one', denoted by $l_{c,\theta}$, and a 'right one', denoted by $r_{c,\theta}$. Hence, if we substitute a copy of a context c into a tree ξ, then, in order to calculate the weight of the run ρ' on the resulting new tree ξ', we just replace the two factors $l_{c,\theta}$ and $r_{c,\theta}$ by their powers. In fact, it turns out that, to calculate $\mathrm{wt}(\xi', \rho')$, an additional 'left factor' $l_{c',\theta'}$ and an additional 'right factor' $r_{c',\theta'}$ are required, which come from dividing the context c' between the root of ξ and the position u (*cf.* Figure 1). Now we show our pumping lemmas.

Theorem 3.2.3. *[1, Thm. 8] and [2, Lm. 5.3] (also* cf. *[11, Lm. 5.3]) Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$, and B be a strong bimonoid. Moreover, let $\mathcal{A} = (Q, \delta, F)$ be a (Σ, B)-wta. Then, for every Σ-contexts c' and c, Σ-tree ξ, states q' and q in Q, (q', q)-run θ' of $\mathcal{A}$ on c', (q, q)-run θ of $\mathcal{A}$ on c, and q-run ρ of $\mathcal{A}$ on ξ, and for each $n \in \mathbb{N}$, we have*

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

Theorem 3.2.4. *[1, Thm. 9] and [2, Thm. 5.4] (also cf. [11, Lm. 5.5]) Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$, and B be a strong bimonoid. Moreover, let $\mathcal{A} = (Q, \delta, F)$ be a (Σ, B)-wta. For every Σ-tree ξ', state q' in Q, and q'-run ρ' of $\mathcal{A}$ on ξ', if $\mathrm{height}(\xi') \geq |Q|$, then there exist Σ-contexts c' and c, Σ-tree ξ, state q in Q, (q', q)-run θ' of $\mathcal{A}$ on c', (q, q)-run θ of $\mathcal{A}$ on c, and q-run ρ of $\mathcal{A}$ on ξ such that the following conditions hold true: $\xi' = c'[c[\xi]]$, $\rho' = \theta'[\theta[\rho]]$, $\mathrm{height}(c) > 0$, $\mathrm{height}(c[\xi]) < |Q|$, and, for each $n \in \mathbb{N}$, we have*

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

Detailed description of Theorems 3.2.3 and 3.2.4 can be found in Section 3.2 of the thesis.

Contribution. The author of the thesis declares that his contribution to Theorems 3.2.3 and 3.2.4 is significant, and also that Theorems 3.2.3 and 3.2.4 are published in [1, 2].

3 Weighted tree generating regular systems

The concept of weighted tree generating regular system over a ranked alphabet Σ and a strong bimonoid B (for short: (Σ, B)-wtgrs, or just: wtgrs) was introduced in [4]. Then the equivalence of wta and weighted tree generating regular systems over semirings was proven, *i.e.*, a further characterization of recognizable weighted tree languages was given in [4]. The concept of (Σ, B)-wtgrs was defined in the way that the following two requirements were fulfilled:

(a) Each wtgrs $\mathcal{S}$ over Σ and the Boolean semiring $\{0, 1\}$ is "equivalent" to a tgrs S over Σ, and vice versa (*cf.* Theorem 4.3.4).

(b) Under some mild conditions, each wtgrs $\mathcal{S}$ over Σ and a semiring B is equivalent to a (Σ, B)-wta $\mathcal{A}$, and vice versa (*cf.* Theorem 4.4.5) (correspondingly to the fact that tgrs and fta are equivalent, *cf.* [16]). (2)

To fully understand those results, here we briefly recall the concept of tgrs and its derivation semantics introduced by Brainerd in [16]. Moreover, we show that the seemingly natural generalization of the derivation semantics to the weighted case does not work, *i.e.*, it does not fulfill Requirement (2)(b). Finally, we explain the two characteristics of our alternative semantics, called reduction semantics. In fact, the reduction semantics is essentially the same as the derivation semantics (*cf.* Theorem 4.2.8).

