Acta Cybernetica **23** (2017) 379–401.

Minimization of Deterministic Top-down Tree Automata*

Zoltán Fülöp[a] and Sándor Vágvölgyi[a]

To the memory of Zoltán Ésik.

Abstract

We consider offline sensing unranked top-down tree automata in which the state transitions are computed by bimachines. We give a polynomial time algorithm for minimizing such tree automata when they are state-separated.

Keywords: bimachines, top-down unranked tree automata, minimization

1 Introduction

Minimization algorithms are necessary for the practical application of tree automata. Over ranked trees, Björklund and Cleophas [1] presented a taxonomy of algorithms for minimizing deterministic bottom-up tree automata, and Gécseg and Steinby [5] minimized deterministic top-down tree automata.

XML data or XML documents can be adequately represented by finite labeled unranked trees, where unranked means that nodes can have arbitrarily many children. This XML setting motivated the development of a theory of unranked tree automata, both bottom-up and top-down computing were studied [2, 10]. Bottom-up and top-down unranked tree automata have the same recognizing power [3]. Researchers usually abstract XML schema languages as Extended Document Type Definitions (EDTDs for short) instead of tree automata. Minimizing unranked tree automata or EDTDs is of both theoretical and practical importance [7].

In the case of bottom-up computing, Martens and Niehren [7] compared several notions of bottom-up determinism for unranked tree automata, minimized various types of deterministic bottom-up unranked tree automata, and showed that the minimization problem is NP-complete for bottom-up unranked tree automata in which the string languages in the transition functions are represented by deterministic finite state automata. For the size of deterministic bottom-up unranked tree

*The publication is supported by the the NKFI grant no K 108448.
[a]Institute of Informatics, University of Szeged, E-mail: {fulop,vagvolgy}@inf.u-szeged.hu

DOI: 10.14232/actacyb.23.1.2017.21

automata, Salomaa and Piao [11] presented upper and lower bounds for the union and intersection operations, and an upper bound for tree concatenation. They [12] presented a lower bound for the size blow-up of determinizing a nondeterministic unranked tree automaton.

For deterministic top-down unranked tree automata a blind version and a sensing version with two variants were introduced. The first variant is the online one, the state of a child depends on the state and the label of its parent and the labels of its left-siblings. Hence the child states are assigned when processing the child string in one pass from left to right. Online deterministic top-down unranked tree automata have been investigated in the context of XML schema languages, they were called as restrained competition EDTDs [9]. The second variant is the offline one, it first reads the complete child string and only then assigns states to all children. The blind, online, and offline sensing deterministic top-down unranked tree automata are increasingly more powerful, and all of them are less powerful than nondeterministic top-down unranked tree automata [8].

Minimization runs in nondeterministic polynomial time for deterministic blind top-down unranked tree automata, but the precise complexity is unknown, and runs in polynomial time for deterministic online sensing top-down unranked tree automata. Martens et al. [8] minimized deterministic offline sensing top-down unranked tree automata, where an unambiguous nondeterministic finite state automaton, associated with the state of the parent, reads the complete child string and assigns to each child the state it enters having read the child's label. They [8] reduced the minimization problem for unambiguous nondeterministic finite state automata, shown to be NP-complete by Jiang and Ravikumar [6], into the minimization for deterministic offline sensing top-down unranked tree automata, hence the minimization is NP-complete for deterministic offline sensing top-down unranked tree automata.

Cristau et al. [4] gave an equivalent formalism for deterministic offline sensing top-down unranked tree automata in terms of bimachines, from now on we refer to this notion simply as a deterministic top-down tree automaton (DTTA for short). A bimachine associated with the state of the parent assigns states to all children during the transition. Its two semi-automaton components read the child string from left-to-right and right-to-left, respectively, and its output function computes the state of a child depending on the label of the node and the states of the two semi-automata. Cristau et al. [4] noted that restrained competition EDTDs can be seen as a restricted version of the DTTAs. Martens and Niehren [7] minimized single-type and restrained completion EDTDs in polynomial time.

We minimize the size of a DTTA by minimizing the number of its states and the number of the states of the bimachines associated with its states. A state of a DTTA is an $\emptyset$-state if it accepts the empty tree language. A DTTA is state-separated if each transition yields a sequence of $\emptyset$-states or a sequence of non $\emptyset$-states. We show that it is decidable if a DTTA is state-separated. As the main result of our paper, we give a polynomial time minimizing algorithm for state-separated DTTAs. We will measure time as the number of elementary steps, assuming that each such step takes a constant time. The core of our algorithm is twofold. Following the ideas of

[5], we compute the connected part of the DTTA, find the equivalent states, and then collapse them into a single state, which is their equivalence class. Then we do similar minimization steps for the semi-automaton components of the bimachines associated with the DTTA states. We compute the connected parts of the semi-automata, find the equivalent semi-automaton states, and then collapse them into a single state, which is their equivalence class. Here two states of either of the semi-automata are equivalent if they yield the same output along all computations on any input word starting from them and any state of the other semi-automaton of the bimachine.

In Section 2, we present a brief review of the notions and notations used in the paper. In Section 3, we recall the concept of a bimachine, then study and minimize bimachines. In Section 4, we recall the concept of a DTTA. Then we present our minimization algorithm for state-separated DTTAs, and show the correctness of our algorithm.

2 Preliminaries

We denote by $\mathbb{N}$ the set of positive integers.

The cardinality of a set A is written as $|A|$. The composition of two mappings $f : A \to B$ and $g : B \to C$ is the mapping $f \circ g : A \to C$ defined by $f \circ g(a) = g(f(a))$ for every $a \in A$.

A *(binary) relation ρ over a set A* is a subset $\rho \subseteq A \times A$. For $(a, b) \in \rho$ we write $a\rho b$. We denote the reflexive and transitive closure of ρ by ρ^*. Let ρ be an *equivalence relation* (i.e., a reflexive, symmetric, and transitive relation) over A. For every $a \in A$, we denote by a/ρ the equivalence class which contains a, i.e., $a/\rho = \{b \in A \mid a\rho b\}$. Moreover, for every $B \subseteq A$ we define $B/\rho = \{a/\rho \mid a \in B\}$. Hence A/ρ is the set of all equivalence classes determined by ρ.

For a set X we denote by X^* the set of all finite words over X. The empty word is denoted by ε. For every $x \in \Sigma^*$, we denote by $|x|$ and x^{-1} the length and the reversal of x, respectively, and define them in the usual way.

A *tree domain* is a non-empty, finite, and prefix-closed subset D of $\mathbb{N}^*$ satisfying the following condition: if $xi \in D$ for $x \in \mathbb{N}^*$ and $i \in \mathbb{N}$, then $xj \in D$ for all j with $1 \leq j < i$.

Let Σ be an *alphabet*, i.e., a finite and non-empty set of symbols. An *unranked tree over* Σ (or just a *tree*) is a mapping $\xi : \mathrm{dom}(\xi) \to \Sigma$, where $\mathrm{dom}(\xi)$ is a tree domain. The elements of $\mathrm{dom}(\xi)$ are called the *nodes* of ξ. For every $x \in \mathrm{dom}(\xi)$ we call the element $\xi(x)$ of Σ the *label* of the node x and the number $\mathrm{rk}_\xi(x) = \max\{i \in \mathbb{N} \mid xi \in \mathrm{dom}(\xi)\}$ the *rank of the node x in ξ*. The *root* of ξ is $\xi(\varepsilon)$. If $xi \in \mathrm{dom}(\xi)$ for some $x \in \mathrm{dom}(\xi)$ and $i \in \mathbb{N}$, then we call xi the successor of x. As usual, a node of ξ without successors is called a *leaf* of ξ. The height $\mathrm{height}(\xi)$ of ξ is defined by $\mathrm{height}(\xi) = \max\{\, |x| \mid x \in \mathrm{dom}(\xi) \,\}$. We denote by T_Σ the set of all trees over Σ.

Furthermore, let $\xi, \xi' \in T_\Sigma$ and $x \in \mathrm{dom}(\xi)$. The *subtree* $\xi|_x$ of ξ at position x is defined by $\mathrm{dom}(\xi|_x) = \{y \in \mathbb{N}^* \mid xy \in \mathrm{dom}(\xi)\}$ and $\xi|_x(y) = \xi(xy)$ for all

$y \in \text{dom}(\xi|_x)$. Moreover, we denote by $\xi[x \leftarrow \xi']$ the tree which is obtained from ξ by "replacing $\xi|_x$ by ξ'", i.e. defined by

$$\text{dom}(\xi[x \leftarrow \xi']) = (\text{dom}(\xi) \setminus \{xy \mid y \in \mathbb{N}^*\}) \cup \{xy \mid y \in \text{dom}(\xi')\}$$

and

$$\xi[x \leftarrow \xi'](z) = \begin{cases} \xi(z) & \text{if } z \in (\text{dom}(\xi) \setminus \{xy \mid y \in \mathbb{N}^*\}) \\ \xi'(y) & \text{if } z = xy \text{ for some } y \in \text{dom}(\xi')\}. \end{cases}$$

If the root of ξ is labeled by a and the root has k successors at which the direct subtrees $\xi_1, \ldots, \xi_k$ are rooted, then we write $\xi = a(\xi_1 \ldots \xi_k)$.

Throughout the paper Σ and Γ denote arbitrary alphabets.

3 Bimachines

In this section we recall the concept of a bimachine, and establish a pumping lemma for bimachines and give a polynomial time algorithm for minimizing a bimachine.

3.1 General concepts

A *semi-automaton* is a quadruple $\mathcal{S} = (S, \Sigma, s_0, \delta)$, where S is a finite set (*states*), Σ is an alphabet (*input alphabet*), $s_0 \in S$ (*initial state*), and $\delta : S \times \Sigma \to S$ is a mapping (*transition mapping*). Let $s \in S$ be a state and $w = a_1 ... a_k \in \Sigma^*$ an input word. The *s-run of $\mathcal{S}$ on w* is the sequence $t_0 \ldots t_k$ of states such that $t_0 = s$ and $t_i = \delta(t_{i-1}, a_i)$ for all $1 \leq i \leq k$. We denote the state t_k also by $sw_{\mathcal{S}}$ or by sw if $\mathcal{S}$ is clear from the context. The s_0-run of $\mathcal{S}$ on w is called *the run of $\mathcal{S}$ on w*. A state $s \in S$ *is reachable (in $\mathcal{S}$)* if there is a $w \in \Sigma^*$ such that $s = s_0 w$. The set of all reachable states is $S^c = \{s_0 w \mid w \in \Sigma^*\}$. Moreover, *the connected part of $\mathcal{S}$* is the semi-automaton $\mathcal{S}^c = (S^c, \Sigma, s_0, \delta^c)$, where $\delta^c(s, a) = \delta(s, a)$ for each $s \in S^c$ and $a \in \Sigma$. (Note that $\delta(s, a) \in S^c$.) We call $\mathcal{S}$ *connected* if $S^c = S$. Obviously, $\mathcal{S}^c$ is connected. The following result is well-known.

Proposition 1. There is a polynomial time algorithm which constructs $\mathcal{S}^c$ for a given $\mathcal{S}$.

Proof. The standard algorithm runs in $\mathcal{O}(|S|^2|\Sigma|)$ time. □

A *congruence of $\mathcal{S}$* is an equivalence relation ρ over S such that $s \rho t$ implies $\delta(s, a) \rho \delta(t, a)$ for every $s, t \in S$ and $a \in \Sigma$. The *factor semi-automaton of $\mathcal{S}$ determined by a congruence ρ* is the semi-automaton $\mathcal{S}/\rho = (S/\rho, \Sigma, s_0/\rho, \delta_\rho)$, where $\delta_\rho(s/\rho, a) = \delta(s, a)/\rho$ for all $s \in S$ and $a \in \Sigma$.

Let $\mathcal{T} = (T, \Sigma, t_0, \delta')$ be a semi-automaton. A mapping $\varphi : S \to T$ is a *homomorphism from $\mathcal{S}$ to $\mathcal{T}$* if

- $\varphi(s_0) = t_0$ and,
- $\varphi(\delta(s, a)) = \delta'(\varphi(s), a)$ for every $s \in S$ and $a \in \Sigma$.

$$\begin{array}{cccccccccc}
s^{\rightarrow} = t_0^{\rightarrow} & & t_1^{\rightarrow} & & \cdots & & t_{k-2}^{\rightarrow} & & t_{k-1}^{\rightarrow} & & t_k^{\rightarrow} \\
& a_1 & & a_2 & & \cdots & & a_{k-1} & & a_k & \\
t_k^{\leftarrow} & & t_{k-1}^{\leftarrow} & & t_{k-2}^{\leftarrow} & \cdots & & & t_1^{\leftarrow} & & t_0^{\leftarrow} = s^{\leftarrow} \\
& & \downarrow & & \downarrow & \cdots & & & \downarrow & & \downarrow \\
& & b_1 & & b_2 & \cdots & & & b_{k-1} & & b_k
\end{array}$$

Figure 1: Visualization of the definition of the mapping $\|\mathcal{B}\|_{(s^{\rightarrow},s^{\leftarrow})}$

For a homomorphism φ from $\mathcal{S}$ to $\mathcal{T}$, we have $\varphi(sw_{\mathcal{S}}) = \varphi(s)w_{\mathcal{T}}$. If φ is a surjective homomorphism, then $\mathcal{T}$ is a *homomorphic image* of $\mathcal{S}$. If, in addition, φ is a bijection, then we say that $\mathcal{S}$ and $\mathcal{T}$ are *isomorphic* and write $\mathcal{S} \cong \mathcal{T}$.

A *bimachine* is a system $\mathcal{B} = (\Sigma, \Gamma, \mathcal{S}^{\rightarrow}, \mathcal{S}^{\leftarrow}, f)$, where Σ and Γ are alphabets (*input* and *output*), $\mathcal{S}^{\rightarrow} = (S^{\rightarrow}, \Sigma, s_0^{\rightarrow}, \delta^{\rightarrow})$ and $\mathcal{S}^{\leftarrow} = (S^{\leftarrow}, \Sigma, s_0^{\leftarrow}, \delta^{\leftarrow})$ are semi-automata, and $f : S^{\rightarrow} \times \Sigma \times S^{\leftarrow} \to \Gamma$ is a mapping (*output function*).

For every $s^{\rightarrow} \in S^{\rightarrow}$ and $s^{\leftarrow} \in S^{\leftarrow}$, we define the mapping

$$\|\mathcal{B}\|_{(s^{\rightarrow},s^{\leftarrow})} : \Sigma^* \to \Gamma^*$$

as follows. Let $\|\mathcal{B}\|_{(s^{\rightarrow},s^{\leftarrow})}(\varepsilon) = \varepsilon$. For every $k \geq 1$ and $w = a_1 \dots a_k \in \Sigma^*$, we let $\|\mathcal{B}\|_{(s^{\rightarrow},s^{\leftarrow})}(w) = b_1 \dots b_k$, where $b_1, \dots, b_k \in \Gamma$ are obtained as follows. Let

- $t_0^{\rightarrow} t_1^{\rightarrow} \dots t_{k-1}^{\rightarrow} t_k^{\rightarrow}$ be the $s^{\rightarrow}$-run of $\mathcal{S}^{\rightarrow}$ on $a_1 \dots a_k$,
- $t_0^{\leftarrow} t_1^{\leftarrow} \dots t_{k-1}^{\leftarrow} t_k^{\leftarrow}$ the $s^{\leftarrow}$-run of $\mathcal{S}^{\leftarrow}$ on the reversed input $a_k \dots a_1$, and
- let $b_i = f(t_{i-1}^{\rightarrow}, a_i, t_{k-i}^{\leftarrow})$ for $1 \leq i \leq k$, see Fig. 1.

