Equivalence of Linear Tree Transducers with Output in the Free Group

Raphaela Löbel, Michael Luttenberger, and Helmut Seidl
TU München, Munich, Germany
{loebel,luttenbe,seidl}@in.tum.de

Abstract. We show that equivalence of deterministic linear tree transducers can be decided in polynomial time when their outputs are interpreted over the free group. Due to the cancellation properties offered by the free group, the required constructions are not only more general, but also simpler than the corresponding constructions for proving equivalence of deterministic linear tree-to-word transducers.

Keywords: Linear tree transducer · Free group · Equivalence problem · Polynomial time

1 Introduction

In 2009, Staworko and Niehren observed that equivalence for sequential tree-to-word transducers [13] can be reduced to the morphism equivalence problem for context-free languages. Since the latter problem is decidable in polynomial time [10], they thus proved that equivalence of sequential tree-to-word transducers is decidable in polynomial time. This decision procedure was later accompanied by a canonical normal form which can be applied to learning [3,4]. Sequentiality of transducers means that subtrees must always be processed from left to right. This restriction was lifted by Boiret who provided a canonical normal form for unrestricted linear tree-to-word transducers [1]. Construction of that normal form, however, may require exponential time implying that the corresponding decision procedure requires exponential time as well. In order to improve on that, Palenta and Boiret provided a polynomial time procedure which just normalizes the order in which an unrestricted linear tree-to-word transducer processes the subtrees of its input [2]. They proved that after that normalization, equivalent transducers are necessarily same-ordered. As a consequence, equivalence of linear tree-to-word transducers can thus also be reduced to the morphism equivalence problem for context-free languages and thus can be decided in polynomial time. Independently of that, Seidl, Maneth and Kemper showed by algebraic means, that equivalence of general (possibly non-linear) tree-to-word transducers is decidable [11]. Their techniques are also applicable if the outputs of transducers are not just in a free monoid of words, but also if outputs are in a free group. The latter means that output words are considered as equivalent not just when they are literally equal, but also when they...
become equal after cancellation of matching positive and negative occurrences of letters. For the special case of linear tree transducers with outputs either in a free monoid or a free group, Seidl et al. provided a randomized polynomial time procedure for in-equivalence. The question remained open whether for outputs in a free group, randomization can be omitted. Here, we answer this question to the affirmative. In fact, we follow the approach of [2] to normalize the order in which tree transducers produce their outputs. For that normalization, we heavily rely on commutation laws as provided for the free group. Due to these laws, the construction as well as the arguments for its correctness, are not only more general but also much cleaner than in the case of outputs in a free monoid only. The observation that reasoning over the free group may simplify arguments has also been made, e.g., by Tomita and Seino and later by Senizergues when dealing with the equivalence problem for deterministic pushdown transducers [12,14]. As morphism equivalence on context-free languages is decidable in polynomial time—even if the morphism outputs are in a free group [10], we obtain a polynomial time algorithm for equivalence of tree transducers with output in the free group.

Missing proofs can be found in the extended version of this paper [5].

2 Preliminaries

We use Σ to denote a finite ranked alphabet, while A is used for an unranked alphabet. \( T_\Sigma \) denotes the set of all trees (or terms) over Σ. The depth \( \text{depth}(t) \) of a tree \( t \in T_\Sigma \) equals 0, if \( t = f() \) for some \( f \in \Sigma \) of rank 0, and otherwise, \( \text{depth}(t) = 1 + \max\{\text{depth}(t_i) \mid i = 1, \ldots, m\} \) for \( t = f(t_1, \ldots, t_m) \). We denote by \( F_A \) the representation of the free group generated by \( A \) where the carrier is the set of reduced words instead of the usual quotient construction: For each \( a \in A \), we introduce its inverse \( a^- \). The set of elements of \( F_A \) then consists of all words over the alphabet \( \{a, a^- \mid a \in A\} \) which do not contain \( aa^- \) or \( a^-a \) as factors. These words are also called reduced. In particular, \( A^* \subseteq F_A \). The group operation \( \cdot \) of \( F_A \) is concatenation, followed by reduction, i.e., repeated cancellation of subwords \( aa^- \) or \( a^-a \). Thus, \( abc^- \cdot cb^-a = F_A a \). The neutral element w.r.t. this operation is the empty word \( \varepsilon \), while the inverse \( w^- \) of some element \( w \in F_A \) is obtained by reverting the order of the letters in \( w \) while replacing each letter \( a \) with \( a^- \) and each \( a^- \) with \( a \). Thus, e.g., \( (abc^-)^- = cb^-a^- \).

In light of the inverse operation \( (.)^- \), we have that \( v \cdot w = F_A v'w' \) where \( v = v'u \) (as words) for a maximal suffix \( u \) so that \( u^- \) is a prefix of \( w \) with \( w = u^-w' \). For an element \( w \in F_A \), \( \langle w \rangle = \{w^l \mid l \in \mathbb{Z}\} \) denotes the cyclic subgroup of \( F_A \) generated from \( w \). As usual, we use the convention that \( w^0 = \varepsilon \), and \( w^{-l} = (w^-)^l \) for \( l > 0 \). An element \( p \in F_A \) different from \( \varepsilon \), is called primitive if \( w^l = F_A p \) for some \( w \in F_A \) and \( l \in \mathbb{Z} \) implies that \( w = F_A p \) or \( w = F_A p^- \), i.e., \( p \) and \( p^- \) are the only (trivial) roots of \( p \). Thus, primitive elements generate maximal cyclic subgroups of \( F_A \). We state two crucial technical lemmas.

**Lemma 1.** Assume that \( y^m = F_A \beta \cdot y^n \cdot \beta^- \) with \( y \in F_A \) primitive. Then \( m = n \), and \( \beta = F_A y^k \) for some \( k \in \mathbb{Z} \).
Proof. Since $\beta \cdot y^n \cdot \beta^{-1} = \mathcal{F}_\Lambda (\beta \cdot y \cdot \beta^{-1})^n$, we find by [8, Proposition 2.17] a primitive element $c$ such that $y$ and $\beta \cdot y \cdot \beta^{-1}$ are powers of $c$. As $y$ is primitive, $c$ can be chosen as $y$. Accordingly,

$$y^j = \mathcal{F}_\Lambda \beta \cdot y \cdot \beta^{-1} \tag{1}$$

holds for some $j$. If $\beta$ is a power of $y$, then $\beta \cdot y \cdot \beta^{-1} = \mathcal{F}_\Lambda y$, and the assertion of the lemma holds. Likewise if $j = 1$, then $\beta$ and $y$ commute. Since $y$ is primitive, then $\beta$ necessarily must be a power of $y$.

