Determinacy and Rewriting of Functional Top-Down and MSO Tree Transformations[☆]

M. Benedikt[a], J. Engelfriet[b], S. Maneth[c,*]

[a] *Department of Computer Science, University of Oxford, United Kingdom*
[b] *Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands*
[c] *School of Informatics, University of Edinburgh, United Kingdom*

Abstract

A query is determined by a view, if the result of the query can be reconstructed from the result of the view. We consider the problem of deciding for two given (functional) tree transformations, whether one is determined by the other. If the view transformation is induced by a tree transducer that may copy, then determinacy is undecidable. For a large class of noncopying views, namely compositions of extended linear top-down tree transducers, we show that determinacy is decidable, where queries are either deterministic top-down tree transducers (with regular look-ahead) or deterministic MSO tree transducers. We show that if a query is determined by a view, then it can be rewritten into a query working over the view which is in the same class of transducers as the query. The proof relies on the decidability of equivalence for the considered classes of queries, and on their closure under composition.

Keywords: view-query determinacy, top-down tree transducers, MSO definable tree transducers

1. Introduction

Given a transformation between data structures, i.e., a function from data structures to data structures, a basic question is what sort of information it preserves. In some contexts, one desires a transformation that is "fully information-preserving": one can recover the input structure from the output structure. In other cases it may be acceptable, or even important, to hide certain pieces of information in the input; but necessarily there is *some* important information in the input that must be recoverable from the output. This notion has been studied in the database community [1, 2]: a query q is determined by another query v if there exists a function f

[☆] Benedikt and Maneth were supported by the Engineering and Physical Sciences Research Council project "Enforcement of Constraints on XML streams" (EPSRC EP/G004021/1).

[*] Corresponding author

Email addresses: michael.benedikt@cs.ox.ac.uk (M. Benedikt), j.engelfriet@liacs.leidenuniv.nl (J. Engelfriet), smaneth@inf.ed.ac.uk (S. Maneth)

such that $q = v \circ f$, where $\circ$ denotes sequential composition, i.e., $q(s) = f(v(s))$ for every input s. The query v is referred to as "view". Note that nothing is said about how efficiently f can be computed (or if it can be computed at all). We can then strengthen determinacy by requiring the function f to lie within a certain class $\mathcal{R}$; then f is a "rewriting in $\mathcal{R}$". These notions have received considerable attention in the database setting [1, 2, 3, 4, 5].

In this paper we study determinacy and rewriting for classes of tree transformations (or, tree translations), as in [6, 7]. Injectivity is undecidable for deterministic top-down tree transducers [8, 9]; hence, one cannot decide if the identity query is determined by such a transducer. This even holds for transducers that copy the input tree only once. Therefore, as in [6, 7], we restrict our attention to views induced by *linear* tree transducers, i.e., noncopying tree transducers. For the same reason we restrict to a single view (while in database research, normally multiple views are considered). Our main result is that determinacy is decidable for views that are compositions of extended linear top-down tree transducers (with regular look-ahead) and for queries that are either deterministic top-down tree transducers (with regular look-ahead) or deterministic MSO tree transducers (where MSO means "definable in Monadic Second-Order logic"). Extended transducers generalize the left-hand sides of conventional finite-state tree transducers, from one input symbol to an arbitrary "pattern tree". They were introduced in [10] and have recently been studied in [11, 12, 13, 14]. Extended linear transducers are convenient because (1) they are more powerful than ordinary linear top-down or bottom-up transducers and (2) they allow to elegantly capture the inverses of translations.