We call $\|\mathcal{B}\|_{(s_0^{\rightarrow},s_0^{\leftarrow})}$ *the mapping computed by* $\mathcal{B}$ and denote it by $\|\mathcal{B}\|$.

Throughout the paper, $\mathcal{B}$ *and* $\mathcal{B}'$ *will denote the bimachines*

- $\mathcal{B} = (\Sigma, \Gamma, \mathcal{S}^{\rightarrow}, \mathcal{S}^{\leftarrow}, f)$, *with semi-automata* $\mathcal{S}^{\rightarrow} = (S^{\rightarrow}, \Sigma, s_0^{\rightarrow}, \delta^{\rightarrow})$ *and* $\mathcal{S}^{\leftarrow} = (S^{\leftarrow}, \Sigma, s_0^{\leftarrow}, \delta^{\leftarrow})$ *and*
- $\mathcal{B}' = (\Sigma, \Gamma, \mathcal{T}^{\rightarrow}, \mathcal{T}^{\leftarrow}, f')$ *with semi-automata* $\mathcal{T}^{\rightarrow} = (T^{\rightarrow}, \Sigma, t_0^{\rightarrow}, \gamma^{\rightarrow})$ *and* $\mathcal{T}^{\leftarrow} = (T^{\leftarrow}, \Sigma, t_0^{\leftarrow}, \gamma^{\leftarrow})$,

respectively.

The bimachines $\mathcal{B}$ and $\mathcal{B}'$ are *equivalent* if $\|\mathcal{B}\| = \|\mathcal{B}'\|$. Next we prove a pumping lemma for bimachines.

Lemma 1. There is an integer $N > 0$ such that for every $x \in \Sigma^*$ with $|x| > N$ and $\|\mathcal{B}\|(x) = y$, there are $x_1, x_2, x_3 \in \Sigma^*$ and $y_1, y_2, y_3 \in \Gamma^*$ such that

- $x = x_1 x_2 x_3$ and $y = y_1 y_2 y_3$,
- $|x_i| = |y_i|$ for $1 \leq i \leq 3$,
- $0 < |x_2| = |y_2| \leq N$, and
- $\|\mathcal{B}\|(x_1 x_2^n x_3) = y_1 y_2^n y_3$ for every $n \geq 0$.

Proof. Let $N = |S^{\rightharpoondown}||S^{\leftharpoondown}||\Sigma|$. Moreover, let $x = a_1 \dots a_k \in \Sigma^*$ be an input string with $a_1, \dots, a_k \in \Sigma$ and $k > N$ and $\|\mathcal{B}\|(x) = y$. Let

- $s_0^{\rightharpoondown} s_1^{\rightharpoondown} \dots s_{k-1}^{\rightharpoondown} s_k^{\rightharpoondown}$ be the run of $\mathcal{S}^{\rightharpoondown}$ on $a_1 \dots a_k$,
- $s_0^{\leftharpoondown} s_1^{\leftharpoondown} \dots s_{k-1}^{\leftharpoondown} s_k^{\leftharpoondown}$ the run of $\mathcal{S}^{\leftharpoondown}$ on $a_k \dots a_1$, and
- let $b_i = f(s_{i-1}^{\rightharpoondown}, a_i, s_{k-i}^{\leftharpoondown})$ for $1 \le i \le k$.

Then $y = b_1 \dots b_k$. Since $k > N$, there are $1 \le i < j \le k$ such that

$$(s_{i-1}^{\rightharpoondown}, a_i, s_{k-i}^{\leftharpoondown}) = (s_{j-1}^{\rightharpoondown}, a_j, s_{k-j}^{\leftharpoondown}).$$

We may assume w.l.o.g. that the triples in the sequence $(s_i^{\rightharpoondown}, a_{i+1}, s_{k-i-1}^{\leftharpoondown}) \dots (s_{j-1}^{\rightharpoondown}, a_j, s_{k-j}^{\leftharpoondown})$ are pairwise different. Then we define

$$x_1 = a_1 \dots a_i, \qquad x_2 = a_{i+1} \dots a_j, \text{ and } \qquad x_3 = a_{j+1} \dots a_k,$$

and decompose y into y_1, y_2, and y_3 accordingly. By standard arguments we can show that these decompositions of x and y satisfy the requirements of the lemma. □

Let $s^{\rightharpoondown} \in S^{\rightharpoondown}$, $s^{\leftharpoondown} \in S^{\leftharpoondown}$, and $x, y \in \Sigma^*$. It should be clear that

$$\|\mathcal{B}\|_{(s^{\rightharpoondown}, s^{\leftharpoondown})}(xy) = \|\mathcal{B}\|_{(s^{\rightharpoondown}, s^{\leftharpoondown} y^{-1})}(x) \|\mathcal{B}\|_{(s^{\rightharpoondown} x, s^{\leftharpoondown})}(y).$$

We will use this fact later in the paper.

Let $s^{\rightharpoondown} \in S^{\rightharpoondown}$ and $s^{\leftharpoondown} \in S^{\leftharpoondown}$. The pair $(s^{\rightharpoondown}, s^{\leftharpoondown})$ is *reachable (in $\mathcal{B}$)* if there is a string $x = a_1 \dots a_k \in \Sigma^*$ with runs

- $s_0^{\rightharpoondown} s_1^{\rightharpoondown} \dots s_{k-1}^{\rightharpoondown} s_k^{\rightharpoondown}$ of $\mathcal{S}^{\rightharpoondown}$ on $a_1 \dots a_k$ and
- $s_0^{\leftharpoondown} s_1^{\leftharpoondown} \dots s_{k-1}^{\leftharpoondown} s_k^{\leftharpoondown}$ of $\mathcal{S}^{\leftharpoondown}$ on $a_k \dots a_1$

such that $(s^{\rightharpoondown}, s^{\leftharpoondown}) = (s_{i-1}^{\rightharpoondown}, s_{k-i}^{\leftharpoondown})$ for some $1 \le i \le k$. We note that $(s^{\rightharpoondown}, s^{\leftharpoondown})$ is reachable in $\mathcal{B}$ if and only if $s^{\rightharpoondown}$ is reachable in $\mathcal{S}^{\rightharpoondown}$ and $s^{\leftharpoondown}$ is reachable in $\mathcal{S}^{\leftharpoondown}$.

The connected part of $\mathcal{B}$ is the bimachine $\mathcal{B}^c = (\Sigma, \Gamma, \mathcal{S}^{\rightharpoondown c}, \mathcal{S}^{\leftharpoondown c}, f^c)$, where:

- $\mathcal{S}^{\rightharpoondown c}$ and $\mathcal{S}^{\leftharpoondown c}$ are the connected parts of $\mathcal{S}^{\rightharpoondown}$ and $\mathcal{S}^{\leftharpoondown}$, respectively,
- $f^c(s^{\rightharpoondown}, a, s^{\leftharpoondown}) = f(s^{\rightharpoondown}, a, s^{\leftharpoondown})$ for every $s^{\rightharpoondown} \in S^{\rightharpoondown c}$, $s^{\leftharpoondown} \in S^{\leftharpoondown c}$, and $a \in \Sigma$.

It is obvious that $\mathcal{B}^c$ is equivalent to $\mathcal{B}$. We call $\mathcal{B}$ *connected* if $\mathcal{B}^c = \mathcal{B}$. We note that $\mathcal{B}$ is connected if both $\mathcal{S}^{\rightharpoondown}$ and $\mathcal{S}^{\leftharpoondown}$ are connected. Hence $\mathcal{B}^c$ is connected. By Proposition 1 we have the following result.

Proposition 2. There is a polynomial time algorithm which constructs $\mathcal{B}^c$ for a given $\mathcal{B}$.

Proof. We compute $\mathcal{S}^{\rightharpoondown c}$ and $\mathcal{S}^{\leftharpoondown c}$. Thus, by Proposition 1, the algorithm runs in $\mathcal{O}((|S^{\rightharpoondown}|^2 + |S^{\leftharpoondown}|^2)|\Sigma|)$ time. □

A *congruence ρ of $\mathcal{B}$* is a pair $(\rho^{\rightharpoondown}, \rho^{\leftharpoondown})$, where

- $\rho^{\rightharpoondown}$ and $\rho^{\leftharpoondown}$ are congruences of the semi-automata $\mathcal{S}^{\rightharpoondown}$ and $\mathcal{S}^{\leftharpoondown}$, respectively, and
- for all $s^{\rightharpoondown}, t^{\rightharpoondown} \in S^{\rightharpoondown}$, $s^{\leftharpoondown}, t^{\leftharpoondown} \in S^{\leftharpoondown}$, and $a \in \Sigma$, if $s^{\rightharpoondown} \rho^{\rightharpoondown} t^{\rightharpoondown}$ and $s^{\leftharpoondown} \rho^{\leftharpoondown} t^{\leftharpoondown}$, then

$$f(s^{\rightharpoondown}, a, s^{\leftharpoondown}) = f(t^{\rightharpoondown}, a, t^{\leftharpoondown}).$$

For a congruence $\rho = (\rho^{\to}, \rho^{\leftarrow})$ of $\mathcal{B}$, we define the *factor bimachine of* $\mathcal{B}$ *determined by* ρ to be

$$\mathcal{B}/\rho = (\Sigma, \Gamma, \mathcal{S}^{\to}/\rho^{\to}, \mathcal{S}^{\leftarrow}/\rho^{\leftarrow}, f_\rho),$$

where $f_\rho(s^{\to}/\rho^{\to}, a, s^{\leftarrow}/\rho^{\leftarrow}) = f(s^{\to}, a, s^{\leftarrow})$ for all $s^{\to} \in S^{\to}, s^{\leftarrow} \in S^{\leftarrow}$, and $a \in \Sigma$.

A pair $\varphi = (\varphi^{\to}, \varphi^{\leftarrow})$ of mappings $\varphi^{\to} : S^{\to} \to T^{\to}$ and $\varphi^{\leftarrow} : S^{\leftarrow} \to T^{\leftarrow}$ is a *homomorphism from* $\mathcal{B}$ *to* $\mathcal{B}'$ if $\varphi^{\to}$ and $\varphi^{\leftarrow}$ are homomorphisms from $\mathcal{S}^{\to}$ to $\mathcal{T}^{\to}$ and $\mathcal{S}^{\leftarrow}$ to $\mathcal{T}^{\leftarrow}$, respectively, and in addition $f(s^{\to}, a, s^{\leftarrow}) = f'(\varphi^{\to}(s^{\to}), a, \varphi^{\leftarrow}(s^{\leftarrow}))$ for every $s^{\to} \in S^{\to}$, $s^{\leftarrow} \in S^{\leftarrow}$, and $a \in \Sigma$. If φ is a homomorphism and both $\varphi^{\to}$ and $\varphi^{\leftarrow}$ are surjective, then $\mathcal{B}'$ is a *homomorphic image* of $\mathcal{B}$. If both $\varphi^{\to}$ and $\varphi^{\leftarrow}$ are bijections, then we say that $\mathcal{B}$ and $\mathcal{B}'$ are *isomorphic* and write $\mathcal{B} \cong \mathcal{B}'$.

Lemma 2. If there is a homomorphism $\varphi = (\varphi^{\to}, \varphi^{\leftarrow})$ from $\mathcal{B}$ to $\mathcal{B}'$, then

$$\|\mathcal{B}\|_{(s^{\to}, s^{\leftarrow})} = \|\mathcal{B}'\|_{(\varphi^{\to}(s^{\to}), \varphi^{\leftarrow}(s^{\leftarrow}))}$$

for every $s^{\to} \in S^{\to}$ and $s^{\leftarrow} \in S^{\leftarrow}$. In particular, $\|\mathcal{B}\| = \|\mathcal{B}'\|$.

Proof. For every $s^{\to} \in S^{\to}$ and $s^{\leftarrow} \in S^{\leftarrow}$, $\|\mathcal{B}\|_{(s^{\to}, s^{\leftarrow})}(\varepsilon) = \varepsilon$ and $\|\mathcal{B}'\|_{(\varphi^{\to}(s^{\to}), \varphi^{\leftarrow}(s^{\leftarrow}))}(\varepsilon) = \varepsilon$.

Let $k \geq 1$ and $w = a_1 \dots a_k \in \Sigma^*$. Then $\|\mathcal{B}\|_{(s^{\to}, s^{\leftarrow})}(w) = b_1 \dots b_k$, where $b_1, \dots, b_k$ are obtained as follows. Let

- $t_0^{\to} t_1^{\to} \dots t_{k-1}^{\to} t_k^{\to}$ be the $s^{\to}$-run of $\mathcal{S}^{\to}$ on $a_1 \dots a_k$,
- $t_0^{\leftarrow} t_1^{\leftarrow} \dots t_{k-1}^{\leftarrow} t_k^{\leftarrow}$ be the $s^{\leftarrow}$-run of $\mathcal{S}^{\leftarrow}$ on the reversed input $a_k \dots a_1$, and
- $b_i = f(t_{i-1}^{\to}, a_i, t_{k-i}^{\leftarrow})$ for $1 \leq i \leq k$, see Fig. 1.

Then

- $\varphi^{\to}(t_0^{\to})\varphi^{\to}(t_1^{\to}) \dots \varphi^{\to}(t_{k-1}^{\to})\varphi^{\to}(t_k^{\to})$ is the $\varphi^{\to}(s^{\to})$-run of $\mathcal{T}^{\to}$ on $a_1 \dots a_k$, and
- $\varphi^{\leftarrow}(t_0^{\leftarrow})\varphi^{\leftarrow}(t_1^{\leftarrow}) \dots \varphi^{\leftarrow}(t_{k-1}^{\leftarrow})\varphi^{\leftarrow}(t_k^{\leftarrow})$ is the $\varphi^{\leftarrow}(s^{\leftarrow})$-run of $\mathcal{T}^{\leftarrow}$ on the reversed input $a_k \dots a_1$.

As φ is a homomorphism, $b_i = f'(\varphi^{\to}(t_{i-1}^{\to}), a_i, \varphi^{\leftarrow}(t_{k-i}^{\leftarrow}))$ for $1 \leq i \leq k$. Hence $\|\mathcal{B}'\|_{(\varphi^{\to}(s^{\to}), \varphi^{\leftarrow}(s^{\leftarrow}))}(w) = b_1 \dots b_k$. ☐

By the corresponding definitions we have the following result.