For a contradiction, therefore now assume that $\beta$ is not a power of $y$ and $j \neq 1$. W.l.o.g., we can assume that $j > 1$. First, assume now that $y$ is cyclically reduced, i.e., the first and last letters, $a$ and $b$, respectively, of $y$ are not mutually inverse. Then for each $n > 0$, $y^n$ is obtained from $y$ by $n$-concatenation of $y$ as a word (no reduction taking place). Likewise, either the last letter of $\beta$ is different $a^{-1}$ or the first letter of $\beta^{-1}$ is different from $b^{-1}$ because these two letters are mutually inverse. Assume that the former is the case. Then $\beta \cdot y$ is obtained by concatenation of $\beta$ and $y$ as words (no reduction taking place). By (1), $\beta \cdot y^n = \mathcal{F}_\Lambda y^{j-n} \cdot \beta$, for every $n \geq 1$. Let $m > 0$ denote the length of $\beta$ as a word. Since $\beta$ can cancel only a suffix of $y^{j-n}$ of length at most $m$, it follows, that the word $\beta y$ must a prefix of the word $y^{m+1}$. Since $\beta$ is not a power of $y$, the word $y$ can be factored into $y = yc$ for some non-empty suffix $c$ such that $\beta = y^j y'$, implying that $yc = cy$ holds. As a consequence, $y = c^l$ for some $l > 1$—in contradiction to the irreducibility of $y$.

If on the other hand, the first letter of $\beta^{-1}$ is not the inverse of the last letter of $y$, then $y \cdot \beta^{-1}$ is obtained as the concatenation of $y$ and $\beta^{-1}$ as words. As a consequence, $y\beta^{-1}$ is a suffix of $y^{m+1}$, and we arrive at a contradiction.

We conclude that the statement of the lemma holds whenever $y$ is cyclically reduced. Now assume that $y$ is not yet cyclically reduced. Then we can find a maximal suffix $r$ of $y$ (considered as a word) such that $y = r^{-1} sr$ holds and $s$ is cyclically reduced. Then $s$ is also necessarily primitive. (If $s = \mathcal{F}_\Lambda c^n$, then $y = \mathcal{F}_\Lambda (r^{-1} cr)^n$). Then assertion (1) can be equivalently formulated as

$$s^j = \mathcal{F}_\Lambda (r \cdot \beta \cdot r^{-1}) \cdot y \cdot (r \cdot \beta \cdot r^{-1})^{-1}$$

We conclude that $r \cdot \beta \cdot r^{-1} = \mathcal{F}_\Lambda s^l$ for some $l \in \mathbb{Z}$. But then $\beta = \mathcal{F}_\Lambda (r^{-1} \cdot s \cdot r)^l = \mathcal{F}_\Lambda y^l$, and the claim of the lemma follows.

Lemma 2. Assume that $x_1, x_2$ and $y_1, y_2$ are distinct elements in $\mathcal{F}_\Lambda$ and that

$$x_i \cdot \alpha \cdot y_j \cdot \beta = \mathcal{F}_\Lambda \gamma \cdot y_j' \cdot \alpha' \cdot x_i' \cdot \beta' \tag{2}$$

holds for $i = 1, 2$ and $j = 1, 2$. Then there is some primitive element $p$ and exponents $r, s \in \mathbb{Z}$ such that $x_1 \cdot \alpha = \mathcal{F}_\Lambda x_2 \cdot \alpha \cdot p^r$ and $y_1 = \mathcal{F}_\Lambda p^s \cdot y_2$.

Proof. By the assumption (2),

$$\gamma = \mathcal{F}_\Lambda (x_1 \cdot \alpha \cdot y_j \cdot \beta) \cdot (y_j' \cdot \alpha' \cdot x_i' \cdot \beta')^{-1} = \mathcal{F}_\Lambda (x_2 \cdot \alpha \cdot y_j \cdot \beta) \cdot (y_j' \cdot \alpha' \cdot x_2' \cdot \beta')^{-1}$$
for all \( j = 1, 2 \). Thus,

\[
\begin{align*}
  x_1 \cdot \alpha \cdot y_j \cdot \beta \beta' y_j^- &= \mathcal{F}_A x_2 \cdot \alpha \cdot y_j \cdot \beta \beta' x'_2^- \cdot \alpha' y_j^- \quad \text{implying} \\
  y_j^- \cdot \alpha^- \cdot x'_2^- \cdot x_1 \cdot \alpha \cdot y_j &= \mathcal{F}_A \beta \cdot \beta' \cdot x'_2^- \cdot x_1' \cdot \beta' \cdot \beta^-
\end{align*}
\]

for \( j = 1, 2 \). Hence,

\[
(x_2 \cdot \alpha)^- x_1 \cdot \alpha = \mathcal{F}_A (y_1 \cdot y_2^-) \cdot ((x_2 \cdot \alpha)^- \cdot x_1 \cdot \alpha) \cdot (y_1 \cdot y_2^-)
\]

Since \( x_1 \) is different from \( x_2 \), also \( x_1 \cdot \alpha \) is different from \( x_2 \cdot \alpha \). Let \( p \) denote a primitive root of \( (x_2 \cdot \alpha)^- \cdot x_1 \cdot \alpha \). Then by Lemma 1,

\[
\begin{align*}
  x_1 \cdot \alpha &= \mathcal{F}_A x_2 \cdot \alpha \cdot p^r \\
  y_1 &= \mathcal{F}_A p^s \cdot y_2
\end{align*}
\]

for suitable exponents \( r, s \in \mathbb{Z} \).

As the elements of \( \mathcal{F}_A \) are words, they can be represented by straight-line programs (SLPs). An SLP is a context-free grammar where each non-terminal occurs as the left-hand side of exactly one rule. We briefly recall basic complexity results for operations on elements of \( \mathcal{F}_A \) when represented as SLPs [7].

**Lemma 3.** Let \( U, V \) be SLPs representing words \( w_1, w_2 \in \{a, a^- \mid a \in A\} \), respectively. Then the following computations/decision problems can be realized in polynomial time

- compute an SLP for \( w_1^- \);
- compute the primitive root of \( w_1 \) if \( w_1 \neq \varepsilon \);
- compute an SLP for \( w = \mathcal{F}_A w_1 \) with \( w \) reduced;
- decide whether \( w_1 = \mathcal{F}_A w_2 \);
- decide whether it exists \( g \in \mathcal{F}_A \), such that \( w_1 \in g \cdot \langle w_2 \rangle \) and compute an SLP for such \( g \).

In the following, we introduce deterministic linear tree transducers which produce outputs in the free group \( \mathcal{F}_A \). For convenience, we follow the approach in [11] where only total deterministic transducers are considered—but equivalence is relative w.r.t. some top-down deterministic domain automaton \( B \). A top-down deterministic automaton (DTA) \( B \) is a tuple \((H, \Sigma, \delta_B, h_0)\) where \( H \) is a finite set of states, \( \Sigma \) is a finite ranked alphabet, \( \delta_B : H \times \Sigma \rightarrow H^* \) is a partial function where \( \delta_B(h, f) \in H^k \) if the the rank of \( f \) equals \( k \), and \( h_0 \) is the start state of \( B \). For every \( h \in H \), we define the set \( \text{dom}(h) \subseteq T_\Sigma \) by \( f(t_1, \ldots, t_m) \in \text{dom}(h) \) iff \( \delta_B(h, f) = h_1 \ldots h_m \) and \( t_i \in \text{dom}(h_i) \) for all \( i = 1, \ldots, k \). \( B \) is called reduced if \( \text{dom}(h) \neq \emptyset \) for all \( h \in H \). The language \( \mathcal{L}(B) \) accepted by \( B \) is the set \( \text{dom}(h_0) \).