A Σ-*tgrs* (or just *tgrs*) S consists of a ground term rewriting system [8, 19] P over some ranked alphabet Δ and a finite subset Z of designated trees over Δ. The ranked alphabet Δ is partitioned into two sets: the set Σ of *terminals* and the set N of *nonterminals*. Moreover, we call elements of P *productions* and elements of Z *axioms*. The ground term rewrite relation $\Rightarrow_S$ induced by S is defined in the standard way (*cf.* [8, Def. 3.1.8]). Furthermore, the *derivation semantics of* S (for short: *d-semantics of* S), denoted by $\mathrm{L_d}(S)$, is the tree language $\mathrm{L_d}(S) \subseteq \mathrm{T}_\Sigma$ defined as follows: a ξ over Σ is in $\mathrm{L_d}(S)$ if there exist an axiom $\zeta \in Z$ and a ζ-computation of P for ξ under $\Rightarrow_S$, *i.e.*, $\zeta \Rightarrow_S^* \xi$. Moreover, a tree language is said to be *d-generated* if it is a d-semantics of some Σ-tgrs.

Observe that if in certain steps of a ζ-computation d of P for ξ under $\Rightarrow_S$ we could replace at incomparable positions[3], then there may exist several other ζ-computations of P for ξ under $\Rightarrow_S$.

Now we define a (Σ, B)-wtgrs to be a Σ-tgrs in which to each production and to each axiom a weight in B is associated, *i.e.*, a (Σ, B)-*wtgrs* $\mathcal{S}$ consists of a Σ-tgrs $S = (N, Z, P)$, a mapping $wt : P \to B$ (production weight mapping), and a mapping $X : Z \to B$ (axiom weight mapping). The natural generalization of the d-semantics of tgrs to the weighted case, *i.e.*, the d-semantics of $\mathcal{S}$, would be as follows. For a tree ξ over Σ and an axiom $\zeta \in Z$, and for a ζ-computation d of P for ξ under $\Rightarrow_S$, to calculate the weight of d, we would multiply the weights of the productions in a fixed order determined by d by applying the multiplication operation $\otimes$ of B. Then we calculate the d-semantics of $\mathcal{S}$ for a tree ξ over Σ as follows: by using the addition operation $\oplus$ of B we sum up all weights of ζ-computations of P for ξ under $\Rightarrow_S$ multiplied by the axiom weight $X(\zeta)$. However, this is not suitable to fulfill Requirement (2)(b) for the following reason. When we associate a (Σ, B)-wtgrs $\mathcal{S}$ to a (Σ, B)-wta $\mathcal{A}$, more than one computation of P may correspond to a single run of $\mathcal{A}$. Furthermore, since $\oplus$ is not necessarily idempotent, this may yield that the d-semantics of $\mathcal{S}$ and the semantics of $\mathcal{A}$ differ.

In order to avoid that phenomenon, we advocate an alternative semantics, called *reduction semantics* (for short: *r-semantics*), for tgrs. The d-semantics and the r-semantics of tgrs are essentially equivalent (*cf.* Theorem 4.2.8). Moreover, we introduce the concept of wtgrs with the natural generalization of the r-semantics of tgrs to the weighted case. The r-semantics of a tgrs S has two characteristics:

(i) it is based on a restriction of the term rewriting relation, denoted by $\Rightarrow_{S,\mathrm{dp}}$, in which replacements can be performed only at the minimal position (with respect to the depth-first post-ordering of positions) at which a replacement is possible and

(ii) the r-semantics of S, denoted by $\mathrm{L_r}(S)$, is the tree language $\mathrm{L_r}(S) \subseteq \mathrm{T}_\Sigma$ defined such that a tree ζ over Σ is in $\mathrm{L_r}(S)$ if there exists an axiom $\xi \in Z$ and a ζ-computation of P for ξ under $\Rightarrow_{S,\mathrm{dp}}$.

Then a tree language is said to be *r-generated* if it is an r-semantics of some Σ-tgrs.

In conclusion, we introduce the r-semantics for the following reasons. For each tgrs S there exists a tgrs S' such that $\mathrm{L_d}(S) = \mathrm{L_r}(S')$. Vice versa, for each tgrs S there exists a tgrs S' such that $\mathrm{L_r}(S) = \mathrm{L_d}(S')$. Hence, we obtain the following result.