Lemma 3. If there is a surjective homomorphism φ from $\mathcal{B}$ to $\mathcal{B}'$, then $|T_q^{\to}| \leq |S_q^{\to}|$ and $|T_q^{\leftarrow}| \leq |S_q^{\leftarrow}|$.

Lemma 4. If ρ is a congruence of $\mathcal{B}$, then $\mathcal{B}/\rho$ is a homomorphic image of $\mathcal{B}$.

Proof. It is easy to check that the mapping $\varphi^{\to} : S^{\to} \to S^{\to}/\rho^{\to}$ defined by $\varphi^{\to}(s^{\to}) = s^{\to}/\rho^{\to}$ is a surjective homomorphism from $\mathcal{S}^{\to}$ to $\mathcal{S}^{\to}/\rho^{\to}$. Also, the mapping $\varphi^{\leftarrow} : S^{\leftarrow} \to S^{\leftarrow}/\rho^{\leftarrow}$ defined analogously is a surjective homomorphism from $\mathcal{S}^{\leftarrow}$ to $\mathcal{S}^{\leftarrow}/\rho^{\leftarrow}$. ☐

Lemma 5. Let $\rho = (\rho^{\to}, \rho^{\leftarrow})$ be a congruence of the bimachine $\mathcal{B}$. Then

$$\|\mathcal{B}\|_{(s^{\to}, s^{\leftarrow})} = \|\mathcal{B}/\rho\|_{(s^{\to}/\rho^{\to}, s^{\leftarrow}/\rho^{\leftarrow})}$$

for all $s^{\to} \in S^{\to}$ and $s^{\leftarrow} \in S^{\leftarrow}$. In particular, $\|\mathcal{B}\| = \|\mathcal{B}/\rho\|$.

Proof. It follows from Lemmas 2 and 4. ☐

3.2 Minimization of bimachines

The bimachine $\mathcal{B}$ is called *minimal* if $|S^{\rightarrow}| \leq |T^{\rightarrow}|$ and $|S^{\leftarrow}| \leq |T^{\leftarrow}|$ for any bimachine $\mathcal{B}'$ which is equivalent to $\mathcal{B}$.

We introduce the relation $\rho_{\mathcal{B}}^{\rightarrow} \subseteq S^{\rightarrow} \times S^{\rightarrow}$ as follows: for all $s^{\rightarrow}, t^{\rightarrow} \in S^{\rightarrow}$, we have $s^{\rightarrow} \rho_{\mathcal{B}}^{\rightarrow} t^{\rightarrow}$ if $\|\mathcal{B}\|_{(s^{\rightarrow}, s^{\leftarrow})} = \|\mathcal{B}\|_{(t^{\rightarrow}, s^{\leftarrow})}$ for all $s^{\leftarrow} \in S^{\leftarrow}$. Analogously, we define $\rho_{\mathcal{B}}^{\leftarrow} \subseteq S^{\leftarrow} \times S^{\leftarrow}$ such that for all $s^{\leftarrow}, t^{\leftarrow} \in S^{\leftarrow}$, we have $s^{\leftarrow} \rho_{\mathcal{B}}^{\leftarrow} t^{\leftarrow}$ if $\|\mathcal{B}\|_{(s^{\rightarrow}, s^{\leftarrow})} = \|\mathcal{B}\|_{(s^{\rightarrow}, t^{\leftarrow})}$ for all $s^{\rightarrow} \in S^{\rightarrow}$. Moreover, let $\rho_{\mathcal{B}} = (\rho_{\mathcal{B}}^{\rightarrow}, \rho_{\mathcal{B}}^{\leftarrow})$.

Lemma 6. The relations $\rho_{\mathcal{B}}^{\rightarrow}$ and $\rho_{\mathcal{B}}^{\leftarrow}$ are congruences of the semi-automata $\mathcal{S}^{\rightarrow}$ and $\mathcal{S}^{\leftarrow}$, respectively. Moreover, $\rho_{\mathcal{B}}$ is a congruence of $\mathcal{B}$.

Proof. We show that $\rho_{\mathcal{B}}^{\rightarrow}$ is a congruence of $\mathcal{S}^{\rightarrow}$. Obviously, $\rho_{\mathcal{B}}^{\rightarrow}$ is an equivalence relation. Now let $s^{\rightarrow}, t^{\rightarrow}, u^{\rightarrow}, v^{\rightarrow} \in \mathcal{S}^{\rightarrow}$, and $a \in \Sigma$ such that $s^{\rightarrow} \rho_{\mathcal{B}}^{\rightarrow} t^{\rightarrow}$, $u^{\rightarrow} = \delta^{\rightarrow}(s^{\rightarrow}, a)$, and $v^{\rightarrow} = \delta^{\rightarrow}(t^{\rightarrow}, a)$. Moreover, let $w \in \Sigma^*$ and $s^{\leftarrow} \in S^{\leftarrow}$. Then we have

$$\|\mathcal{B}\|_{(s^{\rightarrow}, s^{\leftarrow})}(aw) = f(s^{\rightarrow}, a, t^{\leftarrow}) \|\mathcal{B}\|_{(u^{\rightarrow}, s^{\leftarrow})}(w), \text{ and}$$
$$\|\mathcal{B}\|_{(t^{\rightarrow}, s^{\leftarrow})}(aw) = f(t^{\rightarrow}, a, t^{\leftarrow}) \|\mathcal{B}\|_{(v^{\rightarrow}, s^{\leftarrow})}(w),$$

where $t^{\leftarrow} = s^{\leftarrow} w^{-1}$ in $\mathcal{S}^{\leftarrow}$. Since the left-hand side of both equalities are the same, we obtain $\|\mathcal{B}\|_{(u^{\rightarrow}, s^{\leftarrow})}(w) = \|\mathcal{B}\|_{(v^{\rightarrow}, s^{\leftarrow})}(w)$, which proves that $u^{\rightarrow} \rho_{\mathcal{B}}^{\rightarrow} v^{\rightarrow}$.

Analogously, we can show that $\rho_{\mathcal{B}}^{\leftarrow}$ is a congruence of the semi-automata $\mathcal{S}^{\leftarrow}$.

Finally, let $s^{\rightarrow}, t^{\rightarrow} \in S^{\rightarrow}$, $s^{\leftarrow}, t^{\leftarrow} \in S^{\leftarrow}$, and $a \in \Sigma$ such that $s^{\rightarrow} \rho_{\mathcal{B}}^{\rightarrow} t^{\rightarrow}$ and $s^{\leftarrow} \rho_{\mathcal{B}}^{\leftarrow} t^{\leftarrow}$. Then

$$\|\mathcal{B}\|_{(s^{\rightarrow}, s^{\leftarrow})} = \|\mathcal{B}\|_{(t^{\rightarrow}, s^{\leftarrow})} = \|\mathcal{B}\|_{(t^{\rightarrow}, t^{\leftarrow})}.$$

In particular, $\|\mathcal{B}\|_{(s^{\rightarrow}, s^{\leftarrow})}(a) = \|\mathcal{B}\|_{(t^{\rightarrow}, t^{\leftarrow})}(a)$, i.e., $f(s^{\rightarrow}, a, s^{\leftarrow}) = f(t^{\rightarrow}, a, t^{\leftarrow})$. Hence, $\rho_{\mathcal{B}}$ is a congruence of $\mathcal{B}$. □

Let us recall that $\mathcal{B}/\rho_{\mathcal{B}} = (\Sigma, \Gamma, \mathcal{S}^{\rightarrow}/\rho_{\mathcal{B}}^{\rightarrow}, \mathcal{S}^{\leftarrow}/\rho_{\mathcal{B}}^{\leftarrow}, f_{\rho_{\mathcal{B}}})$.

A bimachine $\mathcal{B}$ is called *reduced* if both $\rho_{\mathcal{B}}^{\rightarrow}$ and $\rho_{\mathcal{B}}^{\leftarrow}$ are the identity relation. It is easy to check that $\mathcal{B}/\rho_{\mathcal{B}}$ is reduced.

Lemma 7. Assume that $\mathcal{B}$ and $\mathcal{B}'$ are connected and reduced. Then

$$\|\mathcal{B}\| = \|\mathcal{B}'\| \text{ if and only if } \mathcal{B} \cong \mathcal{B}'.$$

Proof. Assume that $\|\mathcal{B}\| = \|\mathcal{B}'\|$. Let us define the relation $\varphi^{\rightarrow} \subseteq S^{\rightarrow} \times T^{\rightarrow}$ as follows:

$$\varphi^{\rightarrow} = \{(s_0^{\rightarrow} x, t_0^{\rightarrow} x) \mid x \in \Sigma^*\}.$$

The domain of $\varphi^{\rightarrow}$ is $S^{\rightarrow}$ because $\mathcal{B}$ is connected. First we show by contradiction that $\varphi^{\rightarrow}$ is a mapping. For this, we assume that there are $x, y \in \Sigma^*$ such that $s_0^{\rightarrow} x = s_0^{\rightarrow} y$ and $t_0^{\rightarrow} x \neq t_0^{\rightarrow} y$. Since $\mathcal{B}'$ is reduced, there are $u, z \in \Sigma^*$ such that $\|\mathcal{B}'\|_{(t_0^{\rightarrow} x, t_0^{\leftarrow} z)}(u) \neq \|\mathcal{B}'\|_{(t_0^{\rightarrow} y, t_0^{\leftarrow} z)}(u)$. Consequently, $\|\mathcal{B}'\|_{(t_0^{\rightarrow} x, t_0^{\leftarrow})}(uz^{-1}) \neq \|\mathcal{B}'\|_{(t_0^{\rightarrow} y, t_0^{\leftarrow})}(uz^{-1})$. On the other hand, by $\|\mathcal{B}\| = \|\mathcal{B}'\|$,

$$\|\mathcal{B}\|_{(s_0^{\rightarrow}, s_0^{\leftarrow})}(xuz^{-1}) = \|\mathcal{B}'\|_{(t_0^{\rightarrow}, t_0^{\leftarrow})}(xuz^{-1}) \text{ and}$$
$$\|\mathcal{B}\|_{(s_0^{\rightarrow}, s_0^{\leftarrow})}(yuz^{-1}) = \|\mathcal{B}'\|_{(t_0^{\rightarrow}, t_0^{\leftarrow})}(yuz^{-1}).$$

Thus

$$\|\mathcal{B}\|_{(s_0^{\rightarrow}x, s_0^{\leftarrow})}(uz^{-1}) = \|\mathcal{B}'\|_{(t_0^{\rightarrow}x, t_0^{\leftarrow})}(uz^{-1}) \text{ and}$$
$$\|\mathcal{B}\|_{(s_0^{\rightarrow}y, s_0^{\leftarrow})}(uz^{-1}) = \|\mathcal{B}'\|_{(t_0^{\rightarrow}y, t_0^{\leftarrow})}(uz^{-1}).$$

Hence $\|\mathcal{B}\|_{(s_0^{\rightarrow}x, s_0^{\leftarrow})}(uz^{-1}) \neq \|\mathcal{B}\|_{(s_0^{\rightarrow}y, s_0^{\leftarrow})}(uz^{-1})$, which is a contradiction by our assumption $s_0^{\rightarrow}x = s_0^{\rightarrow}y$.

By interchanging the role of $\mathcal{B}$ and $\mathcal{B}'$, we obtain that $\varphi^{\rightarrow}$ is injective. Moreover, it is obvious that $\varphi^{\rightarrow}$ is surjective.

We can also show that $\varphi^{\rightarrow}$ is a homomorphism. For this, let $x \in \Sigma^*$ and $a \in \Sigma$. Then we have

$$\varphi^{\rightarrow}(\delta^{\rightarrow}(s_0^{\rightarrow}x, a)) = \varphi^{\rightarrow}(s_0^{\rightarrow}xa) = t_0^{\rightarrow}xa = \gamma^{\rightarrow}(t_0^{\rightarrow}x, a) = \gamma^{\rightarrow}(\varphi^{\rightarrow}(s_0^{\rightarrow}x), a).$$

Thus $\mathcal{S}^{\rightarrow}$ and $\mathcal{T}^{\rightarrow}$ are isomorphic.

Analogously, we can define the relation $\varphi^{\leftarrow}$ and show that it is an isomorphism between $\mathcal{S}^{\leftarrow}$ and $\mathcal{T}^{\leftarrow}$. Finally, we show that the pair $\varphi = (\varphi^{\rightarrow}, \varphi^{\leftarrow})$ is an isomorphism between $\mathcal{B}$ and $\mathcal{B}'$. For this, let $x, y \in \Sigma^*$ and $a \in \Sigma$. Then

$$\|\mathcal{B}\|(xay^{-1}) = \|\mathcal{B}'\|(xay^{-1}).$$

By the corresponding definition this means that $f(s_0^{\rightarrow}x, a, s_0^{\leftarrow}y) = f'(t_0^{\rightarrow}x, a, t_0^{\leftarrow}y)$. With this we have proved that $\mathcal{B} \cong \mathcal{B}'$. The proof of the other implication is trivial. □

Lemma 8. Assume that $\mathcal{B}$ and $\mathcal{B}'$ are connected. Then

$$\|\mathcal{B}\| = \|\mathcal{B}'\| \text{ if and only if } \mathcal{B}/\rho_{\mathcal{B}} \cong \mathcal{B}'/\rho_{\mathcal{B}'}.$$

Proof. If $\mathcal{B}/\rho_{\mathcal{B}} \cong \mathcal{B}'/\rho_{\mathcal{B}'}$, then by Lemma 5 we obtain $\|\mathcal{B}\| = \|\mathcal{B}'\|$.

Next assume that $\|\mathcal{B}\| = \|\mathcal{B}'\|$. Again, by Lemma 5 we obtain $\|\mathcal{B}/\rho_{\mathcal{B}}\| = \|\mathcal{B}'/\rho_{\mathcal{B}'}\|$. Moreover, both $\mathcal{B}/\rho_{\mathcal{B}}$ and $\mathcal{B}'/\rho_{\mathcal{B}'}$ are connected and reduced. Hence, by Lemma 7, $\mathcal{B}/\rho_{\mathcal{B}} \cong \mathcal{B}'/\rho_{\mathcal{B}'}$. □

Theorem 1. If the bimachine $\mathcal{B}$ is connected, then $\mathcal{B}/\rho_{\mathcal{B}}$ is minimal.

Proof. Let us assume that $\mathcal{B}$ is connected and that $\|\mathcal{B}\| = \|\mathcal{B}'\|$. By Lemma 4, $\mathcal{B}'/\rho_{\mathcal{B}'}$ is a homomorphic image of $\mathcal{B}'$. By Lemma 8, $\mathcal{B}/\rho_{\mathcal{B}} \cong \mathcal{B}'/\rho_{\mathcal{B}'}$. Hence $\mathcal{B}/\rho_{\mathcal{B}}$ is also a homomorphic image of $\mathcal{B}'$. □

In the rest of this section we give an algorithm which computes $\rho_{\mathcal{B}}$, i.e., $\rho_{\mathcal{B}}^{\rightarrow}$ and $\rho_{\mathcal{B}}^{\leftarrow}$. First we deal with $\rho_{\mathcal{B}}^{\rightarrow}$.