We remark that for every DTA \( B \) with \( \mathcal{L}(B) \neq \emptyset \), a reduced DTA \( B' \) can be constructed in polynomial time with \( \mathcal{L}(B) = \mathcal{L}(B') \). Therefore, we subsequently assume w.l.o.g. that each DTA \( B \) is reduced.

A (total deterministic) linear tree transducer with output in \( \mathcal{F}_A \) (LT\( A \) for short) is a tuple \( M = (\Sigma, A, Q, S, R) \) where \( \Sigma \) is the ranked alphabet for the
input trees, $A$ is the finite (unranked) output alphabet, $Q$ is the set of states, $S$ is the axiom of the form $u_0$ or $u_0 \cdot q_0(x_0) \cdot u_1$ with $u_0, u_1 \in \mathcal{F}_A$ and $q_0 \in Q$, and $R$ is the set of rules which contains for each state $q \in Q$ and each input symbol $f \in \Sigma$, one rule of the form

$$q(f(x_1, \ldots, x_m)) \rightarrow u_0 \cdot q_1(x_{\sigma(1)}) \cdot \ldots \cdot u_{n-1} \cdot q_n(x_{\sigma(n)}) \cdot u_n$$  \hspace{1cm} (3)

Here, $m$ is the rank of $f$, $n \leq m$, $u_0, \ldots, u_n \in \mathcal{F}_A$ and $\sigma$ is an injective mapping from $\{1, \ldots, n\}$ to $\{1, \ldots, m\}$. The semantics of a state $q$ is the function $[q] : T_\Sigma \rightarrow \mathcal{F}_A$ defined by

$$[q](f(t_1, \ldots, t_m)) = \mathcal{F}_A u_0 \cdot [q_1](t_{\sigma(1)}) \cdot \ldots \cdot u_{n-1} \cdot [q_n](t_{\sigma(n)}) \cdot u_n$$

if there is a rule of the form (3) in $R$. Then the translation of $M$ is the function $[M] : T_\Sigma \rightarrow \mathcal{F}_A$ defined by $[M](t) = \mathcal{F}_A u_0$ if the axiom of $M$ equals $u_0$, and $[M](t) = \mathcal{F}_A u_0 \cdot [q](t) \cdot u_1$ if the axiom of $M$ is given by $u_0 \cdot q(x_0) \cdot u_1$.

Example 1. Let $A = \{a,b\}$. As a running example we consider the LT$_A$ $M$ with input alphabet $\Sigma = \{f^2, g^1, k^0\}$ where the superscripts indicate the rank of the input symbols. $M$ has axiom $q_0(x_0)$ and the following rules

$$q_0(f(x_1, x_2)) \rightarrow q_1(x_2)bq_2(x_1) \quad q_0(g(x_1)) \rightarrow q_0(x_1) \quad q_0(k) \rightarrow \varepsilon
$$

$$q_1(f(x_1, x_2)) \rightarrow q_0(x_1)q_0(x_2) \quad q_1(g(x_1)) \rightarrow abq_1(x_1) \quad q_1(k) \rightarrow a
$$

$$q_2(f(x_1, x_2)) \rightarrow q_0(x_1)q_0(x_2) \quad q_2(g(x_1)) \rightarrow abq_2(x_1) \quad q_2(k) \rightarrow ab$$

Two LT$_A$s $M, M'$ are equivalent relative to the DTA $B$ iff their translations coincide on all input trees accepted by $B$, i.e., $[M](t) = \mathcal{F}_A [M'](t)$ for all $t \in \mathcal{L}(B)$.

To relate the computations of the LT$_A$ $M$ and the domain automaton $B$, we introduce the following notion. A mapping $\iota : Q \rightarrow H$ from the set of states of $M$ to the set of states of $B$ is called compatible if either the set of states of $M$ is empty (and thus the axiom of $M$ consists of an element of $\mathcal{F}_A$ only), or the following holds:

1. $\iota(q_0) = h_0$;
2. If $\iota(q) = h$, $\delta_B(h, f) = h_1 \ldots h_m$, and there is a rule in $M$ of the form (3) then $\iota(q_i) = h_{\sigma(i)}$ for all $i = 1, \ldots, n$;
3. If $\iota(q) = h$ and $\delta_B(h, f)$ is undefined for some $f \in \Sigma$ of rank $m \geq 0$, then $M$ has the rule $q(f(x_1, \ldots, x_m)) \rightarrow \bot$ for some dedicated symbol $\bot$ which does not belong to $A$.

Lemma 4. For an LT$_A$ $M$ and a DTA $B = (H, \Sigma, \delta_B, h_0)$, an LT$_A$ $M'$ with a set of states $Q'$ together with a mapping $\iota : Q' \rightarrow H$ can be constructed in polynomial time such that the following holds:

1. $M$ and $M'$ are equivalent relative to $B$;
2. $\iota$ is compatible.
Example 2. Let $\mathsf{LT}_A M$ be defined as in Example 1. Consider DTA $B$ with start state $h_0$ and the transition function $\delta_B = \{(h_0, f) \mapsto h_1 h_1, (h_1, g) \mapsto h_1, (h_1, h) \mapsto \varepsilon\}$. According to Lemma 4, $\mathsf{LT}_A M'$ for $M$ then is defined as follows. $M'$ has axiom $\langle q_0, h_0 \rangle(x_0)$ and the rules
\[
\langle q_0, h_0 \rangle(f(x_1, x_2)) \rightarrow \langle q_1, h_1 \rangle(x_2) b \langle q_2, h_1 \rangle(x_1) \\
\langle q_1, h_1 \rangle(g(x_1)) \rightarrow ab \langle q_1, h_1 \rangle(x_1) \\
\langle q_2, h_1 \rangle(g(x_1)) \rightarrow ab \langle q_2, h_1 \rangle(x_1)
\]
where the rules with left-hand sides $\langle q_0, h_0 \rangle(g(x_1))$, $\langle q_0, h_0 \rangle(h)$, $\langle q_1, h_1 \rangle(f(x_1, x_2))$, $\langle q_2, h_1 \rangle(f(x_1, x_2))$, all have right-hand-sides $\perp$. The compatible map $\iota$ is then given by $\iota = \{(q_0, h_0) \mapsto h_0, \langle q_1, h_1 \rangle \mapsto h_1, \langle q_2, h_1 \rangle \mapsto h_1\}$. For convenience, we again denote the pairs $\langle q_0, h_0 \rangle, \langle q_1, h_1 \rangle, \langle q_2, h_1 \rangle$ with $q_0, q_1, q_2$, respectively.