As an example of determinacy, consider the transformation v taking binary trees as input, with internal nodes labeled a, b, c, and leaves labeled l. It relabels the b nodes as a nodes, and otherwise copies the tree as is. A linear top-down tree transducer realizing this translation v has a single state p and these translation rules:

$$\begin{array}{llll} p(a(x,y)) & \to & a(p(x),p(y)) & \qquad p(b(x,y)) \to a(p(x),p(y)) \\ p(c(x,y)) & \to & c(p(x),p(y)) & \qquad p(l) \to l \end{array}$$

Let q_0 be the identity query, i.e., $q_0(s) = s$ for every input tree s. Since information about the (labels of) b nodes and a nodes is lost in the view v, from the output $v(s)$ we cannot determine the answer s to the query q_0. In contrast, information about the l nodes and their relationship to c nodes is maintained. Consider, for example, the following query q_1 that removes a and b nodes (and their second subtrees), realized by the top-down tree transducer with the following translation rules:

$$\begin{array}{llll} p(a(x,y)) & \to & p(x) & \qquad p(b(x,y)) \to p(x) \\ p(c(x,y)) & \to & c(p(x),p(y)) & \qquad p(l) \to l \end{array}$$

Clearly, q_1 is determined by v. In fact, $q_1 = v \circ f$ where f is realized by the same translation rules as q_1, minus the rule $p(b(x,y)) \to p(x)$. Our algorithm can decide that q_0 is not determined by v and that q_1 is.

Our decision procedure for determinacy establishes several results that are interesting on their own. For a view v realized by an extended linear top-down tree transducer, its inverse v^{-1} is a binary relation on trees. Our approach converts v^{-1} into a composition of two nondeterministic translations, a translation τ_1 of a very simple form and a translation τ_2 in the same class as v. We then construct uniformizers u_1, u_2 of τ_1, τ_2 and compose them to form a uniformizer u of v^{-1}. A *uniformizer* of a binary relation R is a function u such that $u \subseteq R$ and u has the same domain as R; thus u selects one of the possibly several elements that R associates with an element of its domain. It is easy to see that a query q is determined by v if and only if $q = v \circ u \circ q$, where u is any uniformizer of v^{-1}. We show that if q is a deterministic top-down or MSO tree translation, then so is $v \circ u \circ q$. This is achieved by proving that u_1, u_2, and v are deterministic top-down *and* MSO tree translations. Since our two query classes are closed under composition and $u = u_1 \circ u_2$, this shows that $v \circ u \circ q$ is in the same class as q. We then decide $q = v \circ u \circ q$, and hence determinacy, making use of the decidability of equivalence for deterministic top-down or MSO tree translations ([15, 16] or [17]). The same proof also shows that if q is determined by v, then $u \circ q$ is a rewriting belonging to the same class as the query q.

The results of this paper were first presented at MFCS 2013 (see [18]).

Related Work. The notion of view-query determinacy was introduced by Segoufin and Vianu in [1]. They focus on relational queries definable in first-order logic and show that if such queries are determined over arbitrary structures, then they can be rewritten in first-order logic, but that if they are determined over finite structures, they may require a much more powerful type of relational query to be rewritten. Nash, Segoufin, and Vianu [2] summarize a number of other results on the relational case. Due to the differing data models and notions of equality used in relational queries and tree transducers, results on determinacy for queries in the relational case do not (directly) apply to tree transducers, and vice versa. In the context of unranked trees, determinacy is considered in Groz's thesis [19] for XML views and queries, see also [20]. Two notions of determinacy are considered, depending on whether or not the output trees preserve provenance information (i.e., node identities) from the input document. It is shown that both notions of determinacy are undecidable for views and queries defined using a transformation language that can select subtrees using regular XPath filters. On the positive side, it is shown that if the views are "interval-bounded" – there is a bound on the number of consecutive nodes skipped along a path – then determinacy can be tested effectively.