[3] We call two positions of a tree *incomparable* if none of them is a prefix of the other one.

Theorem 4.2.8. *[4, Thm. 15] Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$. Then, for each $L \subseteq \mathrm{T}_\Sigma$, the Σ-tree language L is d-generated if and only if it is r-generated.*

The semantics of a (Σ, B)-wtgrs $\mathcal{S}$, denoted by $[\![\mathcal{S}]\!]$, is introduced by the natural generalization of the r-semantics of tgrs to the weighted case.

Since $[\![\mathcal{S}]\!]$ is a weighted tree language over Σ and B, the *support of* $[\![\mathcal{S}]\!]$ *with respect to* B, denoted by $\mathrm{supp}_{\mathsf{B}}([\![\mathcal{S}]\!])$, is the tree language $\mathrm{supp}_{\mathsf{B}}([\![\mathcal{S}]\!]) \subseteq \mathrm{T}_\Sigma$ defined such that a tree $\zeta \in \mathrm{T}_\Sigma$ is in $\mathrm{supp}_{\mathsf{B}}([\![\mathcal{S}]\!])$ if we have $[\![\mathcal{S}]\!](\zeta) \neq \mathbb{0}$. Then the equivalence of tgrs and wtgrs can be formulated as follows (*cf.* Requirement (2)(a)).

Theorem 4.3.4. *[4, Thm. 25] Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$. Moreover, let L be a Σ-tree language. Then the following statements are equivalent.*

1. *We can construct a Σ-tgrs such that $\mathrm{L}_{\mathrm{r}}(S) = L$.*
2. *We can construct a (Σ, Boole)-wtgrs such that $\mathrm{supp}_{\mathsf{Boole}}([\![\mathcal{S}]\!]) = L$.*

A weighted tree language ψ is said to *r-generated* if there exists a wtgrs $\mathcal{S}$ such that $\psi = [\![\mathcal{S}]\!]$. Using this concept, the equivalence of wta and wtgrs over semirings can be stated as follows (*cf.* Requirement (2)(b)).

Theorem 4.4.5. cf. *[4, Thm. 34] Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$. Then, for every semiring B and (Σ, B)-weighted tree language ψ, the following statements hold true.*

1. *If B is complete, then ψ is recognizable iff it is r-generated.*
2. *If B is computable, then we can construct a (Σ, B)-wta $\mathcal{A}$ such that $[\![\mathcal{A}]\!] = \psi$ iff we can construct a finite-reductional (Σ, B)-wtgrs $\mathcal{S}$ such that $[\![\mathcal{S}]\!] = \psi$.*

Detailed description of Theorems 4.2.8, 4.3.4, and 4.4.5 can be found in Sections 4.2, 4.3, and 4.4 of the thesis, respectively.

Contribution. The author of the thesis declares that Theorems 4.2.8, 4.3.4, and 4.4.5 are due to his own work, and those results are published in [4].

4 Crisp-determinization of wta

A crisp-deterministic wta $\mathcal{A}$ over B has several desirable properties such as $[\![\mathcal{A}]\!]$ has a finite image (called *finite-image property*) or, for each $b \in B$, the set of all trees with weight b under $[\![\mathcal{A}]\!]$ is a recognizable tree language (called *preimage property*). In fact, the class of all crisp-deterministic wta can be characterized using only those two properties *cf.* [3]. For further properties of crisp-deterministic wta we refer to [37]. It is worth to study crisp-deterministic wta also for the following reason. Fuzzy automata, languages, and grammars have been of interest for a long time *e.g.* [15, 32, 41]; for a survey we refer to [48]. The underlying weight structure of these formal models is some bounded lattice. Recall that, each bounded lattice is, basically, a bi-locally finite strong bimonoid. In fact, in [3] it was shown that each wta over a bi-locally finite strong bimonoid is crisp-determinizable, *i.e.*, advantages of crisp-deterministic wta are available during the investigation of fuzzy formal models.