Subsequently, we w.l.o.g. assume that each $\mathsf{LT}_A M$ with corresponding DTA $B$ for its domain, comes with a compatible map $\iota$. Moreover, we define for each state $q$ of $M$, the set $\mathcal{L}(q) = \{[q](t) \mid t \in \text{dom}(\iota(q))\}$ of all outputs produced by state $q$ (on inputs in $\text{dom}(\iota(q))$), and $\mathcal{L}(i)(q) = \{[q](t) \mid t \in \text{dom}(\iota(q)), \text{depth}(t) < i\}$ for $i \geq 0$.

Beyond the availability of a compatible map, we also require that all states of $M$ are non-trivial (relative to $B$). Here, a state $q$ of $M$ is called trivial if $\mathcal{L}(q)$ contains a single element only. Otherwise, it is called non-trivial. This property will be established in Theorem 1.

3 Deciding Equivalence

In the first step, we show that equivalence relative to the DTA $B$ of same-ordered \(\mathsf{LT}\)s is decidable. For a DTA $B$, consider the \(\mathsf{LT}\)s $M$ and $M'$ with compatible mappings $\iota$ and $\iota'$, respectively. $M$ and $M'$ are same-ordered relative to $B$ if they process their input trees in the same order. We define set of pairs $\langle q, q' \rangle$ of co-reachable states of $M$ and $M'$. Let $u_0 \cdot q_0(x_1) \cdot u_1$ and $u'_0 \cdot q'_0(x_1) \cdot u'_1$ be the axioms of $M$ and $M'$, respectively, where $\iota(q_0) = \iota'(q'_0)$ is the start state of $B$. Then the pair $\langle q_0, q'_0 \rangle$ is co-reachable. Let $\langle q, q' \rangle$ be a pair of co-reachable states. Then $\iota(q) = \iota'(q')$ should hold. For $f \in \Sigma$, assume that $\delta_B(\iota(q), f)$ is defined. Let
\[
q(f(x_1, \ldots, x_m)) \rightarrow u_0 q_1(x_{\sigma(1)}) u_1 \ldots u_{n-1} q_n(x_{\sigma(n)}) u_n
\]
\[
q'(f(x_1, \ldots, x_m)) \rightarrow u'_0 q'_1(x'_{\sigma'(1)}) u'_1 \ldots u'_{n'-1} q'_n(x'_{\sigma'(n')}) u'_{n'}
\]
be the rules of $q, q'$ for $f$, respectively. Then $\langle q_j, q'_j \rangle$ is co-reachable whenever $\sigma(j) = \sigma'(j')$ holds. In particular, we then have $\iota(q_j) = \iota'(q'_j)$.

The pair $\langle q, q' \rangle$ of co-reachable states is called same-ordered, if for each corresponding pair of rules (4), $n = n'$ and $\sigma = \sigma'$. Finally, $M$ and $M'$ are same-ordered if for every co-reachable pair $\langle q, q' \rangle$ of states of $M, M'$, and every $f \in \Sigma$, each pair of rules (4) is same-ordered whenever $\delta_B(\iota(q), f)$ is defined.

Given that the \(\mathsf{LT}\)s $M$ and $M'$ are same-ordered relative to $B$, we can represent the set of pairs of runs of $M$ and $M'$ on input trees by means of
a single context-free grammar $G$. The set of nonterminals of $G$ consists of a
distinct start nonterminal $S$ together with all co-reachable pairs $\langle q, q' \rangle$ of states
$q, q'$ of $M, M'$, respectively. The set of terminal symbols $T$ of $G$ is given by
$\{a, a^-, \bar{a}, \bar{a}^- \mid a \in A\}$ for fresh distinct symbols $\bar{a}, \bar{a}^-, a \in A$. Let $\langle q, q' \rangle$ be a
co-reachable pair of states of $M, M'$, and $f \in \Sigma$ such that $\delta_B(\iota(q), f)$ is defined.
For each corresponding pair of rules (4), $G$ receives the rule

$$\langle q, q' \rangle \rightarrow u_0 \bar{u}_0 q_1 q'_1 u_1 \bar{u}_1 \ldots u_{n-1} \bar{u}_{n-1} q_n q'_n u_n \bar{u}_n$$

where $\bar{u}_i$ is obtained from $u_i$ by replacing each output symbol $a \in A$ with
its barred copy $\bar{a}$ as well as each inverse $a^-$ with its barred copy $\bar{a}^-$. For the
axioms $u_0 q(x_1) u_1$ and $u'_0 q'(x_1) u'_1$ of $M, M'$, respectively, we introduce the rule
$S \rightarrow u_0 \bar{u}_0 q, q'u_1 \bar{u}_1$ where again $\bar{u}_1$ are the barred copies of $u_1$. We define
morphisms $f, g : T^* \rightarrow \mathcal{F}_A$ by

$$f(a) = a \quad f(a^-) = a^- \quad f(\bar{a}) = f(\bar{a}^-) = \varepsilon$$
$$g(\bar{a}) = a \quad g(\bar{a}^-) = a^- \quad g(a) = g(a^-) = \varepsilon$$

for $a \in A$. Then $M$ and $M'$ are equivalent relative to $B$ iff $g(w) =_{\mathcal{F}_A} f(w)$ for
all $w \in L(G)$. Combining Plandowski’s polynomial construction of a test set for
a context-free language to check morphism equivalence over finitely generated
free groups [10, Theorem 6], with Lohrey’s polynomial algorithm for checking
equivalence of SLPs over the free group [6], we deduce that the equivalence of
the morphisms $f$ and $g$ on all words generated by the context-free grammar $G$,
is decidable in polynomial time. Consequently, we obtain:

**Corollary 1.** Equivalence of same-ordered $\text{LT}_A$s relative to a DTA $B$ is decidable
in polynomial time. \(\square\)

Next, we observe that for every $\text{LT}_A M$ with compatible map $\iota$ and non-trivial
states only, a canonical ordering can be established. We call $M$ ordered (relative
to $B$) if for all rules of the form (3), with $L(q_i) \cdot u_i \cdot \ldots \cdot u_{j-1} \cdot L(q_j) \subseteq v \cdot \langle p \rangle$, $p \in \mathcal{F}_A$ the ordering $\sigma(i) < \ldots < \sigma(j)$ holds. Here we have naturally extended
the operation “<” to sets of elements.

We show that two ordered $\text{LT}_A$s, when they are equivalent, are necessarily
same-ordered. The proof of this claim is split in two parts. First, we prove that the
set of indices of subtrees processed by equivalent co-reachable states are
identical and second, that the order is the same.

**Lemma 5.** Let $M, M'$ be $\text{LT}_A$s with compatible maps $\iota$ and $\iota'$, respectively, and
non-trivial states only so that $M$ and $M'$ are equivalent relative to the DTA $B$. Let $\langle q, q' \rangle$ be a pair of co-reachable states of $M$ and $M'$. Assume that $\delta_B(\iota(q), f)$
is defined for some $f \in \Sigma$ and consider the corresponding pair of rules (4). Then
the following holds:

1. $\{\sigma(1), \ldots, \sigma(n)\} = \{\sigma'(1), \ldots, \sigma'(n')\}$;
2. $\sigma = \sigma'$. 