The most related work is that of Hashimoto *et al.* [6, 7], who consider the determinacy problem (and rewriting) explicitly for tree transducers, in particular bottom-up tree transducers. They solve the problem for views that are realized by extended linear bottom-up tree transducers and queries that are realized by bottom-up tree transducers. Their approach is to decide determinacy by testing functionality of the inverse of the view composed with the query. To this end they generalize

the functionality test for bottom-up tree transducers in [21] to extended bottom-up tree transducers with "grafting" (needed for the inverse of the view). Our main result generalizes the above result of [6, 7], and provides an alternative proof of it. In fact, the classes of views and queries for which we can decide determinacy are larger than those of [6, 7]. In particular, our class of queries is much larger than their class, as top-down tree transducers and MSO tree transducers are much more powerful than bottom-up tree transducers (for defining partial functions). As explained above, for a given view v and query q, rather than testing functionality of $v^{-1} \circ q$, our approach is to test the equivalence $q = v \circ u \circ q$, where u is a uniformizer of v^{-1}. This implies that, for the fixed class of views, our approach is in fact applicable to any class of queries that has a decidable equivalence problem and is closed under left-composition with every v and u. This is discussed in more detail in Section 7.2 and in the Conclusion.

2. Preliminaries

For $k \in \mathbb{N} = \{0, 1, \ldots\}$ let $[k]$ denote the set $\{1, \ldots, k\}$. For a binary relation R we denote by R^{-1} its inverse $\{(y, x) \mid (x, y) \in R\}$. For a set A, $R(A) = \{y \mid \exists x \in A : (x, y) \in R\}$, and for an element x, $R(x) = R(\{x\})$. If $R \subseteq B \times C$ for sets B and C, then the range of R is $\mathsf{ran}(R) = R(B)$ and the domain of R is $\mathsf{dom}(R) = R^{-1}(C)$. For two relations R and S we denote their composition "R followed by S" by $R \circ S$, i.e., for every element x, $(R \circ S)(x) = S(R(x))$. Note that this is in contrast to the conventional use of $\circ$. If $\mathcal{R}, \mathcal{S}$ are classes of binary relations and $n \geq 1$, then $\mathcal{R} \circ \mathcal{S} = \{R \circ S \mid R \in \mathcal{R}, S \in \mathcal{S}\}$, $\mathcal{R}^n = \{R_1 \circ \cdots \circ R_n \mid R_i \in \mathcal{R} \text{ for } i \in [n]\}$, $\mathcal{R}^* = \bigcup_{m \geq 1} \mathcal{R}^m$, and $\mathcal{R}^{-1} = \{R^{-1} \mid R \in \mathcal{R}\}$. A relation R is *functional* if it is a partial function, i.e., if $R(x)$ is either empty or a singleton, for every element x. If $\mathcal{R}$ is a class of relations, then fu-$\mathcal{R}$ denotes the class of all partial functions in $\mathcal{R}$. The identity on a set A is the function $\mathsf{id}_A = \{(x, x) \mid x \in A\}$. For a relation R and a set A, the domain restriction of R to A is $\mathsf{id}_A \circ R = \{(x, y) \in R \mid x \in A\}$, and the range restriction of R to A is $R \circ \mathsf{id}_A = \{(x, y) \in R \mid y \in A\}$.

Determinacy. We define determinacy and rewritability, following [2]. Let $\mathcal{Q}, \mathcal{V}$ be classes of partial functions and let $q \in \mathcal{Q}$ and $v \in \mathcal{V}$. We say that q *is determined by* v, if there exists a partial function f such that $q = v \circ f$. Note that the latter means that the domains of q and $v \circ f$ coincide, and that $q(s) = f(v(s))$ for each s in that domain. *Determinacy for* $\mathcal{Q}$ *under* $\mathcal{V}$ is the problem that takes as input $q \in \mathcal{Q}$ and $v \in \mathcal{V}$ and outputs "yes" if q is determined by v, and "no" otherwise. Determinacy says that there is a functional dependency of q on v, with no limit on how complex it is to reconstruct the answer to q from the answer to v. A finer notion requires that the reconstruction be in a given class: a class $\mathcal{R}$ of partial functions is *complete for* $\mathcal{V}$*-to-*$\mathcal{Q}$ *rewritings*, if for every $q \in \mathcal{Q}$ and $v \in \mathcal{V}$ such that q is determined by v, there is an $f \in \mathcal{R}$ with $q = v \circ f$.