For every $i \geq 1$, we define the relation $\rho_i^{\rightarrow} \subseteq S^{\rightarrow} \times S^{\rightarrow}$, by induction as follows. For all $s^{\rightarrow}, t^{\rightarrow} \in S^{\rightarrow}$,

(i) let $s^{\rightarrow} \rho_1^{\rightarrow} t^{\rightarrow}$ if for all $a \in \Sigma$ and $s^{\leftarrow} \in S^{\leftarrow}$, we have $f(s^{\rightarrow}, a, s^{\leftarrow}) = f(t^{\rightarrow}, a, s^{\leftarrow})$, and

(ii) for each $i \geq 1$, let $s^\to \rho^\to_{i+1} t^\to$ if $s^\to \rho^\to_i t^\to$ and $\delta^\to(s^\to, a)\rho^\to_i \delta^\to(t^\to, a)$ for each $a \in \Sigma$.

Obviously, we have

$$\rho^\to_1 \supseteq \rho^\to_2 \supseteq \cdots$$

and thus there is an integer $i \geq 1$ such that $\rho^\to_i = \rho^\to_{i+1}$.

For the rest of this section, let i_0 be the least integer such that $\rho^\to_{i_0} = \rho^\to_{i_0+1}$. We will show that $\rho^\to_{i_0} = \rho^\to_{\mathcal{B}}$.

Claim 1. $\rho^\to_{i_0+1} = \rho^\to_{i_0+2} = \cdots$.

Proof. We prove by contradiction that $\rho^\to_i = \rho^\to_{i+1}$ implies $\rho^\to_{i+1} = \rho^\to_{i+2}$ for every $i \geq 1$. For this we assume that $\rho^\to_i = \rho^\to_{i+1}$ and $\rho^\to_{i+1} \supset \rho^\to_{i+2}$ for some $i \geq 1$. Then there exist two states $s^\to, t^\to \in S^\to$ such that $s^\to \rho^\to_{i+1} t^\to$ but $s^\to \rho^\to_{i+2} t^\to$ does not hold. This means that there exists a symbol $a \in \Sigma$ such that $\delta^\to(s^\to, a)\rho^\to_{i+1}\delta^\to(t^\to, a)$ does not hold. As $\rho^\to_i = \rho^\to_{i+1}$, we obtain that $\delta^\to(s^\to, a)\rho^\to_i \delta^\to(t^\to, a)$ does not hold either. Hence $s^\to \rho^\to_{i+1} t^\to$ does not hold either. This contradicts our assumption $\rho^\to_i = \rho^\to_{i+1}$. □

Claim 2. For all $l \geq 1$, $s^\to, t^\to \in S^\to$, if $s^\to \rho^\to_l t^\to$, then for each $s^\leftarrow \in S^\leftarrow$ and $w \in \Sigma^*$ with $|w| = l$, we have $\|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(w) = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(w)$.

Proof. We proceed by induction on l. If $l = 1$, then $w = a$ for some $a \in \Sigma$ and hence $\|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(w) = f(s^\to, a, s^\leftarrow) = f(t^\to, a, s^\leftarrow) = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(w)$.

Now assume that the claim holds for $l \geq 1$. Let $w = av \in \Sigma^*$ such that $|v| = l$ (that is, $|w| = l + 1$). Then, by the definition of $\rho^\to_{l+1}$, we have $f(s^\to, a, s^\leftarrow v^{-1}) = f(t^\to, a, s^\leftarrow v^{-1})$ and $(s^\to a)\rho^\to_l (t^\to a)$. From this, by the induction hypothesis, $\|\mathcal{B}\|_{(s^\to a, s^\leftarrow)}(v) = \|\mathcal{B}\|_{(t^\to a, s^\leftarrow)}(v)$ for all $s^\leftarrow \in S^\leftarrow$. Thus

$$\|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(w) = f(s^\to, a, s^\leftarrow v^{-1})\|\mathcal{B}\|_{(s^\to a, s^\leftarrow)}(v) =$$
$$f(t^\to, a, s^\leftarrow v^{-1})\|\mathcal{B}\|_{(t^\to a, s^\leftarrow)}(v) = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(w)$$

for all $s^\leftarrow \in S^\leftarrow$. □

Claim 3. $\rho^\to_{i_0} \subseteq \rho^\to_{\mathcal{B}}$.

Proof. Assume that $s^\to \rho^\to_{i_0} t^\to$. Observe that $\|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(\varepsilon) = \varepsilon = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(\varepsilon)$. Let $w \in \Sigma^*$ with $|w| = l \geq 1$ be arbitrary. By the definition of i_0 and Claim 1, $\rho^\to_{i_0} \subseteq \rho^\to_l$. Consequently, we also have $s^\to \rho^\to_l t^\to$, from which we obtain by Claim 2 that $\|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(w) = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(w)$. Hence $s^\to \rho^\to_{\mathcal{B}} t^\to$. □

Claim 4. $\rho^\to_{i_0} \supseteq \rho^\to_{\mathcal{B}}$.

Proof. It suffices to show that, for all $s^\to, t^\to \in S^\to$, if for each $s^\leftarrow \in S^\leftarrow$ we have $\|\mathcal{B}\|_{(s^\to, s^\leftarrow)} = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}$, then $s^\to \rho^\to_i t^\to$ for all $i \geq 1$.

We proceed by induction on i. Let $i = 1$ and $a \in \Sigma$. By our assumption, $f(s^\to, a, s^\leftarrow) = \|\mathcal{B}\|_{(s^\to, s^\leftarrow)}(a) = \|\mathcal{B}\|_{(t^\to, s^\leftarrow)}(a) = f(t^\to, a, s^\leftarrow)$. Consequently, $s^\to \rho^\to_1 t^\to$.

Now assume that the claim holds for $i \geq 1$, i.e., $s^{\rightarrow}\rho_i t^{\rightarrow}$. Let $s^{\leftarrow} \in S^{\leftarrow}$, $a \in \Sigma$ and $v \in \Sigma^*$ be arbitrary. Then $f(s^{\rightarrow}, a, s^{\leftarrow}v^{-1})\|\mathcal{B}\|_{(s^{\rightarrow}a,s^{\leftarrow})}(v) = \|\mathcal{B}\|_{(s^{\rightarrow},s^{\leftarrow})}(av) = \|\mathcal{B}\|_{(t^{\rightarrow},s^{\leftarrow})}(av) = f(t^{\rightarrow}, a, s^{\leftarrow}v^{-1})\|\mathcal{B}\|_{(s^{\rightarrow}a,s^{\leftarrow})}(v)$. Hence $f(s^{\rightarrow}, a, s^{\leftarrow}v^{-1}) = f(t^{\rightarrow}, a, s^{\leftarrow}v^{-1})$ and $\|\mathcal{B}\|_{(s^{\rightarrow}a,s^{\leftarrow})}(v) = \|\mathcal{B}\|_{(t^{\rightarrow}a,s^{\leftarrow})}(v)$. From the latter, we have $\|\mathcal{B}\|_{(s^{\rightarrow}a,s^{\leftarrow})} = \|\mathcal{B}\|_{(t^{\rightarrow}a,s^{\leftarrow})}$, thus by the induction hypothesis, $(s^{\rightarrow}a)\rho_i^{\rightarrow}(t^{\rightarrow}a)$. Then, by the definition of $\rho_{i+1}^{\rightarrow}$, we obtain $s^{\rightarrow}\rho_{i+1}^{\rightarrow}t^{\rightarrow}$ holds as well. □

Lemma 9. We have $\rho_{i_0}^{\rightarrow} = \rho_{\mathcal{B}}^{\rightarrow}$.

Proof. It follows from Claims 3 and 4. □

Analogously, we can define a decreasing sequence

$$\rho_1^{\leftarrow} \supseteq \rho_2^{\leftarrow} \supseteq \cdots$$

of relations over $S^{\leftarrow}$ such that $\rho_{\mathcal{B}}^{\leftarrow} = \rho_{i_0}^{\leftarrow}$ for the least integer i_0 with $\rho_{i_0}^{\leftarrow} = \rho_{i_0+1}^{\leftarrow}$. Hence we can conclude the following.

Proposition 3. There is a polynomial time algorithm which constructs the minimal bimachine which is equivalent to $\mathcal{B}$.

Proof. By Proposition 2 we compute the connected part $\mathcal{B}^c$ of $\mathcal{B}$ in polynomial time. So assume that $\mathcal{B}$ is connected. We compute $\rho_{\mathcal{B}}^{\rightarrow}$ as follows. We compute $\rho_1^{\rightarrow}$ in $\mathcal{O}(|S^{\rightarrow}|^2|\Sigma||S^{\leftarrow}|)$ time and, for every $1 < i \leq i_0$, we compute $\rho_i^{\rightarrow}$ in $\mathcal{O}(|S^{\rightarrow}|^2|\Sigma|^2)$ time. Since $i_0 \leq |S^{\rightarrow}|$, we compute $\rho_{i_0}^{\rightarrow}$ in $\mathcal{O}(|S^{\rightarrow}|^3|\Sigma|^2)$ time. Analogously, we compute $\rho_{\mathcal{B}}^{\leftarrow}$ in polynomial time. □

4 Deterministic top-down tree automata and their minimization

In this section first we recall the concept of a deterministic top-down tree automaton (DTTA for short) from [4]. Then we give a polynomial time algorithm minimizing a state-separated DTTA. The size of a DTTA is the sum of the sizes of the bimachines associated with its states, hence we minimize it by minimizing the number of its states and the number of the states of the bimachines associated with its states.

4.1 Basic concepts

A *deterministic top-down tree automaton* (DTTA for short) is a system

$$\mathcal{A} = \big(Q, \Sigma, f_{\text{in}}, (\mathcal{B}_q \mid q \in Q), F\big),$$

where

- Q is a finite set (*states*),
- Σ is an alphabet (*input alphabet*),
- $f_{\text{in}} : \Sigma \rightarrow Q$ is the *initial function*,

- $\mathcal{B}_q = (\Sigma, Q, \mathcal{S}_q^{\rightarrow}, \mathcal{S}_q^{\leftarrow}, f_q)$ is a bimachine for every $q \in Q$ with semi-automata $\mathcal{S}_q^{\rightarrow} = (S_q^{\rightarrow}, \Sigma, s_{q,0}^{\rightarrow}, \delta_q^{\rightarrow})$ and $\mathcal{S}_q^{\leftarrow} = (S_q^{\leftarrow}, \Sigma, s_{q,0}^{\leftarrow}, \delta_q^{\leftarrow})$, and
- $F \subseteq Q$ (*final states*).

Let $\xi \in T_\Sigma$ and $q \in Q$. A *q-run of $\mathcal{A}$ on ξ* is a mapping $r : \mathrm{dom}(\xi) \to Q$ such that $r(\varepsilon) = q$ and for each node $x \in \mathrm{dom}(\xi)$ with $k > 0$ successors $x1, x2, \ldots, xk$, we have

$$r(x1)r(x2)\cdots r(xk) = \|\mathcal{B}_{r(x)}\|\big(\xi(x1)\cdots\xi(xk)\big).$$

Note that for each $\xi \in T_\Sigma$ and $q \in Q$, there is exactly one q-run of $\mathcal{A}$ on ξ. This q-run r is *accepting* if it assigns to each leaf a final state, that is, $r(x) \in F$ for every $x \in \mathrm{dom}(\xi)$ which is a leaf. The *tree language $L(\mathcal{A}, q)$ accepted by $\mathcal{A}$ in q* consists of all trees ξ such that the q-run of $\mathcal{A}$ on ξ is accepting. The $f_{\mathrm{in}}(\xi(\varepsilon))$-run of $\mathcal{A}$ on ξ is called *the run of $\mathcal{A}$ on ξ* and the *tree language $L(\mathcal{A})$ accepted by $\mathcal{A}$* consists of all trees ξ such that the run of $\mathcal{A}$ on ξ is accepting.

Two DTTA $\mathcal{A}$ and $\mathcal{A}'$ are *equivalent* if $L(\mathcal{A}) = L(\mathcal{A}')$.

Remark 1. We note that the root $\xi(\varepsilon)$ of ξ does not play any role in (accepting) q-runs on ξ. Hence if $\xi \in L(\mathcal{A}, q)$, then $\xi' \in L(\mathcal{A}, q)$ for each tree ξ' obtained by replacing the root of ξ with an arbitrary $a \in \Sigma$.

A state $q \in Q$ is called a *$\emptyset$-state* if $L(\mathcal{A}, q) = \emptyset$. We write $Q = Q_+ \cup Q_e$, where Q_e is the set of all $\emptyset$-states and $Q_+ = Q \setminus Q_e$. Note that $F \subseteq Q_+$.

Lemma 10. The set Q_+ is effectively computable.

Proof. We define a sequence $Q_0, Q_1, \ldots$ of sets of states by the following algorithm:
(i) Let $Q_0 = F$ and $i = 0$.
(ii) Let $Q_{i+1} = Q_i \cup \{q \in Q \mid \exists(x \in \Sigma^*) : \|\mathcal{B}_q\|(x) \in Q_i^*\}$.
(iii) If $Q_{i+1} = Q_i$, then stop, otherwise $i := i + 1$ and goto (ii).

First we note that for every $i \geq 0$ and $q \in Q$ we can decide whether there is an $x \in \Sigma^*$ with $\|\mathcal{B}_q\|(x) \in Q_i^*$. In fact, it suffices to check if $\|\mathcal{B}_q\|(x) \in Q_i^*$ for input strings x with $|x| \leq N_q$, where N_q is the number provided by Lemma 1 for the bimachine $\mathcal{B}_q$. Hence Q_{i+1} in step (ii) can be computed.

By standard arguments, we can prove the following statements:
- there is an $i \geq 0$ such that $Q_{i+1} = Q_i$,
- if $Q_{i+1} = Q_i$, then $Q_{i+j} = Q_i$ for every $j \geq 1$, and
- if $Q_{i+1} = Q_i$, then $\forall(q \in Q) : \big(q \in Q_i \iff \exists(\xi \in T_\Sigma) : \xi \in L(\mathcal{A}, q)\big)$.

Altogether we obtain that the algorithm terminates with $Q_{i+1} = Q_i$ and in this case $Q_+ = Q_i$. □

Next we introduce the concept of a connected DTTA. For this we define the binary relation $\to_\mathcal{A}$ over Q as follows: for every $q, q' \in Q$, we have $q \to_\mathcal{A} q'$ if there are $k \geq 1$, $a_1 \ldots a_k \in \Sigma^*$ such that $\|\mathcal{B}_q\|(a_1 \ldots a_k) = q_1 \ldots q_k$ and $q' = q_i$ for some $1 \leq i \leq k$. For every $q \in Q$, we define

$$T_q = \{q' \in Q \mid q \to_\mathcal{A}^* q'\}.$$

The DTTA $\mathcal{A}$ is *connected* if, for every $q \in Q$, we have $f_{\mathrm{in}}(a) \to_\mathcal{A}^* q$ for some $a \in \Sigma$.

Proposition 4. There is a polynomial time algorithm which computes T_q for a given state $q \in Q$.