Equivalence of Linear Tree Transducers with Output in the Free Group 213
Proof. Since \( \langle q, q' \rangle \) is a co-reachable pair of states, there are elements \( \alpha, \alpha', \beta, \beta' \in \mathcal{F}_A \) such that
\[
\alpha \cdot [q](t) \cdot \beta = \mathcal{F}_A \alpha' \cdot [q'](t) \cdot \beta'
\]
holds for all \( t \in \text{dom}(\iota(q)) \). Consider the first statement. Assume for a contradiction that \( q_k(x_j) \) occurs on the right-hand side of the rule for \( q \) but \( x_j \) does not occur on the right-hand side of the rule for \( q' \). Then, there are input trees \( t = f(t_1, \ldots, t_m) \) and \( t' = f(t'_1, \ldots, t'_m) \), both in \( \text{dom}(\iota(q)) \), such that \( [q_k](t_j) \neq \mathcal{F}_A [q_k](t'_j) \) and \( t_i = t'_i \) for all \( i \neq j \). Moreover, there are \( \mu_1, \mu_2 \in \mathcal{F}_A \) s.t.
\[
\alpha \cdot [q](t) \cdot \beta = \mathcal{F}_A \alpha' \cdot [q](t) \cdot \beta' = \mathcal{F}_A \alpha' \cdot [q'(t') \cdot \beta'
\]
— a contradiction. By an analogous argument for some \( x_j \) only occurring in the right-hand side of the rule for \( q' \) the first statement follows.

Assume for contradiction that the mappings \( \sigma \) and \( \sigma' \) in the corresponding rules (4) differ. Let \( k \) denote the minimal index so that \( \sigma(k) \neq \sigma'(k) \). W.l.o.g., we assume that \( \sigma'(k) < \sigma(k) \). By the first statement, \( n = n' \) and \( \{\sigma(1), \ldots, \sigma(n)\} = \{\sigma'(1), \ldots, \sigma'(n)\} \). Then there are \( \ell, \ell' > k \) such that
\[
\sigma'(k) = \sigma(\ell) < \sigma(k) = \sigma'(\ell')
\]
Let \( t = f(t_1, \ldots, t_n) \in \text{dom}(\iota(q)) \) be an input tree. For that we obtain
\[
\begin{align*}
\mu_0 &:= u_0 \cdot [q_k](t_{\sigma(1)}) \cdot \ldots \cdot u_{k-1} & \mu_1 &:= u_0' \cdot [q_k](t'_{\sigma'(1)}) \cdot \ldots \cdot u_{k-1}' \\
\mu_2 &:= u_e \cdot [q_k](t_{\sigma(k)}) \cdot \ldots \cdot u_n & \mu_1' &:= u'_k \cdot [q_k'](t'_{\sigma'(k)}) \cdot \ldots \cdot u'_{\ell'-1} \\
\mu_2' &:= u'_k \cdot [q_k'](t_{\sigma(k)}) \cdot \ldots \cdot u_n \\
\end{align*}
\]
Then for all input trees \( t' \in \text{dom}(\iota(q_k)), t'' \in \mathcal{L}(\text{dom}(\iota(q_k))) \),
\[
\alpha \cdot [q_k](t') \cdot [q_k](t'') \cdot \mu_2 \cdot \beta = \mathcal{F}_A \gamma' \cdot [q_k'](t'') \cdot [q_k'](t') \cdot [q_k'](t') \cdot \mu_2' \cdot \beta'
\]
Let \( \gamma' = \mu_0' \alpha' \alpha' \mu_0' \). Then
\[
[q_k](t') \cdot [q_k](t'') \cdot \mu_2 \cdot \beta = \mathcal{F}_A \gamma' \cdot [q_k'](t'') \cdot [q_k'](t') \cdot [q_k'](t') \cdot \mu_2' \cdot \beta'
\]
By Lemma 2, we obtain that for all \( w_1, w_2 \in \mathcal{L}(q_k) \) and \( v_1, v_2 \in \mathcal{L}(q_l) \), \( w_2 \cdot w_1 \in \mu_1 \cdot \langle p \rangle \cdot \mu_1 \) and \( v_1 \cdot v_2 \in \langle p \rangle \) for some primitive \( p \).

If \( \ell = k + 1 \), i.e., there is no further state between \( q_k(x_{\sigma(k)}) \) and \( q_{\ell}(x_{\sigma(\ell)}) \), then \( \mu_1 = \mathcal{F}_A u_k, \mathcal{L}(q_k) \subseteq w \cdot u_k \cdot \langle p \rangle \cdot u_k \) and \( \mathcal{L}(q_k) \subseteq \langle p \rangle \cdot v \) for some fixed \( w \in \mathcal{L}(q_k) \) and \( v \in \mathcal{L}(q_k) \). As \( \sigma(k) > \sigma'(k) = \sigma(\ell) \), this contradicts \( \mathcal{M} \) being ordered.

For the case that there is at least one occurrence of a state between \( q_k(x_{\sigma(k)}) \) and \( q_{\ell}(x_{\sigma(\ell)}) \), we show that for all \( \alpha_1, \alpha_2 \in u_k \cdot \mathcal{L}(q_{k+1}) \cdot \ldots \cdot u_{\ell-1} = \mathcal{F}_A \hat{L} \),
Lemma 6. Assume that $L_1, \ldots, L_n$ are neither empty nor singleton subsets of $F_A$ and $u_1, \ldots, u_{n-1} \in F_A$. Then there are $v_1, \ldots, v_n \in F_A$ such that

$$L_1 \cdot u_1 \cdot \ldots \cdot L_{n-1} \cdot u_{n-1} \cdot L_n \subseteq v \cdot \langle p \rangle$$

holds if and only if for $i = 1, \ldots, n$, $L_i \subseteq v_i \cdot \langle p_i \rangle$ with

$$p_n = F_A \cdot p$$

$$p_i = F_A \cdot (u_i \cdot v_{i+1}) \cdot p_{i+1} \cdot (u_i \cdot v_{i+1})^{-1} \quad \text{for } i < n$$

and

$$v^- \cdot v_1 \cdot u_1 \cdot \ldots \cdot v_{n-1} \cdot u_{n-1} \cdot v_n \in \langle p \rangle$$

Proof. Let $s_1 = \varepsilon$. For $i = 2, \ldots, n$ we fix some word $s_i \in L_1 \cdot u_1 \cdot L_2 \cdot \ldots \cdot L_{i-1} \cdot u_{i-1}$. Likewise, let $t_n = \varepsilon$ and for $i = 1, \ldots, n-1$ fix some word $t_i \in u_i \cdot L_{i+1} \cdot \ldots \cdot L_n$, and define $v_i = F_A \cdot s_i^- \cdot v \cdot t_i^-$. First assume that the inclusion (5) holds. Let $p_i' = F_A \cdot t_i \cdot p \cdot t_i^-$. Then for all $i$, $s_i \cdot L_i \cdot t_i \subseteq v \cdot \langle p \rangle$, and therefore

$$L_i \subseteq s_i^- \cdot v \cdot \langle p \rangle \cdot t_i^- = F_A \cdot s_i^- \cdot v \cdot t_i^- \cdot t_i \cdot \langle p \rangle \cdot t_i^- = F_A \cdot v_i \cdot \langle p_i' \rangle$$