Trees, tree homomorphisms, and tree automata. A ranked alphabet consists of a finite set Σ together with a mapping $\mathsf{rank}_\Sigma : \Sigma \to \mathbb{N}$. We write $a^{(k)}$ to denote

that $\mathsf{rank}_\Sigma(a) = k$ and define $\Sigma^{(k)}$ as the set $\{a \in \Sigma \mid \mathsf{rank}_\Sigma(a) = k\}$. The set of (ranked, ordered, node-labeled, finite) trees over Σ, denoted by T_Σ, is the set of strings defined recursively as the smallest set T such that $a(s_1, \ldots, s_k) \in T$ if $a \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T$. For a tree $a()$ we simply write a. For a set T' of trees, we denote by $T_\Sigma(T')$ the smallest set of trees T containing T' such that $a(s_1, \ldots, s_k) \in T$ if $a \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T$. Thus, $T_\Sigma = T_\Sigma(\emptyset)$. A subset of T_Σ is called a tree language over Σ.

We represent the nodes of a tree as usual by Dewey notation, i.e., by strings of positive natural numbers. The empty string ε represents the root node and, for $i \in \mathbb{N}_+ = \{1, 2, \ldots\}$, vi represents the ith child of the node v. Every node v of a tree s has a label in Σ, denoted $\mathsf{lab}(s, v)$. Formally, the set $V(s) \subseteq \mathbb{N}_+^*$ of nodes of the tree s is inductively defined as: $V(s) = \{\varepsilon\} \cup \{iv \mid i \in [k], v \in V(s_i)\}$ if $s = a(s_1, \ldots, s_k)$, $a \in \Sigma^{(k)}$, and $s_1, \ldots, s_k \in T_\Sigma$; moreover, $\mathsf{lab}(s, \varepsilon) = a$ and $\mathsf{lab}(s, iv) = \mathsf{lab}(s_i, v)$. The subtree of s rooted at $v \in V(s)$ is denoted by s/v; formally, $s/\varepsilon = s$ and if $s = a(s_1, \ldots, s_k)$ then $s/iv = s_i/v$. The size of a tree is the number of its nodes.

Let $T_1, \ldots, T_n$ be sets of trees. For trees $s_1, \ldots, s_n$ that are not subtrees of each other, and a tree s, we denote by $s[s_i \leftarrow T_i \mid i \in [n]]$ the set of trees obtained from s by replacing each occurrence of a subtree s_i of s by a tree from T_i (where different occurrences of s_i need not be replaced by the same tree). If every T_i is a singleton $\{t_i\}$, then we write $s[s_i \leftarrow t_i \mid i \in [n]]$.

We fix a countably infinite set $X = \{x_1, x_2, \ldots\}$ of *variables*, which are assumed to have rank 0. For $k \in \mathbb{N}$, let $X_k = \{x_1, \ldots, x_k\}$. A tree $s \in T_\Sigma(X)$ is *linear* if each variable x_i occurs at most once in s. For $k \in \mathbb{N}$, an *X_k-context* (over Σ) is a tree C in $T_\Sigma(X_k)$ such that each variable $x_i \in X_k$ occurs exactly once in C. For such a context C and trees $s_1, \ldots, s_k$, $C[s_1, \ldots, s_k]$ denotes the tree $C[x_i \leftarrow s_i \mid i \in [k]]$. For a ranked alphabet Q with $Q^{(1)} = Q$ we denote by $Q(X_k)$ the set of trees $\{q(x_i) \mid q \in Q, i \in [k]\}$.

