Approximated Determinisation of Weighted Tree Automata*

Frederic Dörband[1], Thomas Feller[2], and Kevin Stier[3]

[1] Technische Universität Dresden, Faculty of Computer Science, Germany
frederic.doerband@tu-dresden.de, orcid.org/0000-0003-2942-0896
[2] Technische Universität Dresden, Faculty of Computer Science, Germany
thomas.feller@tu-dresden.de, orcid.org/0000-0001-8420-6118
[3] Universität Leipzig, Institute of Computer Science, Germany
stier@informatik.uni-leipzig.de, orcid.org/0000-0002-0564-8688

Abstract. We introduce the notion of *t-approximated determinisation* and the *t-twinning property* of weighted tree automata (WTA) over the tropical semiring. We provide an algorithm that accomplishes t-approximated determinisation of an input automaton $\mathscr{A}$, whenever it terminates. Moreover, we prove that the t-twinning property of $\mathscr{A}$ is a sufficient condition for the termination of our algorithm. Ultimately, we show decidability of the t-twinning property for WTA.

Keywords: Weighted Automata · Approximation · Approximated Determinisation · Tree Automata · Twinning Property.

1 Introduction

In theoretical computer science, automata theory arose as a very potent field of research. Besides having manifold applications in areas like natural language processing, model checking, and computational biology, automata are studied in a vast number of syntactical variations. The most prominent case of finite string automata has been extended to handle more complex input structures like pictures, trees, and forests (cf. [16, 17]). Another direction of generalisation is to allow quantitative calculations rather than simple binary acceptance. Well-studied examples of such automata are weighted string automata and weighted tree automata over some weight structure S (cf. [9] for exhaustive references). Prominent weight structures include commutative semirings [1], and strong bimonoids [10].

One of the major research fields in automata theory is the determinisation of automata. While this problem has a well-known solution for unweighted automata, very little results are known in the weighted setting. In fact, not every

* Research of the first and third author was supported by the DFG through the Research Training Group QuantLA (GRK 1763). The second author was supported by the European Research Council (ERC) through the ERC Consolidator Grant No. 771779 (DeciGUT).

weighted automaton can be determinised [5, Example 5.9]. One endeavour to simplify automata that cannot be determinised is to aim for *approximated determinisation*. Different approaches to this paradigm have been proposed, see e.g. [2], [3], and [4]. The main idea of these papers is to take an automaton $\mathscr{A}$ and construct a deterministic automaton that recognizes a "similar" language to the one of $\mathscr{A}$. The notions of similarity differ in the literature. As the present paper aims to generalise [2] from the string case to the tree case, we subsequently focus on [2].

In [2], the weight structure is the tropical semiring $(\mathbb{R}_\infty, \min, +, \infty, 0)$. The notion of approximation is given as follows. Let $t \geq 1$ be a real number, called the *approximation factor*. A weighted automaton $\mathscr{A}'$ *t-approximates* $\mathscr{A}$, if for every input string $w \in \Sigma^*$ it holds that $[\![\mathscr{A}]\!](w) \leq [\![\mathscr{A}']\!](w) \leq t \cdot [\![\mathscr{A}]\!](w)$, where $[\![\mathscr{A}]\!]$ denotes the weighted language recognised by $\mathscr{A}$.

Aminof et al. [2] give an algorithm, called tDet, that takes as input a weighted string automaton $\mathscr{A}$ and an approximation factor t and (if the algorithm terminates) outputs a deterministic weighted string automaton $\mathscr{A}'$ such that $\mathscr{A}'$ t-approximates $\mathscr{A}$. The algorithm tDet executes a weighted powerset construction (with a fixed factorisation) similar to the one given by Kirsten and Mäurer [11]. That is, the states of $\mathscr{A}'$ are essentially subsets of the state set of $\mathscr{A}$, where each state of $\mathscr{A}$ gets assigned a residual weight. These residual weights keep track of the difference between the weights of runs of $\mathscr{A}'$ and runs of $\mathscr{A}$. For approximated determinisation however, tDet keeps track of two *bounds* for every state of $\mathscr{A}$ rather than a single residual weight. Namely, a *lower bound* and an *upper bound*. These bounds describe intervals of residual weights in order to ensure t-approximation.

Next, Aminof et al. [2] prove that tDet terminates if $\mathscr{A}$ satisfies the so-called t-twinning property. The t-twinning property is a generalisation of the classical twinning property (see [13]). Ultimately, it is proven in [2] that the t-twinning property is decidable.

The approach of the present paper closely follows the approach by Aminof et al. [2]. In Section 2 we introduce some elementary technical machinery and our automaton model. Next, in Section 3 we define t-approximation for weighted tree automata, give an algorithm for t-approximate determinisation, and prove its partial correctness. In Section 4 we introduce the t-twinning property for weighted tree automata and show that it is a sufficient condition for the termination of our algorithm. In Section 5 we prove that our t-twinning property is decidable and in Section 6 we conclude the paper by posing some open questions.

2 Preliminaries

We denote the set of *integers* by $\mathbb{Z}$, the set of *nonnegative integers* by $\mathbb{N}$, and the set of *positive integers* by $\mathbb{N}_+$. Moreover, we denote the set of *real numbers* by $\mathbb{R}$ and define the set $\mathbb{R}_\infty \coloneqq \{x \in \mathbb{R} \mid x \geq 0\} \cup \{\infty\}$. Analogously, we denote the set of *rational numbers* by $\mathbb{Q}$ and define the set $\mathbb{Q}_\infty \coloneqq \{x \in \mathbb{Q} \mid x \geq 0\} \cup \{\infty\}$. For every $x, y \in \mathbb{R}$, we define the *interval* $[x, y] \coloneqq \{z \in \mathbb{R} \mid x \leq z \leq y\}$ and denote the set $[\infty, \infty] \coloneqq \{\infty\}$. For every $k \in \mathbb{N}$, we denote the set $\{i \in \mathbb{N} \mid 1 \le i \le k\}$ by $[k]$. Note that $[0] = \emptyset$. For a set A we denote the *size* of A by $\#A$ and for every $k \in \mathbb{N}_+$ we denote by A^k the k-fold cartesian power of A.

An *alphabet* is a finite and non-empty set A and $A^* = \bigcup_{k \in \mathbb{N}} A^k$ is the set of all (finite) *words* over A, where $A^0 = \{\varepsilon\}$ contains solely the *empty word* ε. We denote by $|w|$ the *length* of the word $w \in A^*$. Given words $v, w \in A^*$, their concatenation is written $v.w$ or simply vw. We write $v \preceq w$ provided that there exists $u \in A^*$ such that $vu = w$. The relation $\preceq$ is in fact a partial order, called the *prefix order*.

A *ranked alphabet* is a pair (Σ, rk) consisting of an alphabet Σ and a mapping $\mathrm{rk}\colon \Sigma \to \mathbb{N}$ that assigns a *rank* to each symbol of Σ. We refer to the ranked alphabet (Σ, rk) by the set Σ whenever the map rk is clear from the context. Furthermore, for every $k \in \mathbb{N}$, we let $\Sigma^{(k)} = \{\sigma \in \Sigma \mid \mathrm{rk}(\sigma) = k\}$ and we write $\sigma^{(k)}$ to indicate that $\mathrm{rk}(\sigma) = k$.

> Throughout the rest of this paper, we assume Σ to be a ranked alphabet and $\Sigma^{(0)} \neq \emptyset$.

Given a set Z, the set of *Σ-trees* indexed by Z, denoted by $\mathrm{T}_\Sigma(Z)$, is the smallest set T such that $Z \subseteq \mathrm{T}$ and $\sigma(\xi_1, \dots, \xi_r) \in \mathrm{T}$ for every $r \in \mathbb{N}$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \dots, \xi_r \in \mathrm{T}$. We abbreviate $\mathrm{T}_\Sigma = \mathrm{T}_\Sigma(\emptyset)$ and call every subset $L \subseteq \mathrm{T}_\Sigma$ a *tree language*.