We claim that $p_i' = p_i$ for all $i = 1, \ldots, n$. We proceed by induction on $n-i$. As $t_n = \varepsilon$, we have that $p_n' = p = p_n$. For $i < n$, we can rewrite $t_i = F_A \cdot u_i \cdot w_{i+1} \cdot t_{i+1}$ where $w_{i+1} \in L_{i+1}$ and thus is of the form $v_{i+1} \cdot p_{i+1}^{k_{i+1}}$ for some exponent $k_{i+1}$.
It remains to prove the inclusion (6). Since \( w_i \in L_i \), we have by (5) that \( v^{-1} \cdot u_1 \cdot \ldots \cdot u_{n-1} \cdot u_n \in \langle p \rangle \) holds. Now we calculate:

\[
v^{-1} \cdot u_1 \cdot u_2 \cdot \ldots \cdot u_{n-1} \cdot u_n = F_A \cdot v^{-1} \cdot u_1 \cdot p_1^{k_1} \cdot u_2 \cdot \ldots \cdot u_{n-1} \cdot v_n \cdot p_n^{k_n} = F_A \cdot v^{-1} \cdot u_1 \cdot u_2 \cdot p_2^{k_1+k_2} \cdot u_3 \cdot \ldots \cdot u_{n-1} \cdot v_n \cdot p_n^{k_n} = F_A \cdot v^{-1} \cdot u_1 \cdot \ldots \cdot u_{n-1} \cdot u_{n-1} \cdot v_n \cdot p_n^{k}
\]

where \( k = k_1 + \ldots + k_n \). Since \( p_n = p \), the claim follows.

The other direction of the claim of the lemma follows directly:

\[
 L_1 u_1 \ldots L_{n-1} u_{n-1} L_n \subseteq v_1 \cdot (p_1) \cdot u_1 \cdot \ldots \cdot v_{n-1} \cdot (p_{n-1}) \cdot u_{n-1} \cdot v_n \cdot (p_n) = F_A \cdot v_1 \cdot u_1 \cdot v_2 \cdot (p_2) \cdot u_2 \cdot \ldots \cdot v_{n-1} \cdot (p_{n-1}) \cdot u_{n-1} \cdot v_n \cdot (p_n) = F_A \cdot v_1 \cdot u_1 \cdot v_2 \cdot (p_2) \cdot u_2 \cdot \ldots \cdot v_{n-1} \cdot (p_{n-1}) \cdot u_{n-1} \cdot v_n \cdot (p_n) = F_A \cdot v_1 \cdot u_1 \cdot v_2 \cdot \ldots \cdot u_{n-1} \cdot v_n \cdot (p_n) = F_A \cdot v_1 \cdot u_1 \cdot v_2 \cdot \ldots \cdot u_{n-1} \cdot v_n \cdot (p) \subseteq v \cdot \langle p \rangle
\]

where the last inclusion follows from (6).

Let us call a non-empty, non-singleton language \( L \subseteq F_A \) periodic, if \( L \subseteq v \cdot \langle p \rangle \) for some \( v, p \in F_A \). Lemma 6 then implies that if a concatenation of languages and elements from \( F_A \) is periodic, then so must be all non-singleton component languages. In fact, the languages in the composition can then be arbitrarily permuted.

**Corollary 2.** Assume for non-empty, nonsingleton languages \( L_1, \ldots, L_n \subseteq F_A \) and \( u_1, \ldots, u_{n-1} \in F_A \) that property (5) holds. Then for every permutation \( \pi \), there are elements \( u_{\pi,0}, \ldots, u_{\pi,n} \in F_A \) such that

\[
 L_1 \cdot u_1 \cdot \ldots \cdot L_{n-1} \cdot u_{n-1} \cdot L_n = u_{\pi,0} \cdot L_{\pi(1)} \cdot u_{\pi,1} \cdot \ldots \cdot u_{\pi_{n-1}} \cdot L_{\pi(n)} \cdot u_{\pi,n}
\]

**Example 3.** We reconsider \( LTA \ M' \) and DTA \( B \) from Example 2. We observe that \( L(q_1) \subseteq a \cdot \langle ba \rangle \), \( L(q_2) \subseteq \langle ab \rangle \), and thus \( L(q_0) = L(q_1) \cdot b \cdot L(q_2) \subseteq \langle ab \rangle \). Accordingly, the rule for state \( q_0 \) and input symbol \( f \) is not ordered. Following the notation of Corollary 2, we find \( v_1 = a \), \( u_1 = b \) and \( v_2 = \varepsilon \), and the rule for \( q_0 \) and \( f \) can be reordered to

\[
 q_0(f(x_1, x_2)) \rightarrow ab \cdot q_2(x_1) \cdot b^{-}a^{-} \cdot q_1(x_2)b
\]

This example shows major improvements compared to the construction in [2]. Since we have inverses at hand, only local changes must be applied to the subsequence \( q_1(x_2) \cdot b \cdot q_2(x_1) \). In contrast to the construction in [2], neither auxiliary states nor further changes to the rules of \( q_1 \) and \( q_2 \) are required.

By Corollary 2, the order of occurrences of terms \( q_k(x_{\sigma(k)}) \) can be permuted in every subsequence \( q_i(x_{\sigma(j)}) \cdot u_i \cdot \ldots \cdot u_{j-1} \cdot L(q_j) \in u \cdot \langle p \rangle \) is periodic, to satisfy the requirements of an ordered \( LTA \). A sufficient
condition for that is, according to Lemma 6, that $L(q_k)$ is periodic for each $q_k$ occurring in that sub-sequence. Therefore we will determine the subset of all states $q$ where $L(q)$ is periodic, and if so elements $v_q, p_q$ such that $L(q) \subseteq v_q \cdot \langle p_q \rangle$. In order to do so we compute an abstraction of the sets $L(q)$ by means of a complete lattice which both reports constant values and also captures periodicity.