There are subclasses of all weighted string automata (for short: wsa) for which the crisp-determinization problem is solved positively [18, 26]. They correspond to subclasses of wta over string ranked alphabets[4], *cf.* [37, Lm. 3.3.3]. In [3] the positive results of [18, 26] were extended to further subclasses of all wta as follows. By a straightforward generalization of [18, Sect. 8] from strings to trees, the concept of the finite-order property of wta was introduced. Moreover, it was shown that if a (Σ, B)-wta $\mathcal{A}$ has finite order, then $\mathcal{A}$ is crisp-determinizable. In the thesis that result is given as follows.

Theorem 5.2.8. *[3, Thm. 7.3] (also* cf. *[18, Thm. 8.2]) Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$, and B be a strong bimonoid. Moreover, let $\mathcal{A}$ be a (Σ, B)-wta such that $\mathcal{A}$ has finite order. Then there exists a (Σ, B)-wta $\mathcal{A}'$ such that $\mathcal{A}'$ is crisp-deterministic and it is equivalent to $\mathcal{A}$.*

If, in addition, B is computable, then we can even construct $\mathcal{A}'$.

Theorem 5.2.12. *Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$, and B be a computable strong bimonoid. Moreover, let $\mathcal{A}$ be a (Σ, B)-wta such that $\mathcal{A}$ has finite order. Then we can construct a (Σ, B)-wta $\mathcal{A}'$ such that $\mathcal{A}'$ is crisp-deterministic and it is equivalent to $\mathcal{A}$.*

Detailed discussion of Theorems 5.2.8 and 5.2.12 can be found in Section 5.2 of the thesis. Furthermore, in the thesis we deal with decidability questions related to the crisp-determinization problem. In the literature there had been some promising partial results regarding the undecidability (decidability) of crisp-determinization. These results justify the relevance of such questions, and create a solid base for further investigations. For instance, each wsa over a finite semiring or over the semiring of natural numbers has the preimage property; or each wsa over a commutative ring which has the finite-image property also has the preimage property [10, 27, 44]. Moreover, for each wsa over any subsemiring of the rational numbers, the finite-image property is decidable [46] (also *cf.* the classical Burnside property for semigroups [49]). Keeping in mind these existing partial results, undecidability (decidability) results related to the crisp-determinization problem were proven in [1–3] as follows.

By Theorems 5.2.8 and 5.2.12, a wta $\mathcal{A}$ is crisp-determinizable if $\mathcal{A}$ has finite order. Hence, in particular, we are interested in the following decidability questions.
(Q1) Is it decidable for an arbitrary wta $\mathcal{A}$, whether $\mathcal{A}$ has finite order?
(Q2) Is it decidable for an arbitrary wta $\mathcal{A}$, whether $\mathcal{A}$ is crisp-determinizable?
In the thesis we recall that the answer to both questions is negative.

Theorem 5.3.7. cf. *[3, Thm. 8.9] It is undecidable, for arbitrary string ranked alphabet Σ, computable and idempotent semiring S, and bottom-up deterministic (Σ, S)-wta $\mathcal{A}$, whether $\mathcal{A}$ has finite order.*

[4] A ranked alphabet Σ is called a *string ranked alphabet* if $\Sigma = (\Sigma^{(1)} \cup \Sigma^{(0)})$, $|\Sigma^{(1)}| \geq 1$, and $|\Sigma^{(0)}| = 1$.

Theorem 5.3.14. cf. *[3, Thm. 8.5] It is undecidable, for arbitrary string ranked alphabet Σ, computable and idempotent semiring S, and bottom-up deterministic (Σ, S)-wta $\mathcal{A}$, whether $\mathcal{A}$ is crisp-determinizable.*

Detailed discussion of Theorems 5.3.7 and 5.3.14 can be found in Section 5.3 of the thesis. Finally, in [1] two subclasses of wta were identified for which the crisp-determinization problem is decidable. For that, in the spirit of [13, Def. 12], the concept of past-finite monotonic strong bimonoid was introduced. These particular weight structures have several desirable properties *cf.* [13]. Hence, if B is past-finite monotonic, then the characterization of crisp-determinizability given in [3] can be simplified as follows: for an arbitrary (Σ, B)-wta $\mathcal{A}$, the wta $\mathcal{A}$ is crisp-determinizable if and only if $\mathrm{im}(\llbracket\mathcal{A}\rrbracket)$ is finite. Moreover, if, in addition, B is additively locally finite or $\mathcal{A}$ is unambiguous, then $\mathrm{im}(\llbracket\mathcal{A}\rrbracket)$ is finite. Consequently, the following result can be obtained.