Next, we recall some common notions and notations for trees. In the following, let $\xi \in \mathrm{T}_\Sigma(Z)$. The set $\mathrm{pos}(\xi)$ of *positions* of ξ is defined inductively by $\mathrm{pos}(z) = \{\varepsilon\}$ for all $z \in Z$, and $\mathrm{pos}(\sigma(\xi_1, \dots, \xi_r)) = \{\varepsilon\} \cup \{i.w \mid i \in [r],\ w \in \mathrm{pos}(\xi_i)\}$ for every $r \in \mathbb{N}$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \dots, \xi_r \in T_\Sigma(Z)$. The *height* of ξ is defined by $\mathrm{height}(\xi) = \max_{w \in \mathrm{pos}(\xi)} |w|$, and the *size* of ξ is defined by $\mathrm{size}(\xi) = \#\mathrm{pos}(\xi)$. A *leaf* is a position $w \in \mathrm{pos}(\xi)$ such that $w.1 \notin \mathrm{pos}(\xi)$. We denote the set of leaves of ξ by $\mathrm{leaf}(\xi)$. Given a position $w \in \mathrm{pos}(\xi)$, the *label* of ξ at w is denoted by $\xi(w)$. The *subtree* of ξ at w, denoted $\xi|_w$, is defined for every $z \in Z$ by $z|_\varepsilon = z$ and for every $r \in \mathbb{N}$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \dots, \xi_r \in \mathrm{T}_\Sigma(Z)$ by

$$\sigma(\xi_1, \dots, \xi_r)|_w = \begin{cases} \sigma(\xi_1, \dots, \xi_r) & \text{if } w = \varepsilon \\ \xi_i|_{w'} & \text{if } w = i.w' \text{ with } i \in \mathbb{N} \text{ and } w' \in \mathrm{pos}(\xi_i). \end{cases}$$

Let Y be a set. The set of *positions of ξ labeled by elements in Y*, denoted by $\mathrm{pos}_Y(\xi)$, is the set $\{w \in \mathrm{pos}(\xi) \mid \xi(w) \in Y\}$. Moreover, the *replacement* of the leaf $w \in \mathrm{leaf}(\xi)$ by the tree $\eta \in T_\Sigma(Z)$, denoted $\xi[\eta]_w$, is given for every $z \in Z$ by $z[\eta]_\varepsilon = \eta$ and for every $r \in \mathbb{N}$, $i \in [r]$, $\sigma \in \Sigma^{(r)}$, $\xi_1, \dots, \xi_r \in \mathrm{T}_\Sigma(Z)$, and $w' \in \mathrm{pos}(\xi_i)$ by $\sigma(\xi_1, \dots, \xi_r)[\eta]_{i.w'} = \sigma(\xi_1, \dots, \xi_{i-1}, \xi_i[\eta]_{w'}, \xi_{i+1}, \dots, \xi_r)$.

The set $\mathrm{path}(\xi) \subseteq (\Sigma \cup Z)^*$ of *paths* of ξ is defined inductively by $\mathrm{path}(z) = \{z\}$ for all $z \in Z$, and $\mathrm{path}(\sigma(\xi_1, \dots, \xi_r)) = \{\sigma w \mid i \in [r],\ w \in \mathrm{path}(\xi_i)\}$ for every $r \in \mathbb{N}$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \dots, \xi_r \in T_\Sigma(Z)$.

We fix the set $X = \{x_1, x_2, \dots\}$ of *variables* (which we impose to be disjoint from any other set we consider), and $X_n = \{x_1, \dots, x_n\}$ for every $n \in \mathbb{N}_+$. A tree $\xi \in \mathrm{T}_\Sigma(X_1)$ is a *context*, if $\#\mathrm{pos}_{x_1}(\xi) = 1$. The set of all contexts is denoted by C_Σ.

Given a context $\zeta \in \mathrm{C}_\Sigma$ and a tree $\xi \in \mathrm{T}_\Sigma(Z)$, the *substitution* of ξ into ζ, denoted by $\zeta[\xi]$, is the tree $\zeta[\xi]_w$, where w is the unique position in $\mathrm{pos}_X(\zeta)$. Note that, given $\zeta, \zeta' \in \mathrm{C}_\Sigma$, also $\zeta[\zeta'] \in \mathrm{C}_\Sigma$. We write ζ^k for $\zeta[\zeta[\cdots\zeta[\zeta]\cdots]]$ containing the context ζ a total of k times.

We recall the *tropical semiring* $(\mathbb{R}_\infty, \min, +, \infty, 0)$, where min and + are binary operations on $\mathbb{R}_\infty$ and are the natural extensions of the respective real-valued operations.

Definition 1 (WTA). A *weighted tree automaton* (short: WTA) is a tuple $(Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$, where Q is an alphabet of *states*, $\mathrm{final}\colon Q \to \mathbb{R}_\infty$ is a map of *final weights*, and T is a family $(T_\sigma\colon Q^r \times Q \to \mathbb{R}_\infty \mid r \geq 0, \sigma \in \Sigma^{(r)})$ of maps of *transition weights*.

We call a tuple $t = (q_1, \ldots, q_r, \sigma, x, q) \in Q^r \times \Sigma \times \mathbb{R}_\infty \times Q$ a *transition* if $\mathrm{rk}(\sigma) = r$ and $T_\sigma(q_1, \ldots, q_r, q) = x$. We sometimes denote t by $\sigma(q_1, \ldots, q_r) \xrightarrow{x} q$.

Definition 2 (run). Let $\mathscr{A} = (Q, \Sigma, S, \mathrm{final}, T)$ be a WTA and $\xi \in \mathrm{T}_\Sigma \cup \mathrm{C}_\Sigma$ be a tree or a context. A *run* of $\mathscr{A}$ on ξ is a map $\rho\colon \mathrm{pos}(\xi) \to Q$.

Let $w \in \mathrm{pos}(\xi)$. The *weight of ρ at position* w, denoted $\mathrm{wt}(\rho, w)$, is an element of $\mathbb{R}_\infty$ defined inductively as follows. If $\mathrm{label}(\xi, w) \in X$, then we define $\mathrm{wt}(\rho, w) \coloneqq 0$ and if $\mathrm{label}(\xi, w) = \sigma$ is in $\Sigma^{(r)}$, then we define $\mathrm{wt}(\rho, w) \coloneqq \mathrm{wt}(\rho, w1) + \cdots + \mathrm{wt}(\rho, wr) + T_\sigma(\rho(w1), \ldots, \rho(wr), \rho(w))$. Furthermore, the *weight* of ρ, denoted $\mathrm{wt}(\rho)$, is defined by $\mathrm{wt}(\rho) \coloneqq \mathrm{wt}(\rho, \varepsilon)$.

We say that ρ *contains* a state $q \in Q$ if there exists $w \in \mathrm{pos}(\xi)$ such that $q = \rho(w)$. We say that ρ is *non-vanishing* if $\mathrm{wt}(\rho) \neq \infty$.

Remark 3. We use the following notation for a run ρ of $\mathscr{A}$ on a tree or context ξ. Let $q \coloneqq \rho(\varepsilon)$ and $x \coloneqq \mathrm{wt}(\rho)$. If $\xi \in T_\Sigma$, then we write $\xrightarrow{\xi|\rho|x} q$. If $\xi \in C_\Sigma$, then we write $p \xrightarrow{\xi|\rho|x} q$, where $p \coloneqq \rho(w)$ for the unique $w \in \mathrm{pos}_X(\xi)$. Whenever we do not care about the name of the run, we simply write $\xrightarrow{\xi|x} q$ and $p \xrightarrow{\xi|x} q$, respectively. Furthermore, if $\xrightarrow{\xi|x} q$ for some tree ξ and $x \neq \infty$, then we call the state q *reachable*.

Definition 4 (sets of runs). Let $\mathscr{A} = (Q, \Sigma, S, \mathrm{final}, T)$ be a WTA and $\xi \in \mathrm{T}_\Sigma \cup \mathrm{C}_\Sigma$ be a tree or a context. The set of runs of $\mathscr{A}$ on ξ is denoted by $\mathrm{Run}_\mathscr{A}(\xi)$. For every $q \in Q$ and $\xi \in \mathrm{T}_\Sigma$ we define the set $\mathrm{Run}_\mathscr{A}(\xi, q) \coloneqq \{\rho \in \mathrm{Run}_\mathscr{A}(\xi) \mid \xrightarrow{\xi|\rho|\mathrm{wt}(\rho)} q\}$ and the *run weight* of ξ into q as $\theta_\mathscr{A}(\xi, q) \coloneqq \min\{\mathrm{wt}(\rho) \mid \rho \in \mathrm{Run}_\mathscr{A}(\xi, q)\}$. Analogously, for every $p, q \in Q$ and $\xi \in \mathrm{C}_\Sigma$ we define the set $\mathrm{Run}_\mathscr{A}(p, \xi, q) \coloneqq \{\rho \in \mathrm{Run}_\mathscr{A}(\xi) \mid p \xrightarrow{\xi|\rho|\mathrm{wt}(\rho)} q\}$ and the *run weight* of ξ from p into q as $\theta_\mathscr{A}(p, \xi, q) \coloneqq \min\{\mathrm{wt}(\rho) \mid \rho \in \mathrm{Run}_\mathscr{A}(p, \xi, q)\}$.

Definition 5 (semantics of WTA). Let $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ be a WTA. The *weighted tree language accepted by* $\mathscr{A}$ is the map $[\![\mathscr{A}]\!]\colon \mathrm{T}_\Sigma \to \mathbb{R}_\infty$, where for every $\xi \in \mathrm{T}_\Sigma$ we define

$$[\![\mathscr{A}]\!](\xi) \coloneqq \min_{q \in Q} \big(\theta_{\mathscr{A}}(\xi, q) + \mathrm{final}(q)\big).$$

Two WTA $\mathscr{A}$ and $\mathscr{B}$ are called *equivalent* if they accept the same weighted tree language, that is, if $[\![\mathscr{A}]\!] = [\![\mathscr{B}]\!]$.

Note that our weighted tree automata are classical semiring-weighted tree automata (cf. [9, Chapter 9]) where we fix the semiring $S = \mathbb{R}_\infty$.

Definition 6 (deterministic). Let $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ be a WTA. We call $\mathscr{A}$ *deterministic* if for all $r \geq 0, \sigma \in \Sigma^{(r)}$, and $q_1, \ldots, q_r \in Q$ there exist at most one $q \in Q$ such that $T_\sigma(q_1, \ldots, q_r, q) \neq \infty$. Moreover, we call $\mathscr{A}$ *unambiguous* if for every $\xi \in \mathrm{T}_\Sigma$ there exists at most one non-vanishing run of $\mathscr{A}$ on ξ. If $\mathscr{A}$ is unambiguous, then we define for every $\xi \in \mathrm{T}_\Sigma$ the value $\theta_{\mathscr{A}}(\xi) \coloneqq \mathrm{wt}(\rho)$ as the weight of the unique non-vanishing run ρ of $\mathscr{A}$ on ξ (if such a run exists, and as ∞ otherwise).