Let $D = 2^F$ denote the complete lattice of subsets of the free group $F$. We define a projection $\alpha : D \rightarrow D$ by $\alpha(\emptyset) = \emptyset$, $\alpha(\{g\}) = \{g\}$, and for languages $L$ with at least two elements,

$$\alpha(L) = \begin{cases} g\langle p \rangle & \text{if } L \subseteq g\langle p \rangle \text{ and } p \text{ is primitive} \\ F & \text{otherwise} \end{cases}$$

The projection $\alpha$ is a closure operator, i.e., is a monotonic function with $L \subseteq \alpha(L)$, and $\alpha(\alpha(L)) = \alpha(L)$. The image of $\alpha$ can be considered as an abstract complete lattice $D^\sharp$, partially ordered by subset inclusion. Thereby, the abstraction $\alpha$ commutes with least upper bounds as well as with the group operation. For that, we define abstract versions $\sqcup, \star : (D^\sharp)^2 \rightarrow D^\sharp$ of set union and the group operation by

$$A_1 \sqcup A_2 = \alpha(A_1 \cup A_2) \quad A_1 \star A_2 = \alpha(A_1 \cdot A_2)$$

In fact, "$\sqcup$" is the least upper bound operation for $D^\sharp$. The two abstract operators can also be more explicitly defined by:

$$\begin{align*}
\emptyset \sqcup L &= L \sqcup \emptyset = L \\
F \sqcup L &= L \sqcup F = F \\
\{g_1\} \sqcup \{g_2\} &= \begin{cases} \{g_1\} & \text{if } g_1 = g_2 \\ g_1 \cdot \langle p \rangle & \text{if } g_1 \neq g_2, p \text{ primitive root of } g_1^{-1} \cdot g_2 \end{cases} \\
\{g_1\} \sqcup g_2 \cdot \langle p \rangle &= g_2 \cdot \langle p \rangle \sqcup \{g_1\} = \begin{cases} g_2 \cdot \langle p \rangle & \text{if } g_1 \in g_2 \cdot \langle p \rangle \\ F & \text{otherwise} \end{cases} \\
g_1 \cdot \langle p_1 \rangle \sqcup g_2 \cdot \langle p_2 \rangle &= \begin{cases} g_1 \cdot \langle p_1 \rangle & \text{if } p_2 \in \langle p_1 \rangle \text{ and } g_2 \cdot g_1 \in \langle p_1 \rangle \\ F & \text{otherwise} \end{cases}
\end{align*}$$

$$\begin{align*}
\emptyset \star L &= L \star \emptyset = \emptyset \\
F \star L &= L \star F = F \quad \text{for } L \neq \emptyset \\
\{g_1\} \star \{g_2\} &= \{g_1 \cdot g_2\} \\
\{g_1\} \star g_2 \cdot \langle p \rangle &= (g_1 \cdot g_2) \cdot \langle p \rangle \\
g_1 \cdot \langle p \rangle \star \{g_2\} &= (g_1 \cdot g_2) \cdot \langle g_2^{-1} \cdot p \cdot g_2 \rangle \\
g_1 \cdot \langle p_1 \rangle \star g_2 \cdot \langle p_2 \rangle &= \begin{cases} (g_1 \cdot g_2) \cdot \langle p_2 \rangle & \text{if } g_2 \cdot p_1 \cdot g_2 \in \langle p_2 \rangle \\ F & \text{otherwise} \end{cases}
\end{align*}$$

**Lemma 7.** For all subsets $L_1, L_2 \subseteq F$, $\alpha(L_1 \cup L_2) = \alpha(L_1) \cup \alpha(L_2)$ and $\alpha(L_1 \cdot L_2) = \alpha(L_1) \star \alpha(L_2)$. 

We conclude that $\alpha$ in fact represents a precise abstract interpretation in the sense of [9]. Accordingly, we obtain:

**Lemma 8.** For every $\text{LT}_A$ $M$ and DTA $B$ with compatible map $\iota$, the sets $\alpha(\mathcal{L}(q))$, $q$ state of $M$, can be computed in polynomial time.

**Proof.** We introduce one unknown $X_q$ for every state $q$ of $M$, and one constraint for each rule of $M$ of the form (3) where $\delta(\iota(q), f)$ is defined in $B$. This constraint is given by:

$$X_q \supseteq u_0 \ast X_{q_1} \ast \ldots \ast u_{n-1} \ast X_{q_n} \ast u_n$$  \hspace{1cm} (7)

As the right-hand sides of the constraints (7) all represent monotonic functions, the given system of constraints has a least solution. In order to obtain this solution, we consider for each state $q$ of $M$, the sequence $X_q(i)$, $i \geq 0$ of values in $D^\sharp$ where $X_q(0) = \emptyset$, and for $i > 0$, we set $X_q(i)$ as the least upper bound of the values obtained from the constraints with left-hand side $X_q$ of the form (7) by replacing the unknowns $X_{q_j}$ on the right-hand side with the values $X_{q_j}^{(i-1)}$. By induction on $i \geq 0$, we verify that for all states $q$ of $M$,

$$X_q(i) = \alpha(\mathcal{L}(i)(q))$$

holds. Note that the induction step thereby, relies on Lemma 7.

As each strictly increasing chain of elements in $D^\sharp$ consists of at most four elements, we have that the least solution of the constraint system is attained after at most $3 \cdot N$ iterations, if $N$ is the number of states of $M$, i.e., for each state $q$ of $M$, $X_q^{(3N)} = X_q(i)$ for all $i \geq 3N$. The elements of $D^\sharp$ can be represented by SLPs where the operations $\ast$ and $\sqcup$ run in polynomial time, cf. Lemma 3. Since each iteration requires only a polynomial number of operations $\ast$ and $\sqcup$, the statement of the lemma follows.

We now exploit the information provided by the $\alpha(\mathcal{L}(q))$ to remove trivial states as well as order subsequences of right-hand sides which are periodic.

**Theorem 1.** Let $B$ be a DTA such that $\mathcal{L}(B) \neq \emptyset$. For every $\text{LT}_A$ $M$ with compatible map $\iota$, an $\text{LT}_A$ $M'$ with compatible map $\iota'$ can be constructed in polynomial time such that

1. $M$ and $M'$ are equivalent relative to $B$;
2. $M'$ has no trivial states;
3. $M'$ is ordered.

**Proof.** By Lemma 8, we can, in polynomial time, determine for every state $q$ of $M$, the value $\alpha(\mathcal{L}(q))$. We use this information to remove from $M$ all trivial states. W.l.o.g., assume that the axiom of $M$ is given by $u_0 \cdot q_0(x_0) \cdot u_1$. If the state $q_0$ occurring in the axiom of $M$ is trivial with $\mathcal{L}(q_0) = \{v\}$, then $M_1$ has no states or rules, but the axiom $u_0 \cdot v \cdot u_1$.

Therefore now assume that $q_0$ is non-trivial. We then construct an $\text{LT}_A$ $M_1$ whose set of states $Q_1$ consists of all non-trivial states $q$ of $M$ where the compatible map $\iota_1$ of $M_1$ is obtained from $\iota$ by restriction to $Q_1$. Since $\mathcal{L}(M) \neq \emptyset$, 