Theorem 5.4.15. *[1, Thm. 10] Let Σ be a ranked alphabet such that $\Sigma^{(0)} \neq \emptyset$. Moreover, let $\mathsf{B} = (B, \oplus, \otimes, \mathbb{0}, \mathbb{1})$ be a strong bimonoid and $\preceq$ be a partial ordering on B such that B is past-finite monotonic with respect to $\preceq$, and B has effective tests for $\{\mathbb{0}, \mathbb{1}\}$. Then the following statements hold true.*

1. *If, in addition, B is additively locally finite, then it is decidable, for arbitrary (Σ, B)-wta $\mathcal{A}$, whether $\mathcal{A}$ is crisp-determinizable.*
2. *It is decidable, for arbitrary unambiguous (Σ, B)-wta $\mathcal{A}$, whether $\mathcal{A}$ is crisp-determinizable.*

Detailed discussion of Theorem 5.4.15 can be found in Section 5.4.

Contribution. The author of the thesis declares that his contribution to Theorems 5.2.8, 5.2.12, 5.3.7, 5.3.14, and 5.4.15 is decisive, that Theorems 5.2.8 and 5.4.15 are published in [3] and [1], respectively, and also that Theorems 5.3.7 and 5.3.14 are slightly stronger than [3, Thm. 8.9] and [3, Thm. 8.5], respectively, but are based on the same ideas. Finally, we mention that [2, Thm. 6.6], [5, Thms. 7 and 11], and [2, Thms. 7.5, 7.7, and 7.15] supersede Theorems 5.2.8, 5.3.14, and 5.4.15, respectively, but the contribution of the author to those stronger results is not decisive.

Declaration

In the PhD thesis of **Dávid Kószó** entitled **Weighted Tree Generating Regular Systems and Crisp-Determinization of Weighted Tree Automata**, the contribution of **Dávid Kószó** was decisive in the following results:

- Theorem 3.2.3 published as [1, Thm. 8] and [2, Lm. 5.3],
- Theorem 3.2.4 published as [1, Thm. 9] and [2, Thm. 5.4],
- Theorems 4.2.8, 4.3.4, and 4.4.5 published as [4, Thm. 15], [4, Thm. 25], and [4, Thm. 34], respectively,
- Theorem 5.2.8 published as [3, Thm. 7.3],
- Theorem 5.2.12,
- Theorems 5.3.7 and 5.3.14 based on the same ideas like [3, Thm. 8.9] and [3, Thm. 8.5], respectively, and
- Theorem 5.4.15 published as [1, Thm. 10].

These results cannot be used to obtain an academic research degree, other than the submitted PhD thesis of **Dávid Kószó**.

Place and date: Szeged, Hungary, 31.03.2023

Kószó Dávid	
Dávid Kószó	Zoltán Fülöp DSc
candidate	*supervisor*

The head of the Doctoral School of Computer Science declares that the declaration above was sent to all of the coauthors and none of them raised any objections against it.

Place and date: Szeged, Hungary, 31.03.2023

UNIVERSITY OF SZEGED DOCTORAL SCHOOL OF COMPUTER SCIENCE SZEGED

Márk Jelasity DSc
head of Doctoral School

Publications of the author

On the subjects of the thesis

[1] M. Droste, Z. Fülöp, **D. Kószó**, and H. Vogler. "Crisp-Determinization of Weighted Tree Automata over Additively Locally Finite and Past-Finite Monotonic Strong Bimonoids Is Decidable". In: *Descriptional Complexity of Formal Systems (DCFS 2020)*. Ed. by G. Jirásková and G. Pighizzini. Vol. 12442. Lecture Notes in Computer Science. Springer Nature Switzerland, 2020, 39–51.