A map $f \colon T_\Sigma \to \mathbb{R}_\infty$ is called *deterministically recognizable* if there exists a deterministic WTA $\mathscr{A}$ such that $[\![\mathscr{A}]\!] = f$.

Example 7. Let $\Sigma = \{\alpha^{(0)}, \beta^{(0)}, \sigma^{(2)}\}$ and consider $\mathscr{A} \coloneqq (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ where $Q \coloneqq \{q_1, q_2\}$, final $\coloneqq 0$, and T is ∞ except in the cases

$$\alpha \xrightarrow{1} q_1, \qquad \alpha \xrightarrow{2} q_2, \qquad \sigma(q_1, q_1) \xrightarrow{0} q_1,$$
$$\beta \xrightarrow{0} q_1, \qquad \sigma(q_2, q_2) \xrightarrow{0} q_2.$$

We depict WTA by hypergraphs (see Figures 1 and 2) which are read in the following way. Each state of the WTA is represented by a circle labeled by the name of the state. A transition of the form $\sigma(q_1, \ldots, q_r) \xrightarrow{x} q$ with $x \neq \infty$ is represented by a box labeled by σ and having r incoming edges and a single outgoing edge. The outgoing edge includes the weight of the transition, x, and is indicated by an arrow. The incoming edges are ordered by counter-clockwise traversal starting to the left of the outgoing edge. A depiction of the automaton $\mathscr{A}$ can be found in Figure 1.

Let $\xi \in \mathrm{T}_\Sigma$. One easily verifies the following statements using the definition of $\mathscr{A}$. If ξ contains at least one β, then there exists a unique non-vanishing run ρ of $\mathscr{A}$ on ξ and it holds that $\mathrm{wt}(\rho) = \#\mathrm{pos}_\alpha(\xi)$. If ξ contains no β, then there exist exactly two non-vanishing runs ρ_1 and ρ_2 of $\mathscr{A}$ on ξ and it holds that $\mathrm{wt}(\rho_1) = \#\mathrm{pos}_\alpha(\xi)$ and $\mathrm{wt}(\rho_2) = 2 \cdot \#\mathrm{pos}_\alpha(\xi)$. In total, we obtain that $[\![\mathscr{A}]\!](\xi) = \#\mathrm{pos}_\alpha(\xi)$.

Clearly, $\mathscr{A}$ is not deterministic, as T contains the transitions $\alpha \xrightarrow{1} q_1$ and $\alpha \xrightarrow{2} q_2$. Furthermore $\mathscr{A}$ is not unambiguous, as there exist two non-vanishing runs of $\mathscr{A}$ on α.

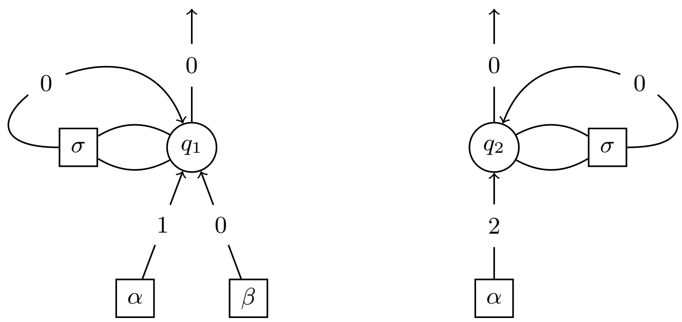

Fig. 1. Non-deterministic WTA $\mathscr{A}$ from Example 7.

3 Approximated Determinisation

In this section we present an algorithm that takes a weighted tree automaton $\mathscr{A}$ as input and generates a tuple $\mathscr{A}'$. Under certain conditions, this tuple is a deterministic weighted tree automaton that approximates $\mathscr{A}$. After applying the algorithm to the automaton from Example 7, we show the partial correctness of the algorithm. That is, if the algorithm terminates, the tuple $\mathscr{A}'$ is in fact a deterministic weighted tree automaton that approximates $\mathscr{A}$. Our approach closely follows [2] and we start by defining approximation of weighted tree automata.

Throughout the rest of this section, we assume $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ to be an arbitrary WTA.

Definition 8 (t-approximation [2]). Let $t \in \mathbb{R}$ be a real number such that $t \geq 1$ and let $\mathscr{B} = (Q', \Sigma, \mathbb{R}_\infty, \mathrm{final}', T')$ be a WTA.

We say that $\mathscr{B}$ *t-approximates* $\mathscr{A}$ if for every $\xi \in \mathrm{T}_\Sigma$ it holds that

$$[\![\mathscr{A}]\!](\xi) \leq [\![\mathscr{B}]\!](\xi) \leq t \cdot [\![\mathscr{A}]\!](\xi).$$

Moreover, we call $\mathscr{A}$ *t-approximate deterministic* (or *t-determinisable*) if there exists a deterministic WTA $\mathscr{B}$ such that $\mathscr{B}$ t-approximates $\mathscr{A}$.

Remark 9. Note that if $\mathscr{B}$ t-approximates $\mathscr{A}$, then $\mathrm{supp}([\![\mathscr{A}]\!]) = \mathrm{supp}([\![\mathscr{B}]\!])$. Moreover, $\mathscr{B}$ 1-approximates $\mathscr{A}$ if and only if $[\![\mathscr{A}]\!] = [\![\mathscr{B}]\!]$.

Throughout the rest of this section, we assume that $t \in \mathbb{R}$ such that $t \geq 1$.

Remark 10. Note that, in general, $\mathscr{A}$ is not t-determinisable. In fact, if Σ contains two distinct symbols $\sigma^{(r)}$ and $\tau^{(s)}$ (where $r, s > 0$), then there exists a WTA $\mathscr{B}$ such that $\mathscr{B}$ is not t'-determinisable for any $t' \geq 1$.

In [2, Theorem 1], this is proven for strings and the constructions can easily be adapted to the tree case by considering comb trees over σ and τ, which behave similarly to strings.

Next we introduce our approximate determinisation algorithm. For a summary of the conceptional details of our approach and how it fits into the existing literature, we refer to Section 1. Recall that our algorithm executes a weighted powerset construction with a fixed factorisation (see [11]). In this intermediate text, we present the intuitive idea of our algorithm and the relevant technicalities.

Given the automaton $\mathscr{A}$ and the approximation factor t, the algorithm builds up a deterministic automaton $\mathscr{A}' = (Q', \Sigma, \mathbb{R}_\infty, \mathrm{final}', T')$ by iteratively adding new states and transitions to $\mathscr{A}'$ (which is initially empty). The states of $\mathscr{A}'$ are subsets of $(\mathbb{R}_\infty \times \mathbb{R}_\infty)^Q$, which we think about as follows. Each state $P \in Q'$ maps every state $q \in Q$ to a *lower bound* l_q^P and an *upper bound* u_q^P. Thus, we denote $(l_q^P, u_q^P) := P(q)$. These bounds represent an interval in $\mathbb{R}_\infty$ and will be determined by the algorithm such that the following holds.

Let ρ be the (unique) non-vanishing run of $\mathscr{A}'$ on a tree ξ and let $\rho(\varepsilon) = P$. For every $q \in Q$ it holds that the interval $[\theta_{\mathscr{A}}(\xi, q), t \cdot \theta_{\mathscr{A}}(\xi, q)]$ contains the interval $[l_q^P + \mathrm{wt}(\rho), u_q^P + \mathrm{wt}(\rho)]$ (see Lemma 14). Note that $[\theta_{\mathscr{A}}(\xi, q), t \cdot \theta_{\mathscr{A}}(\xi, q)]$ is the interval which t-approximates $\theta_{\mathscr{A}}(\xi, q)$. Therefore, $\mathscr{A}'$ t-approximates $\mathscr{A}$ as long as the final weight map of $\mathscr{A}'$ respects the lower and upper bounds.

Moreover, we use of the following concept. Given two states $P, P' \colon Q \to \mathbb{R}_\infty \times \mathbb{R}_\infty$, we say that P' *refines* P if for every $q \in Q$ it holds that $[l_q^{P'}, u_q^{P'}] \subseteq [l_q^P, u_q^P]$. Refinement plays a major role in ensuring the termination of Algorithm 1.

The overall structure of Algorithm 1 is the following. We initialise $\mathscr{A}'$ as an empty WTA (line 1). Next, we iteratively generate non-vanishing transitions for $\mathscr{A}'$, which in some cases add new states to the state set of $\mathscr{A}'$. The family of sets $(\mathrm{Stack}(\sigma) \mid \sigma \in \Sigma)$ is used to keep track of transitions that have already been processed. Given a left-hand side $\sigma(P_1, \ldots, P_r)$ of a transition that has not been processed (lines 4 and 5), we calculate an intermediate successor state P by accumulating the lower bounds and the upper bounds respectively with the transition weights (lines 7 – 9). Next, we determine the new transition weight c' as the minimal resulting *upper* bound in P (line 8). If c' is not ∞, then we define P' as $P - c'$ (lines 11 and 12). We check if P' is refined by some already existing state P'' (line 13). If this is the case, we add a *red* transition to $\mathscr{A}'$ by letting $T'_\sigma(P_1, \ldots, P_r, P'') = c'$ (line 14). Otherwise, we add P' as a new state and add a *green* transition to $\mathscr{A}'$ by letting $T'_\sigma(P_1, \ldots, P_r, P') = c'$ (lines 16 and 18). We ultimately define the new final weights (line 17).