...
the state of $M$ occurring in the axiom is non-trivial. Accordingly, the axiom of $M$ is also used as axiom for $M_1$. Consider a non-trivial state $q$ of $M$ and $f \in \Sigma$. If $\delta(v(q),f)$ is not defined $M_1$ has the rule $q(f(x_1,\ldots,x_m) \rightarrow \perp$. Assume that $\delta(v(q),f)$ is defined and $M$ has a rule of the form (3). Then $M_1$ has the rule

$$q(f(x_1,\ldots,x_m) \rightarrow u_0 \cdot g_1 \cdot \ldots \cdot u_{n-1} \cdot g_n \cdot u_n$$

where for $i = 1,\ldots,n$, $g_i$ equals $q_i(x_{\sigma(i)})$ if $q_i$ is non-trivial, and equals the single word in $\mathcal{L}(q_i)$ otherwise. Obviously, $M$ and $M_1$ are equivalent relative to $B$ where $M_1$ now has no trivial states, while for every non-trivial state $q$, the semantics of $q$ in $M$ and $M_1$ are the same relative to $B$. Our goal now is to equivalently rewrite the right-hand side of each rule of $M_1$ so that the result is ordered. For each state $q$ of the $\text{LT}_A$ we determine whether there are $v,p \in B^*$ such that $\mathcal{L}(q) \subseteq v \cdot p$, cf. Lemma 8. So consider a rule of $M_1$ of the form (3). By means of the values $\alpha(\mathcal{L}(q_i)), i = 1,\ldots,n$, together with the abstract operation "$\cdot\$", we can determine maximal intervals $[i,j]$ such that $\mathcal{L}(q_i) \cdot u_i \cdot \ldots \cdot u_{j-1} \cdot \mathcal{L}(q_j)$ is periodic, i.e., $\alpha(\mathcal{L}(q_i)) \cdot u_i \cdot \ldots \cdot u_{j-1} \cdot * \alpha(\mathcal{L}(q_j)) \subseteq v \cdot p$ for some $v,p \in \mathcal{F}_A$.

We remark that these maximal intervals are necessarily disjoint. By Corollary 2, for every permutation $\pi : [i,j] \rightarrow [i,j]$, elements $u',u'_i,\ldots,u'_j,u'' \in \mathcal{F}_A$ can be found so that $q_i(x_{\sigma(i)}) \cdot u_i \cdot \ldots \cdot u_{j-1} \cdot q_j(x_{\sigma(j)})$ is equivalent to $u' \cdot q_{\pi(i)}(x_{\sigma(\pi(i))}) \cdot u'_i \cdot \ldots \cdot u'_{j-1} \cdot q_{\pi(j)}(x_{\sigma(\pi(j))}) \cdot u''$.

In particular, this is true for the permutation $\pi$ with $\sigma(\pi(i)) < \ldots < \sigma(\pi(j))$. Assuming that all group elements are represented as SLPs, the overall construction runs in polynomial time.

In summary, we arrive at the main theorem of this paper.

**Theorem 2.** The equivalence of $\text{LT}_A$s relative to some DTA $B$ can be decided in polynomial time.

**Proof.** Assume we are given $\text{LT}_A$s $M,M'$ with compatible maps (relative to $B$). By Theorem 1, we may w.l.o.g. assume that $M$ and $M'$ both have no trivial states and are ordered. It can be checked in polynomial time whether or not $M$ and $M'$ are same-ordered. If they are not, then by Lemma 5, they cannot be equivalent relative to $B$. Therefore now assume that $M$ and $M'$ are same-ordered. Then their equivalence relative to $B$ is decidable in polynomial time by Corollary 1. Altogether we thus obtain a polynomial decision procedure for equivalence of $\text{LT}_A$s relative to some DTA $B$.

**4 Conclusion**

We have shown that equivalence of $\text{LT}_A$s relative to a given DTA $B$ can be decided in polynomial time. For that, we considered total transducers only, but defined the domain of allowed input trees separately by means of the DTA. This does not impose any restriction of generality, since any (possibly partial) linear deterministic top-down tree transducer can be translated in polynomial time to
a corresponding total \( LT_A \) together with a corresponding DTA (see, e.g., [11]). The required constructions for \( LT_A \) which we have presented here, turn out to be more general than the constructions provided in [2] since they apply to transducers which may not only output symbols \( a \in A \), but also their inverses \( a^{-} \). At the same time, they are simpler and easier to be proven correct due to the combinatorial and algebraic properties provided by the free group.

Acknowledgements. We also like to thank the anonymous reviewers for their detailed comments and valuable advice.

References

1. Boiret, A.: Normal form on linear tree-to-word transducers. In: Dediu, A.-H., Janoušek, J., Martín-Vide, C., Truthe, B. (eds.) LATA 2016. LNCS, vol. 9618, pp. 439–451. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30000-9_34
2. Boiret, A., Palenta, R.: Deciding equivalence of linear tree-to-word transducers in polynomial time. In: Brlek, S., Reutenauer, C. (eds.) DLT 2016. LNCS, vol. 9840, pp. 355–367. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53132-7_29
3. Laurence, G., Lemay, A., Niehren, J., Staworko, S., Tommasi, M.: Normalization of sequential top-down tree-to-word transducers. In: Dediu, A.-H., Inenaga, S., Martín-Vide, C. (eds.) LATA 2011. LNCS, vol. 6638, pp. 354–365. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21254-3_28
4. Laurence, G., Lemay, A., Niehren, J., Staworko, S., Tommasi, M.: Learning sequential tree-to-word transducers. In: Dediu, A.-H., Martín-Vide, C., Sierra-Rodríguez, J.-L., Truthe, B. (eds.) LATA 2014. LNCS, vol. 8370, pp. 490–502. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04921-2_40
5. Löbel, R., Luttenberger, M., Seidl, H.: Equivalence of linear tree transducers with output in the free group. CoRR abs/2001.03480 (2020). http://arxiv.org/abs/2001.03480
6. Lohrey, M.: Word problems on compressed words. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 906–918. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27836-8_76
7. Lohrey, M.: The Compressed Word Problem for Groups. SM. Springer, New York (2014). https://doi.org/10.1007/978-1-4939-0748-9
8. Lyndon, R.C., Schupp, P.E.: Combinatorial Group Theory. CM, vol. 89. Springer, Heidelberg (2001). https://doi.org/10.1007/978-3-642-61896-3
9. Müller-Olm, M.: Variations on Constants. LNCS, vol. 3800. Springer, Heidelberg (2006). https://doi.org/10.1007/11871743
10. Plandowski, W.: Testing equivalence of morphisms on context-free languages. In: van Leeuwen, J. (ed.) ESA 1994. LNCS, vol. 855, pp. 460–470. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-409431
11. Seidl, H., Maneth, S., Kemper, G.: Equivalence of deterministic top-down tree-to-string transducers is decidable. J. ACM 65(4), 21:1–21:30 (2018). https://doi.org/10.1145/3182653
12. Sénizergues, G.: T(A) = T(B)? In: Wiedermann, J., van Emde Boas, P., Nielsen, M. (eds.) ICALP 1999. LNCS, vol. 1644, pp. 665–675. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48523-6_63
13. Staworko, S., Laurence, G., Lemay, A., Niehren, J.: Equivalence of deterministic nested word to word transducers. In: Kutyłowski, M., Charatonik, W., Gębala, M. (eds.) FCT 2009. LNCS, vol. 5699, pp. 310–322. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03409-1_28

14. Tomita, E., Seino, K.: A direct branching algorithm for checking the equivalence of two deterministic pushdown transducers, one of which is real-time strict. Theory Comput. Sci. 64(1), 39–53 (1989). https://doi.org/10.1016/0304-3975(89)90096-0