Proof. By Proposition 2 we may assume that $\mathcal{B}_p$ is connected for every $p \in Q$.
(i) Let $T_0 = \{q\}$ and $i = 0$.
(ii) Let

$$T_{i+1} = T_i \cup \{f_p(s^{\rightarrow}, a, s^{\leftarrow}) \mid a \in \Sigma \text{ and } \exists(p \in T_i) : s^{\rightarrow} \in S_p^{\rightarrow}, s^{\leftarrow} \in S_p^{\leftarrow}\}.$$

(iii) If $T_{i+1} = T_i$, then stop, otherwise let $i := i + 1$ and goto (ii).
It is an exercise to show that $T_{i+1} = T_i$ for some $i \geq 0$ and for this i we have $T_q = T_i$. The algorithm runs in $\mathcal{O}(|Q|N^{\rightarrow}|\Sigma|N^{\leftarrow})$ time, where $N^{\rightarrow} = \max\{|S_p^{\rightarrow}| \mid p \in Q\}$ and $N^{\leftarrow} = \max\{|S_p^{\leftarrow}| \mid p \in Q\}$. □

For $\mathcal{A}$ we define the DTTA $\mathcal{A}^c = \big(Q^c, \Sigma, f_{\text{in}}^c, (\mathcal{B}_q \mid q \in Q^c), F^c\big)$ called the connected part of $\mathcal{A}$ as follows:
- $Q^c = \bigcup(T_q \mid q = f_{\text{in}}(a)$ for some $a \in \Sigma)$,
- $f_{\text{in}}^c(a) = f_{\text{in}}(a)$ for every $a \in \Sigma$, and
- $F^c = F \cap Q^c$.

The following statement is obvious.

Proposition 5. $\mathcal{A}^c$ is connected and is equivalent to $\mathcal{A}$.

By the definition of $\mathcal{A}^c$ and Proposition 4, we have the following result.

Proposition 6. There is a polynomial time algorithm which constructs $\mathcal{A}^c$.

A *congruence of* $\mathcal{A}$ is an equivalence relation $\tau \subseteq Q \times Q$ satisfying the following two conditions:

(i) for all states $p, q \in Q$, and nonempty word $a_1 \dots a_k \in \Sigma^*$, if $p\tau q$, $\|\mathcal{B}_p\|(a_1 \dots a_k) = p_1 \dots p_k$, and $\|\mathcal{B}_q\|(a_1 \dots a_k) = q_1 \dots q_k$, then $p_i \tau q_i$ for all $1 \leq i \leq k$,

(ii) if $p\tau q$, then $p \in F$ if and only if $q \in F$.

Let τ be an equivalence relation on Q. For every $q \in Q$, we introduce the bimachine

$$\mathcal{B}_{q,\tau} = (\Sigma, Q/\tau, \mathcal{S}_q^{\rightarrow}, \mathcal{S}_q^{\leftarrow}, f_{q,\tau}),$$

where $f_{q,\tau}(s^{\rightarrow}, a, s^{\leftarrow}) = f_q(s^{\rightarrow}, a, s^{\leftarrow})/\tau$ for all $s^{\rightarrow} \in S_q^{\rightarrow}, s^{\leftarrow} \in S_q^{\leftarrow}$, and $a \in \Sigma$. Then, for every $a_1 \dots a_k \in \Sigma^+$, we have $\|\mathcal{B}_{q,\tau}\|(a_1 \dots a_k) = p_1/\tau \dots p_k/\tau$, where $\|\mathcal{B}_q\|(a_1 \dots a_k) = p_1 \dots p_k$.

Lemma 11. Let τ be a congruence on $\mathcal{A}$ and $p, q \in Q$ such that $p\tau q$. Then $\|\mathcal{B}_{p,\tau}\| = \|\mathcal{B}_{q,\tau}\|$.

Proof. By definition $\|\mathcal{B}_{p,\tau}\|(\varepsilon) = \varepsilon = \|\mathcal{B}_{q,\tau}\|(\varepsilon)$.

Let $k \geq 1$ and $a_1 \dots a_k \in \Sigma^+$. Then we have $\|\mathcal{B}_{p,\tau}\|(a_1 \dots a_k) = p_1/\tau \dots p_k/\tau$, where $\|\mathcal{B}_p\|(a_1 \dots a_k) = p_1 \dots p_k$ and $\|\mathcal{B}_{q,\tau}\|(a_1 \dots a_k) = q_1/\tau \dots q_k/\tau$, where $\|\mathcal{B}_q\|(a_1 \dots a_k) = q_1 \dots q_k$. By (i) in the definition of a congruence of a DTTA, we have $p_i \tau q_i$ for all $1 \leq i \leq k$. Hence $p_1/\tau \dots p_k/\tau = q_1/\tau \dots q_k/\tau$. Thus $\|\mathcal{B}_{p,\tau}\|(a_1 \dots a_k) = \|\mathcal{B}_{q,\tau}\|(a_1 \dots a_k)$. □

Given a congruence τ of $\mathcal{A}$, we define the *factor DTTA $\mathcal{A}/\tau$ of $\mathcal{A}$ determined by τ* as $\mathcal{A}/\tau = \big(Q/\tau, \Sigma, f_{\text{in},\tau}, (\mathcal{B}_{q/\tau} \mid q/\tau \in Q/\tau), F/\tau\big)$, where

- $f_{\text{in},\tau}(a) = (f_{\text{in}}(a))/\tau$ for every $a \in \Sigma$,
- $\mathcal{B}_{q/\tau} = \mathcal{B}_{q,\tau}$ for every $q \in Q$.

We note that the definition of the bimachine $\mathcal{B}_{q/\tau}$ and hence that of the DTTA $\mathcal{A}/\tau$ is syntactically ambiguous. Indeed, for $p/\tau = q/\tau$, the bimachines $\mathcal{B}_{p,\tau}$ and $\mathcal{B}_{q,\tau}$ may be different syntactically and we can pick any of them. However, our choice has no impact on $\|\mathcal{A}/\tau\|$ because, by Lemma 11, $p\tau q$ implies $\|\mathcal{B}_{p,\tau}\| = \|\mathcal{B}_{q,\tau}\|$. In other words, $\|\mathcal{A}/\tau\|$ is well-defined.

Throughout the paper $\mathcal{A}$ and $\mathcal{A}'$ will denote the DTTA

- $\mathcal{A} = \big(Q, \Sigma, f_{\text{in}}, (\mathcal{B}_q \mid q \in Q), F\big)$ *with bimachines* $\mathcal{B}_q = (\Sigma, Q, \mathcal{S}_q^{\rightarrow}, \mathcal{S}_q^{\leftarrow}, f_q)$ *and semi-automata* $\mathcal{S}_q^{\rightarrow} = (S_q^{\rightarrow}, \Sigma, s_{q,0}^{\rightarrow}, \delta_q^{\rightarrow})$ *and* $\mathcal{S}_q^{\leftarrow} = (S_q^{\leftarrow}, \Sigma, s_{q,0}^{\leftarrow}, \delta_q^{\leftarrow})$ *for every $q \in Q$, and*
- $\mathcal{A}' = \big(Q', \Sigma, f'_{\text{in}}, (\mathcal{B}'_q \mid q \in Q'), F'\big)$ *with bimachines* $\mathcal{B}'_q = (\Sigma, Q', \mathcal{T}_q^{\rightarrow}, \mathcal{T}_q^{\leftarrow}, f'_q)$ *and semi-automata* $\mathcal{T}_q^{\rightarrow} = (T_q^{\rightarrow}, \Sigma, t_{q,0}^{\rightarrow}, \gamma_q^{\rightarrow})$ *and* $\mathcal{T}_q^{\leftarrow} = (T_q^{\leftarrow}, \Sigma, t_{q,0}^{\leftarrow}, \gamma_q^{\leftarrow})$ *for every $q \in Q'$,*

respectively.

Furthermore, let $\varphi : Q \to Q'$ be a mapping and $\varphi^* : Q^* \to Q'^*$ its unique extension to a monoid homomorphism. The mapping φ is a *homomorphism from $\mathcal{A}$ to $\mathcal{A}'$* if

- $f'_{\text{in}} = f_{\text{in}} \circ \varphi$,
- $\|\mathcal{B}'_{\varphi(q)}\| = \|\mathcal{B}_q\| \circ \varphi^*$ for every $q \in Q$, and
- $q \in F \iff \varphi(q) \in F'$ for every $q \in Q$.

If φ is a surjective homomorphism, then $\mathcal{A}'$ is a *homomorphic image* of $\mathcal{A}$. If, in addition, φ is a bijection, then we say that $\mathcal{A}$ and $\mathcal{A}'$ are *isomorphic* and write $\mathcal{A} \cong \mathcal{A}'$.

Lemma 12. If there is a homomorphism φ from $\mathcal{A}$ to $\mathcal{A}'$, then
(i) $L(\mathcal{A}, q) = L(\mathcal{A}', \varphi(q))$ for every $q \in Q$, and
(ii) $L(\mathcal{A}) = L(\mathcal{A}')$.

Proof. Let φ be a *homomorphism from $\mathcal{A}$ to $\mathcal{A}'$*. To show (i), we prove by induction on height(ξ) that for any $q \in Q$ and $\xi \in T_\Sigma$, $\xi \in L(\mathcal{A}, q)$ if and only if $\xi \in L(\mathcal{A}', \varphi(q))$.

Base of induction: $\text{height}(\xi) = 0$, i.e., $\xi = a$ for some $a \in \Sigma$. Then $\xi \in L(\mathcal{A}, q)$ if and only if $\xi \in L(\mathcal{A}', \varphi(q))$. Thus the statement holds obviously.

Induction step: $\text{height}(\xi) = n > 0$. Then $\xi = a(\xi_1, \dots, \xi_k)$ for some $a \in \Sigma$, $k \geq 1$, and $\xi_1, \dots, \xi_k \in T_\Sigma$. Let $a_i = \xi(i)$ for all $1 \leq i \leq k$. Then we have

$$\begin{array}{ll} & \xi \in L(\mathcal{A}, q) \\ \iff & \|\mathcal{B}_q\|(a_1...a_k) = q_1 \dots q_k \text{ and} \\ & \xi_i \in L(\mathcal{A}, q_i) \text{ for all } 1 \leq i \leq k \\ \iff & \|\mathcal{B}'_{\varphi(q)}\|(a_1...a_k) = \varphi(q_1) \dots \varphi(q_k) \text{ and} \\ & \xi_i \in L(\mathcal{A}', \varphi(q_i)) \text{ for all } 1 \leq i \leq k \\ \iff & \xi \in L(\mathcal{A}', \varphi(q)). \end{array}$$

We now show (ii). Let $\xi \in L(\mathcal{A})$, i.e., $\xi \in L(\mathcal{A}, f_{\text{in}}(\xi(\varepsilon))$. Then by (i), $\xi \in L(\mathcal{A}', \varphi(f_{\text{in}}(\xi(\varepsilon)))$. As φ is a homomorphism, $f'_{\text{in}}(\xi(\varepsilon)) = \varphi(f_{\text{in}}(\xi(\varepsilon)))$. Thus $\xi \in L(\mathcal{A}', f'_{\text{in}}(\xi(\varepsilon))$, which implies $\xi \in L(\mathcal{A}')$.

Conversely, let $\xi \in L(\mathcal{A}')$, i.e., let $\xi \in L(\mathcal{A}', f'_{\text{in}}(\xi(\varepsilon))$. As φ is a homomorphism, $\varphi(f_{\text{in}}(\xi(\varepsilon))) = f'_{\text{in}}(\xi(\varepsilon))$. Then by (i), $\xi \in L(\mathcal{A}, f_{\text{in}}(\xi(\varepsilon))$ which proves that $\xi \in L(\mathcal{A})$. □

Lemma 13. If τ is a congruence of $\mathcal{A}$, then $\mathcal{A}/\tau$ is a homomorphic image of $\mathcal{A}$.

Proof. It is easy to check that the mapping $\varphi : Q \to Q/\tau$ defined by $\varphi(q) = q/\tau$ is a surjective homomorphism from $\mathcal{A}$ to $\mathcal{A}/\tau$. □

Lemma 14. If τ is a congruence of $\mathcal{A}$, then $L(\mathcal{A}, q) = L(\mathcal{A}/\tau, q/\tau)$ for every $q \in Q$. Moreover, $L(\mathcal{A}) = L(\mathcal{A}/\tau)$.

Proof. It follows from Lemmas 12 and 13. □

4.2 Minimization of DTTA

The DTTA $\mathcal{A}$ is called *minimal* if

$$|Q| \leq |Q'|, \sum_{q \in Q} |S_q^{\rightarrow}| \leq \sum_{q \in Q'} |T_q^{\rightarrow}|, \text{ and } \sum_{q \in Q} |S_q^{\leftarrow}| \leq \sum_{q \in Q'} |T_q^{\leftarrow}|$$

for any DTTA $\mathcal{A}'$ which is equivalent to $\mathcal{A}$. Moreover, $\mathcal{A}$ is *state-separated* if

- $\|\mathcal{B}_q\|(x) \in Q_+^* \cup Q_e^*$ for every $q \in Q_+$ and
- $\|\mathcal{B}_q\|(x) \in Q_e^*$ for every $q \in Q_e$

for every $x \in \Sigma^*$.

Lemma 15. For the DTTA $\mathcal{A}$ the following two statements are equivalent.
(i) $\mathcal{A}$ is state-separated.
(ii) If $\|\mathcal{B}_q\|(x) \in Q^* Q_e Q^*$, then $\|\mathcal{B}_q\|(x) \in Q_e^*$ for every $q \in Q$ and $x \in \Sigma^*$.

Proof. It is clear that (i) implies (ii). Now assume that (ii) holds. Let $x \in \Sigma^*$ and $q \in Q$. If $q \in Q_e$, then obviously $\|\mathcal{B}_q\|(x) \in Q^* Q_e Q^*$. Hence by (ii), $\|\mathcal{B}_q\|(x) \in Q_e^*$. Now let $q \in Q_+$. If $\|\mathcal{B}_q\|(x) \in Q^* Q_e Q^*$, then by (ii), $\|\mathcal{B}_q\|(x) \in Q_e^*$. Otherwise, $\|\mathcal{B}_q\|(x) \in Q_+^*$. Hence (i) holds. □

Lemma 16. The DTTA $\mathcal{A}$ is state-separated if and only if for all state $q \in Q$, reachable states $s^\rightarrow \in \mathcal{S}_q^\rightarrow$ and $s^\leftarrow \in \mathcal{S}_q^\leftarrow$, and $a, b \in \Sigma$,
$f_q(s^\rightarrow, a, \delta_q^\leftarrow(s^\leftarrow, b)) \in Q_e$ if and only if $f_q(\delta_q^\rightarrow(s^\rightarrow, a), b, s^\leftarrow) \in Q_e$.