We distinguish between red and green transitions for the following reason. A transition $t = (P_1, \ldots, P_r, \sigma, c, P)$ is green if and only if it was the first non-vanishing transition with successor state P which was generated by Algorithm 1. Otherwise, t is either vanishing or a red transition. This defines a green subgraph of $\mathscr{A}'$ (viewed as a hypergraph). The proofs of our main theorems rely on the green subgraph of $\mathscr{A}'$ in order to use induction over the set of states of $\mathscr{A}'$.

Note that we define states of $\mathscr{A}'$ using a relational notation (see lines 7 and 12) rather than a functional notation, for better readability. Moreover, note that line 3 is merely a technical requirement that forces the second execution of the outermost while-loop (line 2) to happen immediately after each symbol from $\Sigma^{(0)}$ has been processed.

Algorithm 1: Procedure ttDet with input $\mathscr{A}$ and t

1 $Q' := \emptyset$, $\text{final}' := \infty$, $(\text{Stack}(\sigma) := \emptyset \mid \sigma \in \Sigma)$, $T' := (T'_\sigma \mid \sigma \in \Sigma)$ where $T'_\sigma := \infty$
2 **while** $\exists \sigma \in \Sigma : (Q')^{\text{rk}(\sigma)} \setminus \text{Stack}(\sigma) \neq \emptyset$ **do**
3     $Q'' := Q'$
4   **foreach** $r \in \mathbb{N}, \sigma \in \Sigma^{(r)}$ **do**
5   **foreach** $((P_1, \dots, P_r) \in (Q'')^r \setminus \text{Stack}(\sigma))$ **do**
6             $\text{Stack}(\sigma) := \text{Stack}(\sigma) \cup \{(P_1, \dots, P_r)\}$
7             $P := \{(q, (l_q, u_q)) \mid q \in Q\}$ where
8                 $l_q := \min\{l_{q_1}^{P_1} + \dots + l_{q_r}^{P_r} + T_\sigma(q_1, \dots, q_r, q) \mid q_1, \dots, q_r \in Q\}$
9                 $u_q := \min\{u_{q_1}^{P_1} + \dots + u_{q_r}^{P_r} + t \cdot T_\sigma(q_1, \dots, q_r, q) \mid q_1, \dots, q_r \in Q\}$
10             $c' := \min_{q \in Q} u_q^P$
11   **if** $c' < \infty$ **then**
12                 $P' := \{(q, (l_q^P - c', u_q^P - c')) \mid q \in Q\}$
13   **if** $\exists P'' \in Q'$ such that P'' refines P' **then**
14                     $T'_\sigma(P_1, \dots, P_r, P'') := c'$ `// red transition`
15   **else**
16                     $Q' := Q' \cup \{P'\}$
17                     $\text{final}'(P') := \min_{q \in Q}(u_q^{P'} + t \cdot \text{final}(q))$
18                     $T'_\sigma(P_1, \dots, P_r, P') := c'$ `// green transition`
19 **return** $(Q', \Sigma, \mathbb{R}_\infty, \text{final}', T')$

Definition 11. We define $\mathscr{A}'$ as the tuple returned[4] by ttDet applied to $\mathscr{A}$ and t and denote its components by $\mathscr{A}' = (Q', \Sigma, \mathbb{R}_\infty, \text{final}', T')$.

Example 12. We continue Example 7 by applying ttDet to $\mathscr{A}$ and t for $t \geq 2$.

First consider $\alpha \in \Sigma^{(0)}$. Via lines 7 – 10 we calculate

$$P = \{(q_1, (1, t)), (q_2, (2, 2t))\} \qquad \text{and} \qquad c' = t.$$

By line 12 we obtain $P' = \{(q_1, (1 - t, 0)), (q_2, (2 - t, t))\}$. As Q' is still empty, P' is not refined by some other state and we enter the else-case (lines 16 – 18). We denote $P'_1 := P'$ and execute lines 16, 17, and 18 to update

$$Q' = \{P'_1\}, \qquad \text{final}'(P'_1) = 0, \qquad \text{and} \qquad T'_\alpha(P'_1) = t.$$

Note that this transition is a green transition.

Next consider $\beta \in \Sigma^{(0)}$. We calculate $P = \{(q_1, (0, 0)), (q_2, (\infty, \infty))\}$ (lines 7 – 9) and $c' = 0$ (line 10). By line 12 we obtain $P' = P$. As P'_1 does not refine

[4] We denote by $\mathscr{A}'_i$ the tuple $(Q', \Sigma, \mathbb{R}_\infty, \text{final}', T')$ during the i-th execution of line 2. If ttDet does not terminate on the input $\mathscr{A}$ and t, the limit of these tuples for $i \to \infty$ is their componentwise union and we say that ttDet returns this limit.

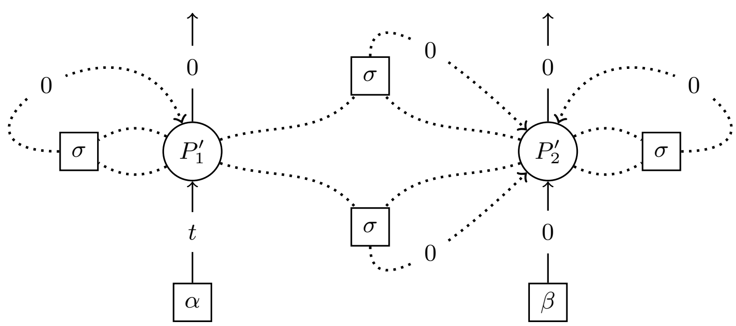

Fig. 2. Deterministic WTA $\mathscr{A}'$ t-approximating the WTA $\mathscr{A}$ from Example 7.

P', we again enter the else-case (lines 16 – 18). We denote $P'_2 \coloneqq P'$ and execute lines 16, 17, and 18 to update

$$Q' = \{P'_1, P'_2\}, \qquad \text{final}'(P'_2) = 0, \qquad \text{and} \qquad T'_\beta(P'_2) = 0.$$

Note that this transition is a green transition.

Next consider $\sigma \in \Sigma^{(2)}$. As Stack(σ) is still empty, we consider $(P'_1, P'_1) \in (Q')^2 \setminus \text{Stack}(\sigma)$ (line 5). By lines 7 – 12 we obtain $P = \{(q_1, (2-2t, 0)), (q_2, (4-2t, 2t))\}$, $c' = 0$, and $P' = P$. Note that P'_2 does not refine P' and that P'_1 refines P' if and only if

$$2-2t \leq 1-t, \qquad 0 \leq 0, \qquad 4-2t \leq 2-t, \qquad \text{and} \qquad t \leq 2t.$$

Therefore, P' is refined by P'_1 if and only if $t \geq 2$, which is true by assumption. Hence, we enter the if-case (line 14) and add the red transition $T'_\sigma(P'_1, P'_1, P'_1) = 0$ to T'.

By continuing to execute the algorithm, we add more red transitions to T'_σ and arrive at the automaton $\mathscr{A}' = (Q', \Sigma, \mathbb{R}_\infty, \text{final}', T')$, where $Q' = \{P'_1, P'_2\}$, $\text{final}' = 0$, and T' is ∞ except in the cases

$$\alpha \xrightarrow{t} P'_1, \qquad \sigma(P'_1, P'_1) \xrightarrow{0} P'_1, \qquad \sigma(P'_1, P'_2) \xrightarrow{0} P'_2,$$
$$\beta \xrightarrow{0} P'_2, \qquad \sigma(P'_2, P'_1) \xrightarrow{0} P'_2, \qquad \sigma(P'_2, P'_2) \xrightarrow{0} P'_2.$$

A depiction of $\mathscr{A}'$ can be found in Figure 2. Note that green transitions are depicted by continuous lines and red transitions are depicted by dotted lines.

Remark 13. Note that ttDet does not preserve the weighted language of $\mathscr{A}$, even if $\mathscr{A}$ is itself deterministic. This is due to the fact that the state normalisation (lines 10 and 12) is done with respect to the *upper* bounds u_q^P. Therefore, $\mathscr{A}'$ realises $t \cdot [\![\mathscr{A}]\!]$ rather than $[\![\mathscr{A}]\!]$ in many basic examples.

A straightforward induction on the depth of states in the green subgraph of $\mathscr{A}'$ shows that if ttDet terminates, then $\mathscr{A}'$ is a deterministic WTA. Similarly, the following lemma can be proven.

Lemma 14. Let $\xi \in \mathrm{T}_\Sigma$ and $P \in Q'$ such that $\xrightarrow{\xi|\theta_{\mathscr{A}'}(\xi)} P$. For every $q \in Q$ it holds that

$$\theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi) \leq l_q^P \leq u_q^P \leq t \cdot \theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi).$$

The following theorem states the partial correctness of Algorithm 1 and follows from Lemma 14 and the definition of final$'$.

Theorem 15. If ttDet terminates on input $\mathscr{A}$ and t, then $\mathscr{A}'$ is a deterministic WTA that t-approximates $\mathscr{A}$. In this case, $\mathscr{A}$ is in particular t-determinisable.