Let Σ and Δ be ranked alphabets. A *tree homomorphism* f from T_Σ to T_Δ is given by a mapping that assigns a tree $f_a \in T_\Delta(X_k)$ to every $a \in \Sigma^{(k)}$, $k \geq 0$. The mapping f is then defined by $f(a(s_1, \ldots, s_k)) = f_a[x_i \leftarrow f(s_i) \mid i \in [k]]$ for every $a \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T_\Sigma$. The tree homomorphism f is *linear* if f_a is linear, i.e., each variable x_i occurs at most once in f_a, *nondeleting* if every x_i, $i \in [k]$, occurs at least once in f_a, and *nonerasing* if $f_a \notin X_k$, for all $a \in \Sigma^{(k)}$ with $k \geq 0$.

A (deterministic bottom-up) *finite tree automaton* (fta, for short) over Σ is a tuple $A = (P, \Sigma, F, \delta)$ where P is a finite set of states, Σ is a ranked alphabet, $F \subseteq P$ is the set of final states, and δ is the transition function. For every $a \in \Sigma^{(k)}$, $k \geq 0$, and $p_1, \ldots, p_k \in P$, $\delta(a, p_1, \ldots, p_k)$ is an element of P. The transition function δ gives rise to a mapping δ^* from T_Σ to P. It is defined by $\delta^*(a(s_1, \ldots, s_k)) = \delta(a, \delta^*(s_1), \ldots, \delta^*(s_k))$ for $a \in \Sigma^{(k)}$, $k \geq 0$, and $s_1, \ldots, s_k \in T_\Sigma$. For convenience, we denote the function δ^* as well by δ. Thus, for a node v of an input tree s, $\delta(s/v)$ is the state reached by A at node v, when A processes s. The language accepted by A is $L(A) = \{s \in T_\Sigma \mid \delta(s) \in F\}$. A tree language is *regular* if it can be accepted by an fta.

We will use the following elementary and well-known fact.

Lemma 1. Let $A_1, \ldots, A_n$ be ftas over Σ, $n \geq 1$. There exist ftas $A'_1, \ldots, A'_n$ over Σ such that

(1) $A'_1, \ldots, A'_n$ have the same set of states and the same transition function, and
(2) $L(A'_i) = L(A_i)$ for every $i \in [n]$.

Proof. Let $A_i = (P_i, \Sigma, F_i, \delta_i)$ for every $i \in [n]$. We now define $A'_i = (P', \Sigma, F'_i, \delta')$ to be the usual product of A_i and all automata $(P_j, \Sigma, P_j, \delta_j)$ for $j \neq i$. That means that $P' = P_1 \times \cdots \times P_n$, $F'_i = P_1 \times \cdots \times P_{i-1} \times F_i \times P_{i+1} \times \cdots \times P_n$, and, for $a \in \Sigma^{(k)}$,

$$\delta'(a, (p_{1,1}, \ldots, p_{n,1}), \ldots, (p_{1,k}, \ldots, p_{n,k})) = \\ (\delta_1(a, p_{1,1}, \ldots, p_{1,k}), \ldots, \delta_n(a, p_{n,1}, \ldots, p_{n,k})).$$

Thus, A'_i ignores all components of a state except the ith. It should therefore be clear that $L(A'_i) = L(A_i)$. □

For an fta $A = (P, \Sigma, F, \delta)$ we extend the transition function δ to trees in $T_\Sigma(X_k)$, in particular to X_k-contexts, as follows. Let $p_1, \ldots, p_k \in P$. For $i \in [k]$, we define $\delta(x_i, p_1, \ldots, p_k) = p_i$. For $a \in \Sigma^{(n)}$, $n \geq 0$, and $s_1, \ldots, s_n \in T_\Sigma(X_k)$, we define $\delta(a(s_1, \ldots, s_n), p_1, \ldots, p_k) = \delta(a, \delta(s_1, p_1, \ldots, p_k), \ldots, \delta(s_n, p_1, \ldots, p_k))$. Clearly, for an X_k-context C and trees $s_1, \ldots, s_k \in T_\Sigma$, $\delta(C[s_1, \ldots, s_k]) = \delta(C, p_1, \ldots, p_k)$ where $\delta(s_i) = p_i$ for $i \in [k]$.