Proof. ($\Rightarrow$) Assume that $\mathcal{A}$ is state-separated and let $q \in Q$. Moreover, let $s^\rightarrow \in \mathcal{S}_q^\rightarrow$ and $s^\leftarrow \in \mathcal{S}_q^\leftarrow$ be reachable states, and $a, b \in \Sigma$. Then there are $j \geq 0$ and $a_1 \ldots a_j \in \Sigma^*$ such that $s_{q,0}^\rightarrow a_1 \ldots a_j = s^\rightarrow$, and there are $k \geq j+3$ and $a_{j+3} \ldots a_k \in \Sigma^*$ such that $s_{q,0}^\leftarrow a_k \ldots a_{j+3} = s^\leftarrow$. Let $a_{j+1} = a$ and $a_{j+2} = b$, and $x = a_1 \ldots a_k$.

Then $\|\mathcal{B}_q\|_{(s_{q,0}^\rightarrow, s_{q,0}^\leftarrow)}(x) = q_1 \ldots q_k$, where $q_1, \ldots, q_k$ are obtained as follows. Let

- $t_0^\rightarrow t_1^\rightarrow \ldots t_{k-1}^\rightarrow t_k^\rightarrow$ be the $s_{q,0}^\rightarrow$-run of $\mathcal{S}_q^\rightarrow$ on $a_1 \ldots a_k$,
- $t_0^\leftarrow t_1^\leftarrow \ldots t_{k-1}^\leftarrow t_k^\leftarrow$ the $s_{q,0}^\leftarrow$-run of $\mathcal{S}_q^\leftarrow$ on the reversed input $a_k \ldots a_1$, and
- let $q_i = f_q(t_{i-1}^\rightarrow, a_i, t_{k-i}^\leftarrow)$ for $1 \leq i \leq k$.

Here $t_j^\rightarrow = s^\rightarrow$, $t_{j+1}^\rightarrow = \delta_q^\rightarrow(s^\rightarrow, a)$, $t_{k-j-2}^\leftarrow = s^\leftarrow$, $t_{k-j-1}^\leftarrow = \delta_q^\leftarrow(s^\leftarrow, b)$, $f_q(s^\rightarrow, a, \delta_q^\leftarrow(s^\leftarrow, b)) = q_j$, and $f_q(\delta_q^\rightarrow(s^\rightarrow, a), b, s^\leftarrow) = q_{j+1}$. If $q_{j+1} \in Q_e$, then by Lemma 15, $\|\mathcal{B}_q\|(x) \in Q_e^*$. Therefore $q_{j+2} \in Q_e$. Conversely, if $q_{j+2} \in Q_e$, then by Lemma 15, $q_{j+1} \in Q_e$.

($\Leftarrow$) By Lemma 15 it is sufficient to show that $\|\mathcal{B}_q\|(x) \in Q^* Q_e Q^*$ implies $\|\mathcal{B}_q\|(x) \in Q_e^*$ for every $q \in Q$ and $x \in \Sigma^*$.

Let $x = a_1 \ldots a_k$, $k \geq 1$, be arbitrary, and let $\|\mathcal{B}_q\|_{(s_{q,0}^\rightarrow, s_{q,0}^\leftarrow)}(x)$ be as in the first part of the proof. Assume that $q_i \in Q_e$ for some $1 \leq i \leq k$. If $i < k$, then by our assumption, $q_{i+1} = q_e$ as well. Iterating this reasoning, we get that $q_j = q_e$ for each $i \leq j \leq k$. If $i > 1$, then by our assumption, $q_{i-1} = q_e$ as well. As before, we get that $q_j = q_e$ for each $1 \leq j \leq i$. Hence $q_i = q_e$ for each $1 \leq i \leq k$. □

Lemma 17. It is decidable whether $\mathcal{A}$ is state-separated or not.

Proof. The sets Q_+ and $Q_e = Q \setminus Q_+$ are effectively computable (cf. Lemma 10). Then, by direct inspection of $\mathcal{A}$, we can decide whether the condition of Lemma 16 holds. □

> *In the rest of this section we assume that $\mathcal{A}$ and $\mathcal{A}'$ are state-separated with $Q = Q_+ \cup Q_e$ and $Q' = Q'_+ \cup Q'_e$, respectively. In fact, our minimization algorithm works only for state-separated DTTA.*

We introduce the equivalence relation $\tau_\mathcal{A} \subseteq Q \times Q$ as follows: for all $p, q \in Q$,

let $p \tau_\mathcal{A} q$ if and only if $L(\mathcal{A}, p) = L(\mathcal{A}, q)$.

The DTTA $\mathcal{A}$ is *reduced* if $\tau_\mathcal{A}$ is the identity relation.

Lemma 18. Let $q \in Q$ and $q' \in Q'$ such that $L(\mathcal{A}, q) = L(\mathcal{A}', q')$. Moreover, let $k \geq 1$, $a_1 \ldots a_k \in \Sigma^*$, and let $\|\mathcal{B}_q\|(a_1 \ldots a_k) = q_1 \ldots q_k$ and $\|\mathcal{B}_{q'}\|(a_1 \ldots a_k) = q'_1 \ldots q'_k$. Then $L(\mathcal{A}, q_i) = L(\mathcal{A}', q'_i)$ for all $i = 1, \ldots, k$.

Proof. Since $L(\mathcal{A}, q) = L(\mathcal{A}', q')$, we have either (1) $q \in Q_e$ and $q' \in Q'_e$ or (2) $q \in Q_+$ and $q' \in Q'_+$. Let us recall that $\mathcal{A}$ and $\mathcal{A}'$ are state-separated.

In case (1) we have $q_1 \dots q_k \in Q_e^*$ and $q'_1 \dots q'_k \in Q_e'^*$, hence the statement holds.

In case (2) either (2a) $q_1 \dots q_k \in Q_e^*$ and $q'_1 \dots q'_k \in Q_e'^*$ or (2b) $q_1 \dots q_k \in Q_+^*$ and $q'_1 \dots q'_k \in Q_+'^*$. (The other two cases are excluded because $L(\mathcal{A}, q) = L(\mathcal{A}', q')$.)

In case (2a) the statement again holds, so let us assume that (2b) holds. Arguing by contradiction, assume that $L(\mathcal{A}, q_i) \neq L(\mathcal{A}', q'_i)$ for some $1 \leq i \leq k$. Then there exists a tree $\xi \in (L(\mathcal{A}, q_i) \setminus L(\mathcal{A}', q'_i)) \cup (L(\mathcal{A}', q'_i) \setminus (L(\mathcal{A}, q_i))$ and there are trees $\eta_j \in L(\mathcal{A}, q_j)$ and $\theta_j \in L(\mathcal{A}', q'_j)$ for each $j = 1, \dots, i-1, i+1, \dots k$. Hence $a(\eta_1, \dots, \eta_{i-1}, \xi, \eta_{i+1}, \dots, \eta_k) \in (L(\mathcal{A}, q) \setminus L(\mathcal{A}', q'))$ or $a(\theta_1, \dots, \theta_{i-1}, \xi, \theta_{i+1}, \dots, \theta_k) \in (L(\mathcal{A}', q') \setminus (L(\mathcal{A}, q))$. Thus $L(\mathcal{A}, q) \neq L(\mathcal{A}', q')$, which is a contradiction. □

Lemma 19. The relation $\tau_{\mathcal{A}}$ is a congruence of $\mathcal{A}$.

Proof. Let $p, q \in Q$ such that $p \tau_{\mathcal{A}} q$. For showing property (i), let $a_1 \dots a_k \in \Sigma^*$ with $\|\mathcal{B}_p\|(a_1 \dots a_k) = p_1 \dots p_k$ and $\|\mathcal{B}_q\|(a_1 \dots a_k) = q_1 \dots q_k$. Then by Lemma 18 with $\mathcal{A} = \mathcal{A}'$, we have $L(\mathcal{A}, p_i) = L(\mathcal{A}, q_i)$ for all $i = 1, \dots, k$. Hence by the definition of $\tau_{\mathcal{A}}$, $p_i \tau_{\mathcal{A}} q_i$ for every $1 \leq i \leq k$.

Finally, we show that (ii) holds by contradiction as follows: if $p \in F$ and $q \notin F$, then $a \in (L(\mathcal{A}, p) \setminus L(\mathcal{A}, q))$ for every $a \in \Sigma$ which contradicts to $p \tau_{\mathcal{A}} q$. □

Lemma 20. The DTTA $\mathcal{A}/\tau_{\mathcal{A}}$ is reduced.

Proof. Assume that $L(\mathcal{A}/\tau_{\mathcal{A}}, p/\tau_{\mathcal{A}}) = L(\mathcal{A}/\tau_{\mathcal{A}}, q/\tau_{\mathcal{A}})$ for some $p, q \in Q$. Then by Lemma 14 and Lemma 19, $L(\mathcal{A}, p) = L(\mathcal{A}/\tau_{\mathcal{A}}, p/\tau_{\mathcal{A}}) = L(\mathcal{A}/\tau_{\mathcal{A}}, q/\tau_{\mathcal{A}}) = L(\mathcal{A}, q)$. Hence $p \tau_{\mathcal{A}} q$, i.e., $p/\tau_{\mathcal{A}} = q/\tau_{\mathcal{A}}$. □

Theorem 2. Assume that $\mathcal{A}$ and $\mathcal{A}'$ are connected and reduced. Then

$$L(\mathcal{A}) = L(\mathcal{A}') \text{ if and only if } \mathcal{A} \cong \mathcal{A}'.$$

Proof. We prove the implication from left to right, because the proof of the other direction is obvious. Assume that $L(\mathcal{A}) = L(\mathcal{A}')$. Let us define the relation $\varphi \subseteq Q \times Q'$ as follows: $\varphi = \{(q, q') \mid L(\mathcal{A}, q) = L(\mathcal{A}', q')\}$. For convenience, we divide the proof in five steps.

(i) We show that for each $q \in Q$, there exists $q' \in Q'$ such that $(q, q') \in \varphi$, i.e., the domain of φ is Q. As $\mathcal{A}$ is connected, we have

$$f_{\text{in}}(a) \to_{\mathcal{A}} q_1 \to_{\mathcal{A}} \cdots \to_{\mathcal{A}} q_n = q$$

for some $a \in \Sigma$, $n \geq 0$, and $q_1, \dots, q_n \in Q$. If $n = 0$, then $q = f_{\text{in}}(a)$. Since $L(\mathcal{A}) = L(\mathcal{A}')$, we have $L(\mathcal{A}, f_{\text{in}}(a)) = L(\mathcal{A}', f'_{\text{in}}(a))$, hence $(q, f'_{\text{in}}(a)) \in \varphi$. If $n \geq 1$, then by Lemma 18 there exists $q'_1, \dots, q'_n \in Q'$ such that

$$f'_{\text{in}}(a) \to_{\mathcal{A}'} q'_1 \to_{\mathcal{A}'} \cdots \to_{\mathcal{A}'} q'_n$$

and $L(\mathcal{A}, q_i) = L(\mathcal{A}', q'_i)$ for each $i = 1, \dots, n$. Thus $(q, q'_n) \in \varphi$.

(ii) We show that φ is a mapping. For any $q \in Q$ and $q'_1, q'_2 \in Q'$, if $(q, q'_1) \in \varphi$ and $(q, q'_2) \in \varphi$, then $L(\mathcal{A}', q'_1) = L(\mathcal{A}, q) = L(\mathcal{A}', q'_2)$, and hence $q'_1 = q'_2$.

(iii) We show that φ is injective. For any $q_1, q_2 \in Q$ and $q' \in Q'$, if $\varphi(q_1) = q'$ and $\varphi(q_2) = q'$, then $L(\mathcal{A}, q_1) = L(\mathcal{A}, q') = L(\mathcal{A}, q_2)$, and hence $q_1 = q_2$.

(iv) We show that φ is surjective. Repeating the argument used in (i) with the roles of $\mathcal{A}$ and $\mathcal{A}'$ reversed we see that for every $q' \in Q'$ there exists a $q \in Q$ such that $L(\mathcal{A}, q) = L(\mathcal{A}', q')$.

(v) We show that φ is a homomorphism.

First we show that $f'_{\text{in}} = f_{\text{in}} \circ \varphi$. As $L(\mathcal{A}) = L(\mathcal{A}')$, we have $L(\mathcal{A}, f_{\text{in}}(a)) = L(\mathcal{A}', f'_{\text{in}}(a))$ for each $a \in \Sigma$. Hence, by the definition of φ, $\varphi(f_{\text{in}}(a)) = f'_{\text{in}}(a)$ for each $a \in \Sigma$. Thus we have $f'_{\text{in}} = f_{\text{in}} \circ \varphi$.

Second, we show that $\|\mathcal{B}'_{\varphi(q)}\| = \|\mathcal{B}_q\| \circ \varphi^*$ for every $q \in Q$. Let $q \in Q$, $q' \in Q'$ and $a_1 \ldots a_k \in \Sigma^*$, $k \geq 1$, with $\|\mathcal{B}_q\|(a_1 \ldots a_k) = q_1 \ldots q_k$ and $\|\mathcal{B}_{q'}\|(a_1 \ldots a_k) = q'_1 \ldots q'_k$. Then by Lemma 18, $\varphi(q_i) = q'_i$ for each $i = 1, \ldots, k$. Hence $\|\mathcal{B}'_{\varphi(q)}\| = \|\mathcal{B}_q\| \circ \varphi^*$ for every $q \in Q$.

Third, we show that $q \in F \iff \varphi(q) \in F'$ for every $q \in Q$. We proceed by contradiction. Assume that $q \in F$ and $\varphi(q) \notin F$ for some $q \in Q$. Then for each $a \in \Sigma$, $a \in L(\mathcal{A}, q)$ and $a \notin L(\mathcal{A}, \varphi(q))$. This is a contradiction. The case $q \notin F$ and $\varphi(q) \in F$ is analogous to the previous case. Thus $\mathcal{A}$ and $\mathcal{A}'$ are isomorphic. □

By Theorem 2, we have the following result.

Corollary 1. Assume that $\mathcal{A}$ and $\mathcal{A}'$ are connected. Then $L(\mathcal{A}) = L(\mathcal{A}')$ if and only if $\mathcal{A}/\tau_{\mathcal{A}} \cong \mathcal{A}'/\tau_{\mathcal{A}'}$.

Proof. Assume that $L(\mathcal{A}) = L(\mathcal{A}')$. Then by Lemmas 14 and 19, we have $L(\mathcal{A}/\tau_{\mathcal{A}}) = L(\mathcal{A}) = L(\mathcal{A}') = L(\mathcal{A}'/\tau_{\mathcal{A}'})$. By Lemma 20, $\mathcal{A}/\tau_{\mathcal{A}}$ and $\mathcal{A}'/\tau_{\mathcal{A}'}$ are connected and reduced. Hence, by Theorem 2 we obtain $\mathcal{A}/\tau_{\mathcal{A}} \cong \mathcal{A}'/\tau_{\mathcal{A}'}$.