4 Approximated Twinning Property

In this section, we prove a sufficient condition for the termination of the algorithm, namely the t-twinning property. Our proof closely follows [2]. We start by defining the t-twinning property of weighted tree automata, which is a natural extension of both, the string case [2] and the tree case without approximation (that is, $t = 1$) [7] and [14].

Definition 16 (t-twinning property). Let $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ be a WTA.

Let $p, q \in Q$. We call p and q *siblings* if there exists a tree $\xi \in \mathrm{T}_\Sigma$ and non-vanishing runs $\rho_1 \in \mathrm{Run}_{\mathscr{A}}(\xi, p)$ and $\rho_2 \in \mathrm{Run}_{\mathscr{A}}(\xi, q)$. Siblings p and q are called *t-twins* if for every $\zeta \in \mathrm{C}_\Sigma$ it holds that either $\theta(p, \zeta, p) = \infty$, $\theta(q, \zeta, q) = \infty$, or $\frac{1}{t} \cdot \theta(q, \zeta, q) \leq \theta(p, \zeta, p) \leq t \cdot \theta(q, \zeta, q)$.

We say that $\mathscr{A}$ has the *t-twinning property* if for all siblings $p, q \in Q$ it holds that p and q are t-twins.

> Throughout the rest of this section, we assume $\mathscr{A} = (Q, \Sigma, \mathbb{Q}_\infty, \mathrm{final}, T)$ to be a WTA with rational weights in $\mathbb{Q}_\infty$ and $t \in \mathbb{R}$ such that $t \geq 1$.

Theorem 17. If $\mathscr{A}$ satisfies the t-twinning property, ttDet terminates on input $\mathscr{A}$ and t.

The proof of Theorem 17 is very similar to the proof of [2, Theorem 8]. The main difference is that, in the tree case, we apply a version of König's Lemma that can handle the hypergraph structure of $\mathscr{A}'$. Note that in [2, Theorem 8], t is a rational number, whereas we allow for t to be a real number. This can be resolved by multiplying t and all weights occurring in $\mathscr{A}$ by $\frac{1}{t}$.

Corollary 18. If $\mathscr{A}$ satisfies the t-twinning property, $\mathscr{A}$ is t-determinisable.

Proof of Corollary 18. This follows immediately from Theorems 15 and 17. □

Example 19. We continue Example 7 by showing that $\mathscr{A}$ satisfies the 2-twinning property but not the 1-twinning property. First note that q_1 and q_2 are siblings as there are two runs ρ_1 and ρ_2 on $\xi = \alpha$ ending in q_1 and q_2, respectively.

Let $\zeta \in \mathrm{C}_\Sigma$ and ρ be a non-vanishing run of $\mathscr{A}$ on ζ. If ζ contains a β, we have that $\theta(q_2, \zeta, q_2) = \infty$ and hence we only have to check the 2-twinning property for the case that ζ does not contain a β. One easily sees that ρ either maps each position to q_1 (in this case $\mathrm{wt}(\rho) = \#\mathrm{pos}_\alpha(\zeta)$) or to q_2 (in this case $\mathrm{wt}(\rho) = 2 \cdot \#\mathrm{pos}_\alpha(\zeta)$). In particular, $\theta(q_2, \zeta, q_2) = 2 \cdot \theta(q_1, \zeta, q_1)$. This proves that $\mathscr{A}$ satisfies the 2-twinning property.

Moreover, $\mathscr{A}$ does not satisfy the 1-twinning property, as q_1 and q_2 are siblings but $\zeta = \sigma(\alpha, x_1)$ does not satisfy $\theta(q_1, \zeta, q_1) = \theta(q_2, \zeta, q_2)$.

Note that ttDet does not terminate on input $\mathscr{A}$ and 1. In Example 12, we generated the state $P' = \{(q_1, (2 - 2t, 0)), (q_2, (4 - 2t, 2t))\}$ by considering the input $\sigma(P'_1, P'_1)$. If $t = 2$, P' is refined by P'_1. For $t = 1$, however, $P' = \{(q_1, (0, 0)), (q_2, (2, 2))\}$ is not refined and therefore added to the state space. Next, considering $\sigma(P', P')$, we obtain another unrefineable state, namely $P'' = \{(q_1, (0, 0)), (q_2, (4, 4))\}$. One easily sees that the construction continues to generate every state of the form $\{(q_1, (0, 0)), (q_2, (2^k, 2^k))\}$ and hence ttDet does not terminate on input $\mathscr{A}$ and 1.

5 Decidability of the Twinning Property

In the following theorem we prove the decidability of the $t-$twinning property. This is due to the fact, that if a WTA $\mathscr{A}$ does not satisfy the t-twinning property, then this non-satisfaction is already witnessed by a small context tree.

Theorem 20. The t-twinning property is decidable for every WTA $\mathscr{A}$ and $t \geq 1$.

6 Outlook

In this paper we generalised [2] from the string case to the tree case. First we provided an algorithm for t-determinisation and proved its correctness, assuming termination of the algorithm. Next, we introduced the t-twinning property for trees and showed that, for WTA with weights in $\mathbb{Q}_\infty$, the t-twinning property implies the termination of our algorithm. We ultimately showed that our t-twinning property is decidable.

We conclude this paper by listing future research directions. Recent work has shown that the twinning property is equivalent to determinisability in some cases (e.g. [8]). It would be worthwhile to determine whether in our case the t-twinning property is necessary for t-determinisability. Another interesting research direction is to introduce approximated determinisation for general classes of semirings rather than only considering the tropical semiring. Moreover, it seems rather arbitrary to say $x \in \mathbb{R}$ is approximated exactly by the values in the interval $[x, t \cdot x]$. It would be interesting to introduce more general notions of "approximation" and find sufficient conditions for this general approximated determinisability.

References

1. Alexandrakis, A., Bozapalidis, S.: Weighted grammars and Kleene's theorem. Information Processing Letters **24**(1), 1–4 (1987)
2. Aminof, B., Kupferman, O., Lampert, R.: Rigorous approximated determinization of weighted automata. Theoretical Computer Science **480**, 104–117 (2013)
3. Boker, U., Henzinger, T.: Exact and Approximate Determinization of Discounted-Sum Automata. Logical Methods in Computer Science **10**(1) (2014). https://doi.org/10.2168/LMCS-10(1:10)2014
4. Boker, U., Henzinger, T.A.: Approximate determinization of quantitative automata. In: IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2012)
5. Borchardt, B.: A Pumping Lemma and Decidability Problems for Recognizable Tree Series. Acta Cybernetica **16**(4), 509–544 (Sep 2004)
6. Borchardt, B.: The Theory of Recognizable Tree Series. Verlag für Wissenschaft und Forschung, Berlin (2005), (PhD thesis, TU Dresden, Germany, 2004)
7. Büchse, M., May, J., Vogler, H.: Determinization of Weighted Tree Automata Using Factorizations. Journal of Automata, Languages and Combinatorics **15**(3/4), 229–254 (2010)
8. Daviaud, L., Jecker, I., Reynier, P., Villevalois, D.: Degree of Sequentiality of Weighted Automata. In: FOSSACS 2017. LNCS, vol. 10203, pp. 215–230 (2017). https://doi.org/10.1007/978-3-662-54458-7_13
9. Droste, M., Kuich, W., Vogler, H. (eds.): Handbook of Weighted Automata. EATCS Monographs in Theoretical Computer Science, Springer-Verlag (2009)
10. Droste, M., Stüber, T., Vogler, H.: Weighted finite automata over strong bimonoids. Information Sciences **180**, 156–166 (2010)
11. Kirsten, D., Mäurer, I.: On the Determinization of Weighted Automata. Journal of Automata, Languages and Combinatorics **10**, 287–312 (01 2005)
12. König, D.: Über eine Schlussweise aus dem Endlichen ins Unendliche. Acta Scientiarum Mathematicarum (Szeged) **3**(2-3), 121–130 (1927)
13. Mohri, M.: Finite-State Transducers in Language and Speech Processing. Computational Linguistics **23**(2), 269–311 (Jun 1997)
14. Paul, E.: Finite sequentiality of unambiguous max-plus tree automata. In: 36th International Symposium on Theoretical Aspects of Computer Science (STACS 2019). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2019)
15. Radovanovic, D.: Weighted tree automata over strong bimonoids. Novi Sad Journal of Mathematics **40**(3), 89–108 (2010)
16. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer-Verlag, Berlin, Heidelberg (1997)
17. Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 3: Beyond Words. Springer-Verlag, Berlin, Heidelberg (1997)

A Appendix

The following auxiliary lemma is a simple result that can be proven in a straightforward manner by induction on ξ using distributivity (cf. Equation 7 of the proof of [15, Theorem 4.1.] and [6, Lemma 4.1.13]).

Lemma 21. Let $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ be a WTA, $\xi \in \mathrm{T}_\Sigma$, and $q \in Q$. Moreover, let $r \geq 0$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \ldots, \xi_r \in \mathrm{T}_\Sigma$ such that $\xi = \sigma(\xi_1, \ldots, \xi_r)$. It holds that

$$\theta_{\mathscr{A}}(\xi, q) = \min\{\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i)\Big) + T_\sigma(q_1, \ldots, q_r, q) \mid q_1, \ldots, q_r \in Q\}.$$

The remainder of this appendix supplies proofs to Section 3. We recall that $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ is a WTA, $t \geq 1$ is an approximation factor, and $\mathscr{A}' = (Q', \Sigma, \mathbb{R}_\infty, \mathrm{final}', T')$ is the tuple returned by ttDet applied to $\mathscr{A}$ and t.