Convention: All lemmas, propositions, theorems, etc., stated in this paper (except in Section 5) are *effective*.

3. Extended Top-Down Tree Transducers

In this section we define extended top-down tree transducers (with regular look-ahead), prove a normal form for them, and discuss a number of their properties. Extended top-down tree transducers are studied in, e.g., [10, 11, 12, 13, 14].

3.1. Definition

An *extended top-down tree transducer with regular look-ahead* (ET$^{\mathrm{R}}$ transducer, for short) is a tuple $M = (Q, \Sigma, \Delta, I, R, A)$ where Q is a ranked alphabet of states all of rank 1, Σ and Δ are ranked alphabets of input and output symbols, respectively, $I \subseteq Q$ is the set of initial states, $A = (P, \Sigma, F, \delta)$ is an fta called the look-ahead automaton, and R is a finite set of rules of the form

$$q(C) \to \zeta \langle p_1, \ldots, p_k \rangle$$

where $q \in Q$, $k \geq 0$, $C \neq x_1$ is an X_k-context over Σ, $\zeta \in T_\Delta(Q(X_k))$, and $p_1, \ldots, p_k \in P$. For every state $q \in Q$ we define the *q-translation* $[\![q]\!]_M \subseteq T_\Sigma \times T_\Delta$ as follows. For an input tree $s \in T_\Sigma$, the q-translation $[\![q]\!]_M(s)$ of s is the smallest set of trees $T \subseteq T_\Delta$ such that for every rule $q(C) \to \zeta \langle p_1, \ldots, p_k \rangle$ and all

$s_1, \ldots, s_k \in T_\Sigma$, if $s = C[s_1, \ldots, s_k]$ and $\delta(s_i) = p_i$ for every $i \in [k]$, then T contains the set of trees

$$\zeta[q'(x_i) \leftarrow [\![q']\!]_M(s_i) \mid q' \in Q, i \in [k]].$$

The *translation* $[\![M]\!]$ *realized by* M is the binary relation

$$\{(s, t) \in T_\Sigma \times T_\Delta \mid s \in L(A), t \in \cup_{q \in I} [\![q]\!]_M(s)\}.$$

The class of all translations realized by ET^R transducers is denoted ET^R (and similarly for other types of transducers). Two ET^R transducers M_1 and M_2 are *equivalent* if $[\![M_1]\!] = [\![M_2]\!]$.

Since we do not allow the left-hand side of a rule to be of the form $q(x_1)$ with $q \in Q$, the application of a rule always consumes part of the input tree, i.e., our ET^R transducers are "epsilon-free" (cf. [11, 12, 14]).

The ET^R transducer M is *linear* (or noncopying), if the right-hand side ζ of each rule is linear, i.e., each variable x_i occurs at most once in ζ. We use "L" to abbreviate "linear", i.e., an ELT^R transducer is a linear ET^R transducer, and so ELT^R is the class of translations realized by linear ET^R transducers.

Transducers *without look-ahead* are defined by transducers with a trivial one-state look-ahead automaton (accepting T_Σ); this is indicated by omitting the superscript "R" for transducers and classes. In the notation for such transducers we omit the look-ahead automaton and we omit the sequence of look-ahead states at the end of each rule.

The ET^R transducer M is an (ordinary, not extended) *top-down tree transducer with regular look-ahead* (T^R transducer) if the left-hand side context C of each of its rules contains exactly one symbol in Σ, more precisely, each rule is of the form $q(a(x_1, \ldots, x_k)) \to \zeta\,\langle p_1, \ldots, p_k\rangle$ with $a \in \Sigma^{(k)}$ and $k \geq 0$.

A T^R transducer M is *finite-copying* (a $\text{T}^\text{R}_\text{fc}$ transducer) if each input node is translated only a bounded number of times. Formally this means there exists a number K such that for every $p \in P$, $s \in T_\Sigma(\{\Box\})$, and $t \in [\![M_p]\!](s)$, if $\Box$ occurs exactly once in s, then $\Box$ occurs $\leq K$ times in t; here we assume that $\Box$ is a new input and output symbol of rank 0, and that M_p is M extended with the look-ahead transition $\delta(\Box) = p$ and the rules $q(\Box) \to \Box$ for every state q.