Conversely, assume that $\mathcal{A}/\tau_{\mathcal{A}} \cong \mathcal{A}'/\tau_{\mathcal{A}'}$. Then by Lemmas 14 and 19, we have $L(\mathcal{A}) = L(\mathcal{A}/\tau_{\mathcal{A}}) = L(\mathcal{A}'/\tau_{\mathcal{A}'}) = L(\mathcal{A}')$. □

Lemma 21. Let $\varphi : Q \to Q'$ be a homomorphism from $\mathcal{A}$ to $\mathcal{A}'$. Moreover, assume that $\mathcal{B}_q$ is connected for each $q \in Q$ and $\mathcal{B}'_{q'}$ is connected and reduced for each $q' \in Q'$. For every $q \in Q$ and $q' \in Q'$ with $\varphi(q) = q'$, the bimachine $\mathcal{B}'_{q'}$ is a homomorphic image of $\mathcal{B}_q$.

Proof. Let $q \in Q$ and $q' \in Q'$ with $\varphi(q) = q'$. First we show that $\mathcal{T}^{\rightarrow}_{q'}$ is a homomorphic image of $\mathcal{S}^{\rightarrow}_q$. For this, let us define the relation $\psi^{\rightarrow}_{q,q'} \subseteq S^{\rightarrow}_q \times T^{\rightarrow}_{q'}$ by

$$\psi^{\rightarrow}_{q,q'} = \{(s^{\rightarrow}_{q,0}x, t^{\rightarrow}_{q',0}x) \mid x \in \Sigma^*\}.$$

We note that the domain of $\psi^{\rightarrow}_{q,q'}$ is $S^{\rightarrow}_q$ because $\mathcal{B}_q$ is connected. Next we show by contradiction that $\psi^{\rightarrow}_{q,q'}$ is a mapping. For this, let us assume that there are $x, y \in \Sigma^*$ such that $s^{\rightarrow}_{q,0}x = s^{\rightarrow}_{q,0}y$ and $t^{\rightarrow}_{q',0}x \neq t^{\rightarrow}_{q',0}y$. Since $\mathcal{B}'_{q'}$ is reduced, there are $u, z \in \Sigma^*$ such that $\|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}x, t^{\leftarrow}_{q',0}z)}(u) \neq \|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}y, t^{\leftarrow}_{q',0}z)}(u)$, i.e.,

$$\|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}x, t^{\leftarrow}_{q',0})}(uz^{-1}) \neq \|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}y, t^{\leftarrow}_{q',0})}(uz^{-1}).$$

On the other hand, by $\|\mathcal{B}_q\| \circ \varphi^* = \|\mathcal{B}'_{q'}\|$, we have

$$\varphi^*(\|\mathcal{B}_q\|(xuz^{-1})) = \|\mathcal{B}'_{q'}\|(xuz^{-1}) \text{ and } \varphi^*(\|\mathcal{B}_q\|(yuz^{-1})) = \|\mathcal{B}'_{q'}\|(yuz^{-1}).$$

Thus

$$\varphi^*(\|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}x, s^{\leftarrow}_{q,0})}(uz^{-1})) = \|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}x, t^{\leftarrow}_{q',0})}(uz^{-1}) \text{ and}$$
$$\varphi^*(\|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}y, s^{\leftarrow}_{q,0})}(uz^{-1})) = \|\mathcal{B}'_{q'}\|_{(t^{\rightarrow}_{q',0}y, t^{\leftarrow}_{q',0})}(uz^{-1}).$$

Hence $\varphi^*(\|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}x, s^{\leftarrow}_{q,0})}(uz^{-1})) \neq \varphi^*(\|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}y, s^{\leftarrow}_{q,0})}(uz^{-1}))$ and thus $\|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}x, s^{\leftarrow}_{q,0})}(uz^{-1}) \neq \|\mathcal{B}_q\|_{(s^{\rightarrow}_{q,0}y, s^{\leftarrow}_{q,0})}(uz^{-1})$. This is a contradiction by our assumption $s^{\rightarrow}_{q,0}x = s^{\rightarrow}_{q,0}y$.

Since $\mathcal{B}'_{q'}$ is connected, the mapping $\psi^{\rightarrow}_{q,q'}$ is surjective. Finally we show that $\psi^{\rightarrow}_{q,q'}$ is a homomorphism. Obviously, $\psi^{\rightarrow}_{q,q'}(s^{\rightarrow}_{q,0}) = t^{\rightarrow}_{q',0}$. Moreover, for every $x \in \Sigma^*$ and $a \in \Sigma$, we have

$$\psi^{\rightarrow}_{q,q'}(\delta^{\rightarrow}_{q,0}(s^{\rightarrow}_{q,0}x, a)) = \psi^{\rightarrow}_{q,q'}(s^{\rightarrow}_{q,0}xa) = t^{\rightarrow}_{q',0}xa = \gamma^{\rightarrow}_{q',0}(t^{\rightarrow}_{q',0}x, a) = \gamma^{\rightarrow}_{q',0}(\psi^{\rightarrow}_{q,q'}(s^{\rightarrow}_{q,0}x), a).$$

Analogously, we can define the relation $\psi^{\leftarrow}_{q,q'} \subseteq S^{\leftarrow}_q \times T^{\leftarrow}_{q'}$ and show that it is a homomorphism from $\mathcal{S}^{\leftarrow}_q$ onto $\mathcal{T}^{\leftarrow}_{q'}$. Hence $\mathcal{B}'_{q'}$ is a homomorphic image of $\mathcal{B}_q$ via $(\psi^{\rightarrow}_{q,q'}, \psi^{\leftarrow}_{q,q'})$. □

Lemma 22. Assume that $\mathcal{A}'$ is a homomorphic image of $\mathcal{A}$, that $\mathcal{B}_q$ is connected for each $q \in Q$, and that $\mathcal{B}'_{q'}$ is connected and reduced for each $q' \in Q'$. Then

$$|Q'| \leq |Q|, \sum_{q \in Q'} |T^{\rightarrow}_q| \leq \sum_{q \in Q} |S^{\rightarrow}_q|, \text{ and } \sum_{q \in Q'} |T^{\leftarrow}_q| \leq \sum_{q \in Q} |S^{\leftarrow}_q|.$$

Proof. Let $\varphi : Q \to Q'$ be a surjective homomorphism from $\mathcal{A}$ to $\mathcal{A}'$. By Lemma 21, for every $q \in Q$, the bimachine $\mathcal{B}'_{\varphi(q)}$ is a homomorphic image of $\mathcal{B}_q$. Thus, by Lemma 3, $|T^{\rightarrow}_{\varphi(q)}| \leq |S^{\rightarrow}_q|$ and $|T^{\leftarrow}_{\varphi(q)}| \leq |S^{\leftarrow}_q|$ for every $q \in Q$. Consequently, as φ is a surjective mapping, the statement of the lemma holds. □

Lemma 23. Assume that $\mathcal{A}'$ is a homomorphic image of $\mathcal{A}$ and that $\mathcal{B}'_{q'}$ is connected and reduced for each $q' \in Q'$. Then

$$|Q'| \leq |Q|, \sum_{q \in Q'} |T^{\rightarrow}_q| \leq \sum_{q \in Q} |S^{\rightarrow}_q|, \text{ and } \sum_{q \in Q'} |T^{\leftarrow}_q| \leq \sum_{q \in Q} |S^{\leftarrow}_q|.$$

Proof. Let $\mathcal{B}^c_q$ be the connected part of $\mathcal{B}_q$ for each $q \in Q$. As mentioned, the bimachine $\mathcal{B}^c_q$ is equivalent to $\mathcal{B}_q$ for each $q \in Q$. Hence the DTTA $\big(Q, \Sigma, f, (\mathcal{B}^c_q \mid q \in Q), F\big)$ is equivalent to $\mathcal{A}$ and, obviously,

$$\sum_{q \in Q} |S^{\rightarrow c}_q| \leq \sum_{q \in Q} |S^{\rightarrow}_q|, \text{ and } \sum_{q \in Q} |S^{\leftarrow c}_q| \leq \sum_{q \in Q} |S^{\leftarrow}_q|.$$

Moreover, $\mathcal{A}'$ is a homomorphic image of $\big(Q, \Sigma, f, (\mathcal{B}^c_q \mid q \in Q), F\big)$. Hence by Lemma 22,

$$|Q'| \leq |Q|, \sum_{q \in Q'} |T^{\rightarrow}_q| \leq \sum_{q \in Q} |S^{\rightarrow c}_q|, \text{ and } \sum_{q \in Q'} |T^{\leftarrow}_q| \leq \sum_{q \in Q} |S^{\leftarrow c}_q|.$$

These and the above inequalities imply the lemma. □

Lemma 24. Assume that $\mathcal{A}$ is connected and consider $\mathcal{A}/\tau_{\mathcal{A}} = \big(Q/\tau_{\mathcal{A}}, \Sigma, f_{\text{in},\tau_{\mathcal{A}}}, (\mathcal{B}_{q/\tau_{\mathcal{A}}} \mid q/\tau_{\mathcal{A}} \in Q/\tau_{\mathcal{A}}), F/\tau_{\mathcal{A}}\big)$. For each $q/\tau_{\mathcal{A}} \in Q/\tau_{\mathcal{A}}$, let $\mathcal{B}^c_{q/\tau_{\mathcal{A}}}$ be the connected part of $\mathcal{B}_{q/\tau_{\mathcal{A}}}$ and let

$$\mathcal{M} = \big(Q/\tau_{\mathcal{A}}, \Sigma, f_{\text{in},\tau_{\mathcal{A}}}, (\mathcal{B}^c_{q/\tau_{\mathcal{A}}}/\rho_{\mathcal{B}^c_{q/\tau_{\mathcal{A}}}} \mid q/\tau_{\mathcal{A}} \in Q/\tau_{\mathcal{A}}), F/\tau_{\mathcal{A}}\big).$$

Then $\mathcal{M}$ is a minimal DTTA and equivalent to $\mathcal{A}$.

Proof. Let $L(\mathcal{A}) = L(\mathcal{A}')$. By Propositions 4 and 5, we may assume that $\mathcal{A}'$ is connected. Then, by Corollary 1, $\mathcal{A}'/\tau_{\mathcal{A}'} \cong \mathcal{A}/\tau_{\mathcal{A}}$. Hence, by Lemmas 4 and 19, there is a surjective homomorphism $\varphi : Q' \to Q/\tau_{\mathcal{A}}$ from $\mathcal{A}'$ to $\mathcal{A}/\tau_{\mathcal{A}}$. Therefore, φ is a surjective homomorphism from $\mathcal{A}'$ to $\mathcal{M}$. Consequently, by Lemma 23,

- $|Q/\tau_{\mathcal{A}}| \leq |Q'|$,
- $\sum_{q/\tau_{\mathcal{A}} \in Q/\tau_{\mathcal{A}}} |S^{\rightarrow c}_{q/\tau_{\mathcal{A}}}/\rho_{\mathcal{B}^c_{q/\tau_{\mathcal{A}}}}| \leq \sum_{q \in Q'} |T^{\rightarrow}_q|$, and
- $\sum_{q/\tau_{\mathcal{A}} \in Q/\tau_{\mathcal{A}}} |S^{\leftarrow c}_{q/\tau_{\mathcal{A}}}/\rho_{\mathcal{B}^c_{q/\tau_{\mathcal{A}}}}| \leq \sum_{q \in Q'} |T^{\leftarrow}_q|$.

Therefore, $\mathcal{M}$ is a minimal DTTA. By Lemma 14, $\mathcal{A}/\tau_{\mathcal{A}}$ is equivalent to $\mathcal{A}$. Hence, by Lemma 5, $\mathcal{M}$ is equivalent to $\mathcal{A}$ as well. □

In the rest of the paper we give an algorithm which computes the minimal DTTA which is equivalent to $\mathcal{A}$. For this we will need the concept of the direct product of bimachines. The *direct product of the semi-automata* $\mathcal{S}$ *and* $\mathcal{T}$ is the semi-automaton $\mathcal{S} \times \mathcal{T} = (S \times T, \Sigma, (s_0, t_0), \delta'')$, where $\delta''((s,t),a) = (\delta(s,a), \delta'(t,a))$ for every $(s,t) \in S \times T$ and $a \in \Sigma$. The *direct product of the bimachines* $\mathcal{B}$ *and* $\mathcal{B}'$ is the bimachine

$$\mathcal{B} \times \mathcal{B}' = (\Sigma, \Gamma \times \Gamma, \mathcal{S}^{\rightarrow} \times \mathcal{T}^{\rightarrow}, \mathcal{S}^{\leftarrow} \times \mathcal{T}^{\leftarrow}, f''),$$

where $f''((s^{\rightarrow}, t^{\rightarrow}), a, (s^{\leftarrow}, t^{\leftarrow})) = (f(s^{\rightarrow}, a, s^{\leftarrow}), f'(t^{\rightarrow}, a, t^{\leftarrow}))$ for all $(s^{\rightarrow}, t^{\rightarrow}) \in S^{\rightarrow} \times T^{\rightarrow}$, $(s^{\leftarrow}, t^{\leftarrow}) \in S^{\leftarrow} \times T^{\leftarrow}$, and $a \in \Sigma$.

To give an algorithm which computes the minimal automaton equivalent to $\mathcal{A}$, we define the relation $\tau_n \subseteq Q \times Q$ for every $n \geq 0$, by induction on n.

Base of induction: For each $p, q \in Q$, let $p\tau_0 q$ if and only if ($p \in F \iff q \in F$).

Induction step: Let $n \geq 0$ and assume that we have defined τ_n. For each $p, q \in Q$, let $p\tau_{n+1}q$ if and only if

- $p\tau_n q$ and

- for the bimachine $\mathcal{B}_p \times \mathcal{B}_q = (\Sigma, Q \times Q, \mathcal{S}_p^{\rightarrow} \times \mathcal{S}_q^{\rightarrow}, \mathcal{S}_p^{\leftarrow} \times \mathcal{S}_q^{\leftarrow}, f_{(p,q)})$ and for any reachable pair $((s^{\rightarrow}, t^{\rightarrow}), (s^{\leftarrow}, t^{\leftarrow}))$ in $\mathcal{B}_p \times \mathcal{B}_q$ and $a \in \Sigma$, if $f_{(p,q)}((s^{\rightarrow}, t^{\rightarrow}), a, (s^{\leftarrow}, t^{\leftarrow})) = (r_1, r_2)$, then we have $r_1 \tau_n r_2$.

Lemma 25. For each $n \geq 0$, τ_n is an equivalence relation.

Proof. We proceed by induction on n.

Base of induction: $n = 0$. By definition, τ_0 is an equivalence relation.