Definition 22. For every $P \in Q'$ we define the *green-depth* of P, denoted $\mathrm{gdepth}(P)$, as the minimal height of a tree $\xi \in \mathrm{T}_\Sigma$ such that there exists a green run $\xrightarrow{\xi \mid \theta_{\mathscr{A}'}(\xi)} P$ if such a tree ξ exists and as ∞ otherwise.

Lemma 23. Let $P \in Q'$.

(a) $\mathrm{gdepth}(P)$ is finite. In particular, there exists a green run ρ of $\mathscr{A}$ such that $\rho(\varepsilon) = P$.
(b) For every green transition of the form $T'_\sigma(P_1, \ldots, P_r, P)$ and every $i \in [r]$ it holds that $\mathrm{gdepth}(P_i) < \mathrm{gdepth}(P)$.

Proof. By definition of $\mathscr{A}'$ there exists an $i \in \mathbb{N}_+$ such that P is an element of the state set of $\mathscr{A}'_i$ (recall that $\mathscr{A}'_i$ denotes the tuple $(Q', \Sigma, \mathbb{R}_\infty, \mathrm{final}', T')$ during the i-th execution of line 2). We show that $\mathrm{gdepth}(P)$ is finite by induction on i.

Every state P' of $\mathscr{A}'_1$ has been generated by some $\alpha \in \Sigma^{(0)}$ and hence the claim follows for $i = 1$. Assume that the claim holds for all states of $\mathscr{A}'_i$ for some $i \in \mathbb{N}_+$. If P is a state of $\mathscr{A}'_{i+1}$ but not a state of $\mathscr{A}'_i$, there is a green transition of the form $T'_\sigma(P_1, \ldots, P_r, P)$ in $\mathscr{A}'_{i+1}$ such that $P_1, \ldots, P_r$ are states of $\mathscr{A}'_i$. Therefore, we obtain $\mathrm{gdepth}(P) = 1 + \max\{\mathrm{gdepth}(P_i) \mid i \in [r]\}$. This immediately implies claim (b) and moreover implies claim (a) by the induction assumption. □

In the following we prove the partial correctness of the algorithm (Theorem 15). The main tool is the fact that all calculated lower and upper bounds within states of $\mathscr{A}'$ describe ranges of real numbers respecting the desired t-approximation of $\mathscr{A}$ (Lemma 27). Lemma 27 relies on multiple technical lemmas, namely Lemmas 24 and 25. Note that we also have to prove that $\mathscr{A}'$ is indeed a WTA. In order to do this, we have to check that T' and final' have the correct types, that is, that all occurring weights are in $\mathbb{R}_\infty$. This is derived from Lemma 24 and is proven in Lemma 25.

Lemma 24. Let $P \in Q'$. For every $q \in Q$ it holds that $u_q^P \geq 0$. Moreover, there exists $q \in Q$ such that $u_q^P = 0$.

Proof. We prove the claim by induction on $\mathrm{gdepth}(P)$. Let $P \in Q'$ such that the claim holds for every $P' \in Q'$ such that $\mathrm{gdepth}(P') < \mathrm{gdepth}(P)$.

By item (a) of Lemma 23 there exists $\xi \in \mathrm{T}_\Sigma$ and a green run $\xrightarrow{\xi|\theta_{\mathscr{A}'}(\xi)} P$. Let $r \geq 0$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \ldots, \xi_r \in \mathrm{T}_\Sigma$ such that $\xi = \sigma(\xi_1, \ldots, \xi_r)$. Moreover, let $P_1, \ldots, P_r \in Q'$ be the states such that $\xrightarrow{\xi_i|x_i} P_i$ for every $i \in [r]$ (and some weights $x_1, \ldots, x_r \in \mathbb{R}_\infty$). Item (b) of Lemma 23 and the fact that $T'_\sigma(P_1, \ldots, P_r, P)$ is a green transition imply that $\mathrm{gdepth}(P_i) < \mathrm{gdepth}(P)$ (and hence the claim holds for P_i) for every $i \in [r]$.

By the definition of T', the for-loop starting in line 5 of Algorithm 1 was executed where the variables $P_1, \ldots, P_r$ were assigned to the states $P_1, \ldots, P_r$ of Q', respectively, and the variable P' was assigned to the state P. By the induction assumption and the nonnegativity of T, each upper bound u_q defined via line 9 is nonnegative. Hence by lines 10 and 12, we obtain the claim of the lemma for P. This concludes the induction step and hence the proof of the lemma. ☐

Lemma 25. For all $\sigma \in \Sigma$ it holds that $\mathrm{im}(T'_\sigma) \subseteq \mathbb{R}_\infty$ and $\mathrm{im}(\mathrm{final}') \subseteq \mathbb{R}_\infty$.

Proof. One easily sees that all occurring weights are in $\mathbb{R} \cup \{\infty\}$. Therefore, we only show their nonnegativity.

Every transition weight in T' is either ∞ or defined by line 10 of Algorithm 1. Thus, it suffices to prove that the variables u_q defined in line 9 only take on nonnegative values. However, this easily follows from Lemma 24, the fact that $t \geq 1$, and the fact that $\mathrm{im}(T_\sigma) \subseteq \mathbb{R}_\infty$ for every $\sigma \in \Sigma$.

Analogously, every final weight in final' is either ∞ or defined by line 17. However, by Lemma 24, the fact that $t \geq 1$, and the fact that $\mathrm{im}(\mathrm{final}) \subseteq \mathbb{R}_\infty$, we obtain that line 17 only defines nonnegative final weights. ☐

Corollary 26. If ttDet terminates on input $\mathscr{A}$ and t, then $\mathscr{A}'$ is a deterministic WTA.

Lemma 27. Let $\xi \in \mathrm{T}_\Sigma$ and $P \in Q'$ such that $\xrightarrow{\xi|\theta_{\mathscr{A}'}(\xi)} P$. It holds that for every $q \in Q$,

$$\theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi) \overset{\star}{\leq} l_q^P \leq u_q^P \overset{\star}{\leq} t \cdot \theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi). \tag{1}$$

Moreover, if $\xrightarrow{\xi|\theta_{\mathscr{A}'}(\xi)} P$ is a green[5] run, the $\star$-inequalities hold as equalities.

Proof. We first prove the inequality "$l_q^P \leq u_q^P$" by induction on $\mathrm{gdepth}(P)$. Note that the inequality is independent of ξ and hence by item (a) of Lemma 23 we can assume w.l.o.g. that there exists a green run ρ of $\mathscr{A}'$ on ξ. Let $r \geq 0$, $\sigma \in \Sigma^{(r)}$,

[5] We call a run ρ *green* if all transitions occurring in ρ are green.

and $\xi_1, \ldots, \xi_r \in \mathrm{T}_\Sigma$ such that $\xi = \sigma(\xi_1, \ldots, \xi_r)$. Moreover, let $P_1, \ldots, P_r \in Q'$ be the states such that $\xrightarrow{\xi_i | x_i} P_i$ for every $i \in [r]$ (and some weights $x_1, \ldots, x_r \in \mathbb{R}_\infty$). By item (b) of Lemma 23 and the induction hypothesis we obtain the claimed inequality for $P_1, \ldots, P_r$. The transition $T'_\sigma(P_1, \ldots, P_r, P)$ is green by assumption and hence has been generated by line 18 of Algorithm 1 (where the variables $P_1, \ldots, P_r$ were assigned to the states $P_1, \ldots, P_r$, respectively, and the variable P' was assigned to the state P). By lines 10 and 12 it suffices to show that $l_q^P \leq u_q^P$ for every $q \in Q$ and the variable P defined in line 7. However, for every $q_1, \ldots, q_r \in Q$ it holds that

$$l_{q_1}^{P_1} + \cdots + l_{q_r}^{P_r} + T_\sigma(q_1, \ldots, q_r, q) \leq u_{q_1}^{P_1} + \cdots + u_{q_r}^{P_r} + t \cdot T_\sigma(q_1, \ldots, q_r, q)$$

by the induction assumption and the fact that $t \geq 1$. This shows $l_q^P \leq u_q^P$ for the variable P defined in line 7, as desired.