[2] M. Droste, Z. Fülöp, **D. Kószó**, and H. Vogler. "Finite-image property of weighted tree automata over past-finite monotonic strong bimonoids". In: *Theoretical Computer Science* 919 (2022), 118–143.

[3] Z. Fülöp, **D. Kószó**, and H. Vogler. "Crisp-determinization of weighted tree automata over strong bimonoids". In: *Discrete Mathematics & Theoretical Computer Science* 23(1) (2021), #18.

[4] **D. Kószó**. "Weighted Tree Generating Regular Systems over Strong Bimonoids with Reduction Semantics". In: *Journal of Automata, Languages and Combinatorics* 27(4) (2022), 271–307.

Further related publications

[5] M. Droste, Z. Fülöp, and **D. Kószó**. "Decidability Boundaries for the Finite-Image Property of Weighted Finite Automata". In: *International Journal of Foundations of Computer Science* (To appear).

[6] **D. Kószó**. "Tree generating context-free grammars and regular tree grammars are equivalent". In: *Annales Mathematicae et Informaticae* 56 (2022), 58–70.

Other references

[7] A. Alexandrakis and S. Bozapalidis. "Weighted grammars and Kleene's theorem". In: *Information Processing Letters* 24(1) (1987), 1–4.

[8] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge: Cambridge University Press, 1998.

[9] J. Berstel and C. Reutenauer. "Recognizable formal power series on trees". In: *Theoretical Computer Science* 18(2) (1982), 115–148.

[10] J. Berstel and Ch. Reutenauer. *Rational Series and Their Languages*. Vol. 12. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1988.

[11] B. Borchardt. "A pumping lemma and decidability problems for recognizable tree series". In: *Acta Cybernetica* 16(4) (2004), 509–544.

[12] B. Borchardt. *The Theory of Recognizable Tree Series*. (Ph.D. thesis, 2004, TU Dresden, Germany). Verlag für Wissenschaft und Forschung, 2005.

[13] B. Borchardt, Z. Fülöp, Z. Gazdag, and A. Maletti. "Bounds for tree automata with polynomial costs". In: *Journal of Automata, Languages and Combinatorics* 10(2–3) (2005), 107–157.

[14] B. Borchardt and H. Vogler. "Determinization of finite state weighted tree automata". In: *Journal of Automata, Languages and Combinatorics* 8(3) (2003), 417–463.

[15] S. Bozapalidis and O. Louscou-Bozapalidou. "Fuzzy tree language recognizability". In: *Fuzzy Sets and Systems* 161 (2010), 716–734.

[16] W. S. Brainerd. "Tree Generating Regular Systems". In: *Information and Control* 14 (1969), 217–231.

[17] M. Büchse, J. May, and H. Vogler. "Determinization of weighted tree automata using factorizations". In: *Journal of Automata, Languages and Combinatorics* 15(3–4) (2010), 229–254.

[18] M. Ćirić, M. Droste, J. Ignjatović, and H. Vogler. "Determinization of weighted finite automata over strong bimonoids". In: *Information Sciences* 180(18) (2010), 3479–3520.

[19] H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. *Tree Automata Techniques and Applications*. Available on: `http://tata.gforge.inria.fr`. 2008.

[20] J. Doner. "Tree acceptors and some of their applications". In: *Journal of Computer and System Sciences* 4 (1970), 406–451.

[21] M. Droste and P. Gastin. "Weighted Automata and Weighted Logics". In: *32nd Int. Colloquium Automata, Languages and Programming (ICALP 2005)*. Ed. by L. Caires, G. F. Italiano, L. Monteiro, C. Palamidessi, and M. Yung. Vol. 3580. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2005, 513–525.

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

[23] M. Droste and P. Gastin. "Weighted Automata and Weighted Logics". In: *Handbook of Weighted Automata*. Ed. by M. Droste, W. Kuich, and H. Vogler. Monographs in Theoretical Computer Science. An EATCS Series. Springer-Verlag, 2009. Chap. 5.

[24] M. Droste, D. Heusel, and H. Vogler. "Weighted unranked tree automata over tree valuation monoids and their characterization by weighted logics". In: *Algebraic Informatics (CAI 2015)*. Vol. 9270. Lecture Notes in Computer Science. Springer, 2015, 90–102.