Induction step: We assume that the lemma holds for $n \geq 0$, and show that it also holds for $n+1$. By definition and the induction hypothesis, τ_{n+1} is reflexive and symmetric. We will show that τ_{n+1} is transitive. To this end, let $p, q, r \in Q$, and assume that $p\tau_{n+1}q$ and $q\tau_{n+1}r$. Since $p\tau_{n+1}q$ and $q\tau_{n+1}r$, we have $p\tau_n q$ and $q\tau_n r$. By the induction hypothesis, $p\tau_n r$. All is left to show is that for the bimachine $\mathcal{B}_p \times \mathcal{B}_r = (\Sigma, Q \times Q, \mathcal{S}_p^{\rightarrow} \times \mathcal{S}_r^{\rightarrow}, \mathcal{S}_p^{\leftarrow} \times \mathcal{S}_r^{\leftarrow}, f_{(p,r)})$ and for any reachable pair $((s^{\rightarrow}, t^{\rightarrow}), (s^{\leftarrow}, t^{\leftarrow}))$ in $\mathcal{B}_p \times \mathcal{B}_r$ and $a \in \Sigma$, if $f_{(p,r)}((s^{\rightarrow}, t^{\rightarrow}), a, (s^{\leftarrow}, t^{\leftarrow})) = (p', r')$, then we have $p'\tau_n r'$. To this end, take a word $w = a_1 \dots a_k \in \Sigma^*$, $k \geq 1$, such that

- $(s_0^{\rightarrow}, t_0^{\rightarrow})(s_1^{\rightarrow}, t_1^{\rightarrow}) \dots (s_{k-1}^{\rightarrow}, t_{k-1}^{\rightarrow})(s_k^{\rightarrow}, t_k^{\rightarrow})$ is the run of $\mathcal{S}_p^{\rightarrow} \times \mathcal{S}_r^{\rightarrow}$ on $a_1 \dots a_k$,
- $(s_0^{\leftarrow}, t_0^{\leftarrow})(s_1^{\leftarrow}, t_1^{\leftarrow}) \dots (s_{k-1}^{\leftarrow}, t_{k-1}^{\leftarrow})(s_k^{\leftarrow}, t_k^{\leftarrow})$ is the run of $\mathcal{S}_p^{\leftarrow} \times \mathcal{S}_r^{\leftarrow}$ on $a_k \dots a_1$,
- $((s^{\rightarrow}, t^{\rightarrow}), a, (s^{\leftarrow}, t^{\leftarrow})) = ((s_{j-1}^{\rightarrow}, t_{j-1}^{\rightarrow}), a_j, (s_{k-j}^{\leftarrow} t_{k-j}^{\leftarrow}))$ for some $1 \leq j \leq k$ and
- $(p_i, r_i) = f_{(p,r)}((s_{i-1}^{\rightarrow}, t_{i-1}^{\rightarrow}), a_i, (s_{k-i}^{\leftarrow}, t_{k-i}^{\leftarrow}))$ for $1 \leq i \leq k$.

Then $\|\mathcal{B}_p \times \mathcal{B}_r\|(w) = (p_1, r_1) \dots (p_k, r_k)$ and $(p', r') = (p_j, r_j)$.

Let $y_0^{\rightarrow} y_1^{\rightarrow} \dots y_{k-1}^{\rightarrow} y_k^{\rightarrow}$ be the run of $\mathcal{S}_q^{\rightarrow}$ on $a_1 \dots a_k$, and $y_0^{\leftarrow} y_1^{\leftarrow} \dots y_{k-1}^{\leftarrow} y_k^{\leftarrow}$ the run of $\mathcal{S}_q^{\leftarrow}$ on the reversed input $a_k \dots a_1$, and let $q_i = f(y_{i-1}^{\rightarrow}, a_i, y_{k-i}^{\leftarrow})$ for $1 \leq i \leq k$. Then $\|\mathcal{B}_q\|(w) = q_1 \dots q_k$ and $\|\mathcal{B}_p \times \mathcal{B}_q\|(w) = (p_1, q_1) \dots (p_k, q_k)$ and $\|\mathcal{B}_q \times \mathcal{B}_r\|(w) = (q_1, r_1) \dots (q_k, r_k)$. Since $p\tau_{n+1}q$ and $q\tau_{n+1}r$, we have $p_j \tau_n q_j$ and $q_j \tau_n r_j$. By the induction hypothesis, τ_n is an equivalence relation, hence $p_j \tau_n r_j$. Since $(p', r') = (p_j, r_j)$, we have $p'\tau_n r'$. Therefore $p\tau_{n+1}r$, and hence τ_{n+1} is transitive. □

Obviously, we have

$$\tau_0 \supseteq \tau_1 \supseteq \tau_2 \supseteq \cdots$$

and thus there is an integer $n_0 \geq 0$ such that $\tau_{n_0} = \tau_{n_0+1}$. Moreover, we can prove that $\tau_{n_0} = \tau_{n_0+1}$ implies $\tau_{n_0+1} = \tau_{n_0+2} = \cdots$ for every $n_0 \geq 0$.

Lemma 26. For all $n, l \geq 0$, $p, q \in Q$, $\xi \in T_\Sigma$ with $\mathrm{height}(\xi) \geq l$, $x \in \mathrm{dom}(\xi)$ with $|x| = l$, p-run r_p of $\mathcal{A}$ on ξ and q-run r_q of $\mathcal{A}$ on ξ, if $p\tau_{n+l}q$, then $r_p(x)\tau_n r_q(x)$.

Proof. We proceed by induction on l. If $l = 0$, then $p = r_p(x)$ and $q = r_q(x)$. By our assumption $p\tau_{n+0}q$, we have $r_p(x)\tau_n r_q(x)$.

Induction step: We assume that the lemma holds for $l \geq 0$, and show that it also holds for $l+1$. To this end, let $\xi \in T_\Sigma$ with $\xi = a(\xi_1 \dots \xi_k)$, $\mathrm{height}(\xi) \geq l+1$, and let $x = iy$, where $0 \leq i \leq k$, $|x| = l+1$ and hence $|y| = l$, and assume that $p\tau_{n+l+1}q$. Consider an arbitrary p-run r_p of $\mathcal{A}$ on ξ and an arbitrary q-run r_q of $\mathcal{A}$ on ξ. If $r_p(i) = p'$ and $r_q(i) = q'$, then by the definition of τ_{n+l+1}, $p'\tau_{n+l}q'$. Hence, by the induction hypothesis, for the p'-run $r_{p'}$ of $\mathcal{A}$ on ξ_i and for the q'-run $r_{q'}$ of $\mathcal{A}$ on ξ_i, we have $r_{p'}(y)\tau_n r_{q'}(y)$. Observe that $r_{p'}(y) = r_p(x)$ and $r_{q'}(y) = r_q(x)$. Consequently, $r_p(x)\tau_n r_q(x)$. □

Lemma 27. Let n_0 be the least integer with $\tau_{n_0} = \tau_{n_0+1}$. Then $\tau_{n_0} = \tau_{\mathcal{A}}$.

Proof. First we show that $\tau_{n_0} \subseteq \tau_{\mathcal{A}}$. Let $p\tau_{n_0}q$. Then $p\tau_{n_0+l}q$ for each $l \geq 0$, hence by Lemma 26, for all $l \geq 0$, $\xi \in T_\Sigma$ with height$(\xi) \geq l$, $x \in \text{dom}(\xi)$ with $|x| = l$, p-run r_p of $\mathcal{A}$ on ξ and q-run r_q of $\mathcal{A}$ on ξ, we have $r_p(x)\tau_n r_q(x)$. By the inclusion $\tau_0 \supseteq \tau_{n_0}$, we have $r_p(x)\tau_0 r_q(x)$. Hence, by the definition of ρ_0, we have ($r_p(x) \in F$ if and only if $r_q(x) \in F$). Since $l \geq 0$, $\xi \in T_\Sigma$, and $x \in \text{dom}(\xi)$ are arbitrary, $L(\mathcal{A}, p) = L(\mathcal{A}, q)$.

We now show that $\tau_{\mathcal{A}} \subseteq \tau_{n_0}$. To this end we show that for all $p, q \in Q$, $n \geq 0$, if $(p, q) \notin \tau_n$, then $(p, q) \notin \tau_{\mathcal{A}}$. We proceed by induction on n.

Base of induction: $n = 0$. If $(p, q) \notin \tau_0$, then ($p \in F$ if and only if $q \notin F$). Hence $L(\mathcal{A}, p) \neq L(\mathcal{A}, q)$ and thus $p\tau_A q$ does not hold.

Induction step. Assume that $p\tau_{n+1}q$ does not hold. Then $p\tau_n q$ does not hold or $p\tau_n q$ and there is a word $z \in \Sigma^*$ such that $\|\mathcal{B}_p\|(z) = p_1 \ldots p_k$ and $\|\mathcal{B}_q\|(z) = q_1 \ldots q_k$ and $(p_i, q_i) \notin \tau_n$ for some $1 \leq i \leq k$. In the first case, by the induction hypothesis, $(p, q) \notin \tau_{\mathcal{A}}$. In the second case, $L(\mathcal{A}, p_i) \setminus L(\mathcal{A}, q_i) \neq \emptyset$ or $L(\mathcal{A}, q_i) \setminus L(\mathcal{A}, p_i) \neq \emptyset$. If $L(\mathcal{A}, p_i) \setminus L(\mathcal{A}, q_i) \neq \emptyset$, then let $\xi_i \in (L(\mathcal{A}, p_i) \setminus L(\mathcal{A}, q_i))$, otherwise let $\xi_i \in L(\mathcal{A}, p_i)$. If $L(\mathcal{A}, q_i) \setminus L(\mathcal{A}, p_i) \neq \emptyset$, then let $\zeta_i \in (L(\mathcal{A}, q_i) \setminus L(\mathcal{A}, p_i))$, otherwise let $\zeta_i \in L(\mathcal{A}, q_i)$. For each $1 \leq j \leq k$ with $j \neq i$, let $\xi_j \in L(\mathcal{A}, p_j)$ and $\zeta_j \in L(\mathcal{A}, q_j)$. Then let $\xi = a(\xi_1 \ldots \xi_k)$ and $\zeta = a(\zeta_1 \ldots \zeta_k)$. Consequently, $\xi \in (L(\mathcal{A}, p) \setminus L(\mathcal{A}, q))$ or $\zeta \in (L(\mathcal{A}, q) \setminus L(\mathcal{A}, p))$. Hence $L(\mathcal{A}, p) \neq L(\mathcal{A}, q)$ and thus $p\tau_A q$ does not hold. □

Proposition 7. There is a polynomial time algorithm which constructs $\mathcal{A}/\tau_{\mathcal{A}}$ for a given $\mathcal{A}$.

Proof. We compute τ_1 in $\mathcal{O}(|Q|^2)$ time. For every $1 < n \leq n_0$, the relation τ_n can be computed in $\mathcal{O}(|Q|^2(N^{\rightarrow})^2|\Sigma|(N^{\leftarrow})^2)$ time, where $N^{\rightarrow} = \max\{|S_p^{\rightarrow}| \mid p \in Q\}$ and $N^{\leftarrow} = \max\{|S_p^{\leftarrow}| \mid p \in Q\}$. Since there are at most $|Q|$ steps, the relation τ_{n_0} can be computed in $\mathcal{O}(|Q|^3(N^{\rightarrow})^2|\Sigma|(N^{\leftarrow})^2)$ time. □

Theorem 3. There is a polynomial time algorithm which constructs for $\mathcal{A}$ an equivalent minimal DTTA.

Proof. By Propositions 6, 7, 2, and 3, respectively, we compute the following sequence of DTTAs in polynomial time.

1) The connected part $\mathcal{A}^c = \big(Q^c, \Sigma, f^c_{\text{in}}, (\mathcal{B}_q \mid q \in Q^c), F^c\big)$ of $\mathcal{A}$.

2) The congruence $\tau_{\mathcal{A}^c}$ and the DTTA

$$\mathcal{A}^c/\tau_{\mathcal{A}^c} = \big(Q^c/\tau_{\mathcal{A}^c}, \Sigma, f^c_{\text{in},\tau^c_{\mathcal{A}}}, (\mathcal{B}_{q/\tau_{\mathcal{A}^c}} \mid q/\tau_{\mathcal{A}^c} \in Q^c/\tau_{\mathcal{A}^c}), F^c/\tau_{\mathcal{A}^c}\big).$$

3) For each $q/\tau_{\mathcal{A}^c} \in Q^c/\tau_{\mathcal{A}^c}$, the connected part $\mathcal{B}^c_{q/\tau_{\mathcal{A}^c}}$ of $\mathcal{B}_{q/\tau_{\mathcal{A}^c}}$.

4) The DTTA

$$\big(Q^c/\tau_{\mathcal{A}^c}, \Sigma, f^c_{\text{in},\tau_{\mathcal{A}^c}}, (\mathcal{B}^c_{q/\tau_{\mathcal{A}^c}}/\rho_{\mathcal{B}^c_{q/\tau_{\mathcal{A}^c}}} \mid q/\tau_{\mathcal{A}^c} \in Q^c/\tau_{\mathcal{A}^c}), F^c/\tau_{\mathcal{A}^c}\big).$$

By Lemma 24, the latter one is a minimal DTTA which is equivalent to $\mathcal{A}$. □

References

[1] Björklund J. and Cleophas L. A Taxonomy of Minimisation Algorithms for Deterministic Tree Automata. Journal of Universal Computer Science vol. 22(2): 180–196, 2016.

[2] Brüggemann-Klein A., Murata M., and Wood D., *Regular tree and regular hedge languages over unranked trees.* Technical Report HKUST-TCSC-2001-0, The Hong Kong University of Science and Technology, Hong Kong, China, 2001.

[3] Comon H., Dauchet M., Gilleron R., Löding C., Jacquemard F., Lugiez D., Tison S., and Tommasi M. *Tree Automata Techniques and Applications.* http://www.grappa.univ-lille3.fr/tata, 2007.

[4] Cristau J., Löding C., and Thomas W. Deterministic Automata on Unranked Trees. In Liskiewicz M. and Reischuk R. editors, *Fundamentals of Computation Theory, 15th International Symposium, FCT 2005, Proceedings*, Lecture Notes in Computer Science 3623, pages 68–79. Springer-Verlag, Berlin, 2005.

[5] Gécseg F. and Steinby M. Minimal ascending tree automata. *Acta Cybernetica* 4(1): 37–44, 1978.

[6] Jiang T. and Ravikumar B. Minimal NFA problems are hard. *SIAM Journal on Computing* 22(6): 1117–1141, 1993.

[7] Martens W. and Niehren J. On the minimization of XML Schemas and tree automata for unranked trees. *Journal of Computer and Systems Sciences* 73(4): 550–583, 2007.

[8] Martens W., Neven F., and Schwentick T. Deterministic Top-down Tree Automata: Past, Present, and Future. In Flum J., Grädel E, and Wilke T. editors, *Logic and Automata – History and Perspectives*, pages 505–530. Amsterdam University Press, 2008.

[9] Martens W., Neven F., Schwentick T., and Bex G. J. Expressiveness and complexity of XML Schema, *Journal ACM Transactions on Database Systems (TODS)* 31(3): 770–813, 2006.

[10] Neven F. Automata theory for XML researchers. *ACM Sigmod Record* 31(3): 39–46, 2002.

[11] Piao X. and Salomaa K. Operational State Complexity of Deterministic Unranked Tree Automata, In McQuillan I. and Pighizzini G. editors, *Proceedings Twelfth Annual Workshop on Descriptional Complexity of Formal Systems, DCFS 2010*, *Electronic Proceedings in Theoretical Computer Science* 31: 149–158, 2010.

[12] Piao X. and Salomaa K. Lower bounds for the size of deterministic unranked tree automata, *Theoretical Computer Science* 454: 231–239, 2012.