Next we prove the $\star$-inequalities. The proof is done by induction on ξ. Let $r \geq 0$, $\sigma \in \Sigma^{(r)}$, and $\xi_1, \ldots, \xi_r \in \mathrm{T}_\Sigma$ such that $\xi = \sigma(\xi_1, \ldots, \xi_r)$. We obtain the claimed inequalities for ξ_i for every $i \in [r]$ by the induction assumption. Moreover, let $P_i \in Q'$ be the state such that $\xrightarrow{\xi_i | \theta_{\mathscr{A}'}(\xi_i)} P_i$ for every $i \in [r]$. We define the weight

$$c' \coloneqq \min\{u_{q_1}^{P_1} + \cdots + u_{q_r}^{P_r} + t \cdot T_\sigma(q_1, \ldots, q_r, q) \mid q, q_1, \ldots, q_r \in Q\}$$

and obtain the following inequality chain for every $q \in Q$.

$$\begin{aligned}
&\theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi) \\
&\overset{\star_1}{=} \min\{\big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i)\big) + T_\sigma(q_1, \ldots, q_r, q) \mid q_1, \ldots, q_r \in Q\} - \theta_{\mathscr{A}'}(\xi) \\
&\overset{\star_2}{=} \min\{\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i)\Big) + T_\sigma(q_1, \ldots, q_r, q) - \Big(\sum_{i=1}^{r} \theta_{\mathscr{A}'}(\xi_i)\Big) + c' \mid q_1, \ldots, q_r \in Q\} \\
&\overset{\star_3}{=} \min\{\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i) - \theta_{\mathscr{A}'}(\xi_i)\Big) + T_\sigma(q_1, \ldots, q_r, q) - c' \mid q_1, \ldots, q_r \in Q\} \\
&\overset{\star_4}{\leq} \min\{l_{q_1}^{P_1} + \cdots + l_{q_r}^{P_r} + T_\sigma(q_1, \ldots, q_r, q) - c' \mid q_1, \ldots, q_r \in Q\} \overset{\star_5}{\leq} l_q^P
\end{aligned}$$

Equality $\star_1$ uses Lemma 21. Equality $\star_2$ first pulls the term $\theta_{\mathscr{A}'}(\xi)$ inside the minimum and then uses the following argumentation. $\theta_{\mathscr{A}'}(\xi)$ is the weight $\mathrm{wt}(\rho)$ of the unique non-vanishing run ρ of $\mathscr{A}'$ on ξ and analogously $\theta_{\mathscr{A}'}(\xi_i) = \mathrm{wt}(\rho_i)$ for the unique non-vanishing run ρ_i of $\mathscr{A}'$ on ξ_i for every $i \in [r]$. By assumption, $\rho(\varepsilon) = P$ and $\rho_i(\varepsilon) = P_i$ for every $i \in [r]$. By the uniqueness of these runs, $\mathrm{wt}(\rho) = T'_\sigma(P_1, \ldots, P_r, P) + \sum_{i=1}^{r} \mathrm{wt}(\rho_i)$, whence $\theta_{\mathscr{A}'}(\xi) = \sum_{i=1}^{r} \theta_{\mathscr{A}'}(\xi_i) + c'$ by lines 14 and 18. Equality $\star_3$ simply rearranges the weights. Inequality $\star_4$ applies the induction hypothesis. Inequality $\star_5$ can be seen as follows. If $T_\sigma(P_1, \ldots, P_r, P)$ is a green transition, l_q^P is defined by lines 8, 10, and 12 and therefore, inequality $\star_5$ in fact holds as an equality. If $T_\sigma(P_1, \ldots, P_r, P)$ is a red transition, l_q^P is by refinement (line 13) greater or equal than the value defined by lines 8, 10, and 12.

We moreover obtain the following inequality chain for every $q \in Q$.

$$\begin{aligned}
u_q^P &\leq \min\{u_{q_1}^{P_1} + \cdots + u_{q_r}^{P_r} + t \cdot T_\sigma(q_1, \ldots, q_r, q) - c' \mid q_1, \ldots, q_r \in Q\} \\
&\leq \min\{\Big(\sum_{i=1}^{r} t \cdot \theta_{\mathscr{A}}(\xi_i, q_i) - \theta_{\mathscr{A}'}(\xi_i)\Big) + t \cdot T_\sigma(q_1, \ldots, q_r, q) - c' \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mid q_1, \ldots, q_r \in Q\} \\
&= \min\{t \cdot \Big(\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i)\Big) + T_\sigma(q_1, \ldots, q_r, q)\Big) - \Big(\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}'}(\xi_i)\Big) + c'\Big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mid q_1, \ldots, q_r \in Q\} \\
&= t \cdot \min\{\Big(\sum_{i=1}^{r} \theta_{\mathscr{A}}(\xi_i, q_i)\Big) + T_\sigma(q_1, \ldots, q_r, q) \mid q_1, \ldots, q_r \in Q\} - \theta_{\mathscr{A}'}(\xi) \\
&= t \cdot \theta_{\mathscr{A}}(\xi, q) - \theta_{\mathscr{A}'}(\xi)
\end{aligned}$$

Note that these equalities and inequalities are proven analogously to $\star_1, \ldots, \star_5$, exchanging lower residues by upper residues.

If $\xrightarrow{\xi \mid \theta_{\mathscr{A}'}(\xi)} P$ is a green run, inequality $\star_5$ holds as an equality by lines 8, 10, and 12 of Algorithm 1. Furthermore, inequality $\star_4$ holds as an equality by the induction assumption, as $\xrightarrow{\xi_i \mid x_i} P_i$ is a green run for every $i \in [r]$. This concludes the proof of the lemma. □

Proof of Theorem 15. Recall that we have to show $[\![\mathscr{A}]\!](\xi) \leq [\![\mathscr{A}']\!](\xi) \leq t \cdot [\![\mathscr{A}]\!](\xi)$ for every $\xi \in \mathrm{T}_\Sigma$.

Let $\xi \in \mathrm{T}_\Sigma$ and let $P \in Q'$ be the state such that there exists a run $\xrightarrow{\xi \mid \theta_{\mathscr{A}'}(\xi)} P$ of $\mathscr{A}'$ on ξ. It thus holds that $[\![\mathscr{A}']\!](\xi) = \theta_{\mathscr{A}'}(\xi) + \mathrm{final}'(P)$. Therefore, by subtracting $\theta_{\mathscr{A}'}(\xi)$ from all sides of the inequalities and using that $[\![\mathscr{A}]\!](\xi) = \min\{\theta_{\mathscr{A}}(\xi, q) + \mathrm{final}(q) \mid q \in Q\}$, we only need to show that

$$\begin{aligned}
\min_{q \in Q} \Big(\theta_{\mathscr{A}}(\xi, q) + \mathrm{final}(q)\Big) - \theta_{\mathscr{A}'}(\xi) &\overset{\star_1}{\leq} \mathrm{final}'(P) \\
&\overset{\star_2}{\leq} t \cdot \min_{q \in Q} \Big(\theta_{\mathscr{A}}(\xi, q) + \mathrm{final}(q)\Big) - \theta_{\mathscr{A}'}(\xi).
\end{aligned}$$

However, we already know that $\mathrm{final}'(P) = \min_{q \in Q}(u_q^P + t \cdot \mathrm{final}(q))$ (line 17) and hence by applying Lemma 27 to each u_q^P we obtain

$$\begin{aligned}
\min_{q \in Q} \Big(\theta_{\mathscr{A}}(\xi, q) + t \cdot \mathrm{final}(q)\Big) - \theta_{\mathscr{A}'}(\xi) &\overset{\star_3}{\leq} \mathrm{final}'(P) \\
&\overset{\star_4}{\leq} \min_{q \in Q} \Big(t \cdot \theta_{\mathscr{A}}(\xi, q) + t \cdot \mathrm{final}(q)\Big) - \theta_{\mathscr{A}'}(\xi).
\end{aligned}$$

Inequalities $\star_3$ and $\star_4$ imply inequalities $\star_1$ and $\star_2$, respectively, as it holds that $t \geq 1$. This concludes the proof. □

Proof of Theorem 17. First note the following fact. Given a constant $c \in \mathbb{R}_\infty$, denote by $c \cdot \mathscr{A}$ the WTA constructed from $\mathscr{A}$ by multiplying all occurring weights by c. If some deterministic WTA $\mathscr{B}$ t-approximates $\mathscr{A}$, then $c \cdot \mathscr{B}$ t-approximates $c \cdot \mathscr{A}$. Define d as the common denominator of all weights occurring in $\mathscr{A}$ and $c \coloneqq \frac{1}{t} \cdot d$. We show that $c \cdot \mathscr{A}$ is t-approximate determinisable, which proves the claim for $\mathscr{A} = \frac{1}{c} \cdot c \cdot \mathscr{A}$. Therefore, we henceforth assume that all weights of $\mathscr{A}$ are in $\frac{1}{t} \cdot \mathbb{N} \cup \{\infty\}$.

Assume that $\mathscr{A}$ satisfies the t-twinning property and ttDet does not terminate on input $\mathscr{A}$ and t. Hence, ttDet applied to $\mathscr{A}$ and t generates an infinite number of states. Observe that there exists an infinite sequence of states $\pi = P_0 P_1 P_2 \ldots$ such that for every $n \in \mathbb{N}$ the finite sequence $P_n \ldots P_0$ is a path of a green run[6] of $\mathscr{A}'$. The proof of this observation can be done similarly to the proof of König's Lemma [12]. In essence, the proof goes as follows. Consider the hypergraph $\mathcal{G}$ of green transitions of $\mathscr{A}'$. Note that for every $i \in \mathbb{N}$ there exist finitely many (but more than zero) vertices of depth i in $\mathcal{G}$. Therefore, by induction, each P_i can be chosen such that infinitely many green runs contain $P_i \ldots P_0$ as a suffix of a path. Then, the axiom of dependent choice yields the claim. In order to maintain the focus of this paper, we omit the formal proof of the existence of π.