A T^R transducer is *deterministic* if it has exactly one initial state and for each q, a, and $\langle p_1, \ldots, p_k\rangle$ it has at most one rule as above. Determinism is denoted by the letter "D", thus we have DT^R and $\text{DT}^\text{R}_\text{fc}$ transducers. Note that their translations are functional.

Example 2. Let $\Sigma = \Delta = \{a^{(3)}, b^{(2)}, e^{(0)}\}$. We will define an ELT^R transducer $M = (Q, \Sigma, \Delta, I, R, A)$ such that

$$[\![M]\!] = \{(a(b(s_1, s_2), b(s_3, s_4), e)\,,\; a(s_1, s_2, s_3)) \mid s_1, s_2, s_3, s_4 \in T_{\{b,e\}}\}.$$

This translation cannot be realized by an LT^R transducer (as can easily be shown using the proof of [12, Lemma 4.7] and the fact that LT^R is closed under composition [22, Theorem 2.11]). We first define the look-ahead automaton of M.

Let $A = (P, \Sigma, F, \delta)$ with $P = F = \{p, \overline{p}\}$ and with $\delta(e) = p$, $\delta(b, p, p) = p$ and $\delta(\dots) = \overline{p}$ in all remaining cases. Thus, for $s \in T_\Sigma$, $\delta(s) = p$ if $s \in T_{\{b,e\}}$, and $\delta(s) = \overline{p}$ otherwise. The set of states of M is $Q = \{q_0, q\}$, where q_0 is the unique initial state, i.e., $I = \{q_0\}$. The rules of M are defined as follows. First, R contains the rule

$$q_0(a(b(x_1, x_2), b(x_3, x_4), e)) \to a(q(x_1), q(x_2), q(x_3))\ \langle p, p, p, p\rangle.$$

Second, R contains the rules $q(b(x_1, x_2)) \to b(q(x_1), q(x_2))\ \langle p', p''\rangle$ for all $p', p'' \in \{p, \overline{p}\}$, and the rule $q(e) \to e$. Clearly, $[\![q]\!]_M$ is the identity on $T_{\{b,e\}}$, and $[\![M]\!] = [\![q_0]\!]_M$ is the required translation. □

3.2. Dead Ends

In this subsection we prove a simple normal form for ET^R transducers, to be used in the proofs of Lemmas 25 and 34. We will need the following elementary fact, that allows us to add as much regular look-ahead information to an ET^R transducer as we wish. It is an obvious extension of Lemma 1.

Lemma 3. Let $M = (Q, \Sigma, \Delta, I, R, A_1)$ be an ET^R transducer, and let $A_2, \dots, A_n$ be ftas over Σ, $n \geq 2$. There exist an ET^R transducer $M' = (Q, \Sigma, \Delta, I, R', A'_1)$ and ftas $A'_2, \dots, A'_n$ over Σ such that

(1) $A'_1, A'_2, \dots, A'_n$ have the same set of states and the same transition function,
(2) $L(A'_i) = L(A_i)$ for every $i \in [n]$,
(3) $[\![q]\!]_{M'} = [\![q]\!]_M$ for every $q \in Q$, and $[\![M']\!] = [\![M]\!]$, and
(4) if M is an ELT^R transducer, a T^R transducer, or a $\text{T}^\text{R}_\text{fc}$ transducer, then so is M'.

Proof. Let $A_i = (P_i, \Sigma, F_i, \delta_i)$ for every $i \in [n]$. We define $A'_i = (P', \Sigma, F'_i, \delta')$ as in the proof of Lemma 1 (with $P' = P_1 \times \cdots \times P_n$).