[25] M. Droste, Chr. Pech, and H. Vogler. "A Kleene Theorem for Weighted Tree Automata". In: *Theory of Computing Systems* 38 (2005), 1–38.

[26] M. Droste, T. Stüber, and H. Vogler. "Weighted finite automata over strong bimonoids". In: *Information Sciences* 180(1) (2010), 156–166.

[27] M. Droste and H. Vogler. "Weighted Tree Automata and Weighted Logics". In: *Theoretical Computer Science* 366(3) (2006), 228–247.

[28] M. Droste and H. Vogler. "Weighted automata and multi-valued logics over arbitrary bounded lattices". In: *Theoretical Computer Science* 418 (2012), 14–36.

[29] J. Engelfriet. *Tree automata and tree grammars*. Tech. rep. DAIMI FN-10. see also: arXiv:1510.02036v1 [cs.FL] 7 Oct 2015. Aarhus, Denmark: Inst. of Mathematics, University of Aarhus, Department of Computer Science, 1975.

[30] Z. Ésik and W. Kuich. *Modern Automata Theory*. see: `https://www.dmg.tuwien.ac.at/kuich/mat.pdf`.

[31] Z. Ésik and W. Kuich. "Formal Tree Series". In: *Journal of Automata, Languages and Combinatorics* 8(2) (2003), 219–285.

[32] Z. Ésik and G. Liu. "Fuzzy tree automata". In: *Fuzzy Sets and Systems* 158(13) (2007), 1450–1460.

[33] Z. Fülöp. "Local Weighted Tree Languages". In: *Acta Cybernetica* 22(2) (2015), 393–402.

[34] Z. Fülöp, A. Maletti, and H. Vogler. "A Kleene theorem for weighted tree automata over distributive multioperator monoids". In: *Theory of Computing Systems* 44 (2009), 455–499.

[35] Z. Fülöp, T. Stüber, and H. Vogler. "A Büchi-like theorem for weighted tree automata over multioperator monoids". In: *Theory of Computing Systems* 50 (2012), 241–278.

[36] Z. Fülöp and H. Vogler. "Weighted tree automata and tree transducers". In: *Handbook of Weighted Automata*. Ed. by M. Droste, W. Kuich, and H. Vogler. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 2009. Chap. 9, 313–403.

[37] Z. Fülöp and H. Vogler. *Weighted Tree Automata – May it be a little more?* see: arXiv:2212.05529v1 [cs.FL] 11 Dec 2022. 2022.

[38] F. Gécseg and M. Steinby. *Tree Automata*. see also: arXiv:1509.06233v1 [cs.FL] 21 Sep 2015. Akadémiai Kiadó, Budapest, 1984.

[39] L. Herrmann. "A Medvedev Characterization of Recognizable Tree Series". In: *21th Int. Conf. on Developments in Language Theory (DLT 2017)*. Ed. by É. Charlier, J. Leroy, and M. Rigo. Vol. 10396. Lecture Notes in Computer Science. Springer, Cham, 2017, 210–221.

[40] L. Herrmann. *Weighted Tree Automata with Storage*. Ph.D. Thesis, 2020, TU Dresden, Germany). 2020.

[41] Y. Inagaki and T. Fukumura. "On the description of fuzzy meaning of context-free languages". In: *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. Academic Press, New York, 1975, 301–328.

[42] W. Kuich. "Formal power series over trees". In: *3rd International Conference on Developments in Language Theory (DLT 1997), Thessaloniki, Greece*. Ed. by S. Bozapalidis. Aristotle University of Thessaloniki, 1998, 61–101.

[43] W. Kuich. "Linear systems of equations and automata on distributive multioperator monoids". In: *Contributions to General Algebra 12 - Proceedings of the 58th Workshop on General Algebra "58. Arbeitstagung Allgemeine Algebra", Vienna University of Technology. June 3-6, 1999*. Ed. by D. Dorninger, G. Eigenthaler, M. Goldstern, H. K. Kaiser, W. More, and W. B. Müller. Verlag Johannes Heyn, 1999, 1–10.