Next observe that there exist states $\hat{q}, \bar{q} \in Q$ and a subsequence $\pi' = P_{i_0} P_{i_1} \ldots$ of π such that the sequence $(l_{\hat{q}}^{P_{i_k}})_{k \in \mathbb{N}}$ monotonically increases towards infinity for $k \to \infty$ and $u_{\bar{q}}^{P_{i_k}} = 0$ for all $k \in \mathbb{N}$. In fact, the very same property is proven during the proof of [2, Theorem 8] (for rational t). Our argumentation differs merely in the fact that all weights are multiplied by the factor $\frac{1}{t}$ from the argumentation presented in [2] and hence we omit the proof of this property.

We define the value

$$x \coloneqq \max\{\mathrm{wt}(\hat{\rho}) - t \cdot \mathrm{wt}(\bar{\rho}) \mid \xi \in \mathrm{T}_\Sigma, \mathrm{height}(\xi) \leq \#Q^2, \\ \hat{\rho} \in \mathrm{Run}_{\mathscr{A}}(\xi, \hat{q}), \bar{\rho} \in \mathrm{Run}_{\mathscr{A}}(\xi, \bar{q})\}.$$

Recall that for every $k \in \mathbb{N}$ the state P_{i_k} is reachable by a green run on some tree $\xi_k \in \mathrm{T}_\Sigma$. Therefore, Lemma 27 implies that $\theta_{\mathscr{A}}(\xi_k, \hat{q}) - \theta_{\mathscr{A}'}(\xi_k) = l_{\hat{q}}^{P_{i_k}}$ and $t \cdot \theta_{\mathscr{A}}(\xi_k, \bar{q}) - \theta_{\mathscr{A}'}(\xi_k) = u_{\bar{q}}^{P_{i_k}} (= 0)$. Subtracting the second equation from the first, we obtain $\theta_{\mathscr{A}}(\xi_k, \hat{q}) - t \cdot \theta_{\mathscr{A}}(\xi_k, \bar{q}) = l_{\hat{q}}^{P_{i_k}}$ for all $k \in \mathbb{N}$. By our construction of π', we obtain that there exists a $k \in \mathbb{N}$ such that $\theta_{\mathscr{A}}(\xi_k, \hat{q}) - t \cdot \theta_{\mathscr{A}}(\xi_k, \bar{q}) > x$. Therefore, by the definition of x we know that $\mathrm{height}(\xi_k) > \#Q^2$. Let $\hat{\rho} \in \mathrm{Run}_{\mathscr{A}}(\xi_k, \hat{q})$ and $\bar{\rho} \in \mathrm{Run}_{\mathscr{A}}(\xi_k, \bar{q})$ be runs such that $\mathrm{wt}(\hat{\rho}) = \theta_{\mathscr{A}}(\xi_k, \hat{q})$ and $\mathrm{wt}(\bar{\rho}) = \theta_{\mathscr{A}}(\xi_k, \bar{q})$. There exist $\zeta', \zeta \in \mathrm{C}_\Sigma$, and $\xi' \in \mathrm{T}_\Sigma$ such that $\xi_k = \zeta'[\zeta[\xi']]$, $\mathrm{size}(\zeta) > 1$, and both $\hat{\rho}$ and $\bar{\rho}$ loop[7] on ζ. Consider the restrictions[8] $\hat{\rho}|_{\zeta'[\xi']}$ and

[6] This is well-defined since runs of $\mathscr{A}'$ can be interpreted as trees in $\mathrm{T}_{Q'}$.

[7] In order to prove this, we consider the direct product automaton $\mathscr{A} \times \mathscr{A}$ and identify a loop in $\hat{\rho} \times \bar{\rho}$.

[8] Formally, let u be the unique position in $\mathrm{pos}_X(\zeta')$ and v be the unique position in $\mathrm{pos}_X(\zeta'[\zeta])$. Then, $\hat{\rho}|_{\zeta'[\xi']} \colon \mathrm{pos}(\zeta'[\xi']) \to Q$ where $w \mapsto \hat{\rho}(w)$ for every $w \in \mathrm{pos}_\Sigma(\zeta')$ and $uw \mapsto \hat{\rho}(vw)$ for every $w \in \mathrm{pos}(\xi)$.

$\bar{\rho}|_{\zeta'[\xi']}$ of $\hat{\rho}$ and $\bar{\rho}$ to $\zeta'[\xi']$, respectively. Note that $\mathrm{wt}(\hat{\rho}) - \mathrm{wt}(\hat{\rho}_{\zeta'[\xi']}) = \theta(\hat{q}, \zeta, \hat{q})$ by the assumption that $\mathrm{wt}(\hat{\rho}) = \theta_{\mathscr{A}}(\xi_k, \hat{q})$. Analogously, $\mathrm{wt}(\bar{\rho}) - \mathrm{wt}(\bar{\rho}_{\zeta'[\xi']}) = \theta(\bar{q}, \zeta, \bar{q})$. Therefore, the fact that $\mathscr{A}$ satisfies the t-twinning property implies that $\mathrm{wt}(\hat{\rho}) - \mathrm{wt}(\hat{\rho}_{\zeta'[\xi']}) \leq t \cdot (\mathrm{wt}(\bar{\rho}) - \mathrm{wt}(\bar{\rho}_{\zeta'[\xi']}))$, which can be rearranged as follows.

$$\mathrm{wt}(\hat{\rho}_{\zeta'[\xi']}) - t \cdot \mathrm{wt}(\bar{\rho}_{\zeta'[\xi']}) \geq \mathrm{wt}(\hat{\rho}) - t \cdot \mathrm{wt}(\bar{\rho})$$

Therefore, in total we obtain $\mathrm{wt}(\hat{\rho}_{\zeta'[\xi']}) - t \cdot \mathrm{wt}(\bar{\rho}_{\zeta'[\xi']}) > x$. As $\mathrm{size}(\zeta'[\xi']) < \mathrm{size}(\xi_k)$, repeated removal of loops from $\hat{\rho}$ and $\bar{\rho}$ results in a contradiction with the definition of x. This concludes the proof. □

Lemma 28. Let $\mathscr{A} = (Q, \Sigma, \mathbb{R}_\infty, \mathrm{final}, T)$ be a WTA and $t \in \mathbb{R}$ such that $t \geq 1$.

If $\mathscr{A}$ does not satisfy the t-twinning property, then there are siblings $p, q \in Q$ and a context $\zeta \in \mathrm{C}_\Sigma$ such that $\mathrm{size}(\zeta) \leq \mathrm{maxrk}(\Sigma)^{\#Q^2+1}$ and it holds that $\infty > \theta(p, \zeta, p) > t \cdot \theta(q, \zeta, q)$.

Proof. Assume that $\mathscr{A}$ does not satisfy the t-twinning property. Thus, there exist siblings $p, q \in Q$ and a context $\zeta \in \mathrm{C}_\Sigma$ such that $\infty > \theta(p, \zeta, p) > t \cdot \theta(q, \zeta, q)$ (after swapping p and q, if necessary). Among all possible such ζ, we chose the context such that $\mathrm{size}(\zeta)$ is minimal. Assume by way of contradiction that $\mathrm{size}(\zeta) > \mathrm{maxrk}(\Sigma)^{\#Q^2+1}$.

Let ρ_1 and ρ_2 be two runs of $\mathscr{A}$ on ζ such that $\mathrm{wt}(\rho_1) = \theta(p, \zeta, p)$ and $\mathrm{wt}(\rho_2) = \theta(q, \zeta, q)$. Hence, there exist $\zeta', \zeta'', \eta \in \mathrm{C}_\Sigma$ such that $\zeta = \zeta'[\eta[\zeta'']]$, $\mathrm{size}(\zeta) > \mathrm{size}(\eta) > 1$, and both ρ_1 and ρ_2 loop on η (in states q_1 and q_2, respectively). Consider the restrictions $\rho_1|_{\zeta'[\zeta'']}$ and $\rho_2|_{\zeta'[\zeta'']}$ of ρ_1 and ρ_2 to $\zeta'[\zeta'']$, respectively. The fact that $\mathrm{size}(\zeta) > \mathrm{size}(\eta)$ implies that $\theta(q_1, \eta, q_1) \leq t \cdot \theta(q_2, \eta, q_2)$. We ultimately obtain

$$\theta(p, \zeta'[\zeta''], p) \stackrel{\star}{=} \theta(p, \zeta, p) - \theta(q_1, \eta, q_1) > t \cdot (\theta(q, \zeta, q) - \theta(q_2, \eta, q_2)) \stackrel{\star}{=} t \cdot \theta(q, \zeta'[\zeta''], q),$$

where the $\star$-equations follow from the fact that ρ_1 and ρ_2 have minimal weights on ζ, on η, and on $\zeta'[\zeta'']$. In particular, we have found a smaller witness of the non-satisfaction of the t-twinning property than ζ, which is a contradiction. □

Proof of Theorem 20. First note that we can determine the set of siblings in Q by only considering trees $\xi \in \mathrm{T}_\Sigma$ such that $\mathrm{size}(\xi) \leq \mathrm{maxrk}(\Sigma)^{\#Q^2+1}$. This fact is proven analogously to Lemma 28 by removing synchronised loops from runs on bigger input trees.

By Lemma 28 $\mathscr{A}$ does not satisfy the t-twinning property if and only if there is a small witness to the non-satisfaction of the t-twinning property. Hence, we can calculate the finitely many values $\theta(p, \zeta, p)$ for states $p \in Q$ and small contexts $\zeta \in \mathrm{C}_\Sigma$ and check for the t-twinning property using the fact that we have already determined the set of siblings of Q. □