[44] O. Louscou-Bozapalidou. “Some remarks on recognizable tree series”. In: *International Journal of Computer Mathematics* 70 (1999), 649–655.

[45] M. Magidor and G. Moran. *Finite Automata over Finite Trees*. Tech. rep. 30. Hebrew University, Jerusalem, 1969.

[46] A. Mandel and I. Simon. “On finite semigroups of matrices”. In: *Theoretical Computer Science* 5 (1977), 101–111.

[47] D. Radovanović. “Weighted tree automata over strong bimonoids”. In: *Novi Sad Journal of Mathematics* 40(3) (2010), 89–108.

[48] G. Rahonis. “Fuzzy languages”. In: *Handbook of Weighted Automata*. Ed. by M. Droste, W. Kuich, and H. Vogler. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, 2009. Chap. 12, 481–517.

[49] A. Restivo and C. Reutenauer. “Rational Languages and the Burnside Problem”. In: *Theoretical Computer Science* 40 (1985), 13–30.

[50] W. C. Rounds. “Mappings and grammars on trees”. In: *Mathematical Systems Theory* 4(3) (1970), 257–287.

[51] J. W. Thatcher. “Characterizing derivation trees of context-free grammars through a generalization of finite-automata theory”. In: *Journal of Computer and System Sciences* 1(4) (1967), 317–322.

[52] J. W. Thatcher. “Generalized sequential machine maps”. In: *Journal of Computer and System Sciences* 4(4) (1970), 339–367.

[53] J. W. Thatcher and J. B. Wright. “Generalized finite automata theory with an application to a decision problem of second-order logic”. In: *Mathematical Systems Theory* 2(1) (1968), 57–81.

Összefoglalás

Az informatikában a fa egy széles körben használt absztrakt adattípus. Elsősorban arra használjuk ezt az adattípust, hogy függelmi viszonyokat, fölé- és alárendeltségek rendszerét tudjuk kifejezni. A következő gyakorlati alkalmazási területeken találkozhatunk a fa adattípussal: egy fájlrendszer könyvtárszerkezete, többszörös öröklődés nélküli objektumorientált programozásban az osztályhierarchia, a programozási nyelvek absztrakt szintaxis fái, a természetes nyelvi feldolgozásban az elemzőfák, az XML és HTML dokumentumok Dokumentum Objektum Modelljei, stb.

Ebben az értekezésben csak rangolt ábécé feletti véges fákkal foglalkozunk. Az ezen fákat felismerő formális modelleket véges faautomatáknak nevezzük. A véges faautomaták természetes kiterjesztései a súlyozott esetre a súlyozott faautomaták.

Az értekezés két fő témakörből áll. Az első témakör keretében a súlyozott faautomaták és a súlyozott fageneráló reguláris rendszerek ekvivalenciáját tárgyaljuk. A súlyozott fageneráló reguláris rendszer fogalma a fageneráló reguláris rendszer fogalmának természetes általánosítása a súlyozott esetre, azonban a szemantika kiterjesztése a súlyozott esetben nem közvetlenül történik. Az értekezés 4.2-es fejezetében egy másik, de a meglévővel ekvivalens szemantikát definiálunk a fageneráló reguláris rendszereknek. Majd a 4.3-as fejezetben a súlyozott fageneráló reguláris rendszer szemantikáját már ennek a másik szemantikának a kiterjesztésével adjuk meg. Végül a 4.4-es fejezetben bebizonyítjuk a korábban említett ekvivalenciát.

A másik témakör a súlyozott faautomaták egységdeterminizálása, amely a determinizálás speciális esete. Az 5.2-es fejezetben egy elegendőségi feltételt adunk meg erre vonatkozóan. Az 5.3-as fejezetben az egységdeterminizálásra vonatkozó eldönthetetlenségi, míg az 5.4-es fejezetben eldönthetőségi eredményeket közlünk. Az eldönthetőségi eredmények bizonyítása közben mellékeredményként új pumpáló lemmákat is bebizonyítunk, melyeket a 3.2-es fejezetben tárgyalunk.