Moreover, if $q(C) \to \zeta\ \langle p_1, \dots, p_k\rangle$ is a rule of M, then the set R' of rules of M' contains all rules $q(C) \to \zeta\ \langle p'_1, \dots, p'_k\rangle$ such that the first component of p'_i is p_i for every $i \in [k]$. Thus, M' ignores all components of a look-ahead state except the first.

It should be clear that M' and $A'_2, \dots, A'_n$ as defined above, satisfy the requirements. For the finite-copying property, note that $[\![M'_{(p_1,\dots,p_n)}]\!](s) = [\![M_{p_1}]\!](s)$ for all $p_i \in P_i$ and $s \in T_\Sigma(\{\Box\})$. □

An ET^R transducer M is *without dead ends* if the following holds for every rule $q(C) \to \zeta\ \langle p_1, \dots, p_k\rangle$, every $\overline{q}(x_i)$ that occurs in ζ (with $\overline{q} \in Q$ and $i \in [k]$), and every $s \in T_\Sigma$: if $\delta(s) = p_i$, then $s \in \mathsf{dom}([\![\overline{q}]\!]_M)$. Intuitively this means that M does not have unsuccessful computations.

Lemma 4. For every ET^R transducer M there is an equivalent ET^R transducer M' without dead ends. If M is an ELT^R, T^R, or $\text{T}^\text{R}_\text{fc}$ transducer, then so is M'. Moreover, if M has exactly one initial state, then so does M'.

Proof. Let $M = (Q, \Sigma, \Delta, I, R, A)$ with $A = (P, \Sigma, F, \delta)$. Since $\mathsf{dom}([\![\overline{q}]\!]_M)$ is a regular tree language for every $\overline{q} \in Q$ (because $\text{ET}^\text{R} = \text{T}^\text{R}$ by Lemma 11 and the domain of a T^R translation is regular by [22, Corollary 2.7]), it is easy to transform M into M' by adding appropriate look-ahead information, as follows.

For every $\overline{q} \in Q$, let $A_{\overline{q}} = (P, \Sigma, F_{\overline{q}}, \delta)$ be an fta such that $L(A_{\overline{q}}) = \mathsf{dom}([\![\overline{q}]\!]_M)$. Note that by Lemma 3 we can assume that these automata and the automaton A have the same set of states and the same transition function. We let M' be the ET^R transducer $(Q, \Sigma, \Delta, I, R', A)$, where R' is the set of rules $q(C) \to \zeta\ \langle p_1, \ldots, p_k \rangle$ in R such that if $\overline{q}(x_i)$ occurs in ζ then $p_i \in F_{\overline{q}}$.

It should be clear that $[\![M']\!] = [\![M]\!]$. Obviously, if M is an ELT^R, T^R, or $\text{T}^\text{R}_\text{fc}$ transducer, then so is M'. For the finite-copying property, note that $[\![M'_p]\!](s) \subseteq [\![M_p]\!](s)$ for every $p \in P$ and $s \in T_\Sigma(\{\Box\})$. □

Example 5. Let $\Sigma = \Delta = \{a^{(1)}, b^{(0)}, c^{(0)}\}$. Let $A = (P, \Sigma, F, \delta)$ be the fta with $P = \{p_\text{e}, p_\text{o}\}$ and $F = \{p_\text{e}\}$ such that, for $s \in T_\Sigma$, $\delta(s) = p_\text{e}$ if and only if the number of a's in s is even. We define an LT^R transducer $M = (Q, \Sigma, \Delta, I, R, A)$ such that

$$[\![M]\!] = \{(a^{2n}b, a^{2n}b) \mid n \geq 0\} \cup \{(a^{2n}c, c) \mid n \geq 0\}.$$

Let $Q = I = \{q_b, q_c\}$, and let R contain the rules $q_b(b) \to b$ and $q_c(c) \to c$, and for every $p \in \{p_\text{e}, p_\text{o}\}$ the rules $q_b(a(x_1)) \to a(q_b(x_1))\ \langle p \rangle$ and $q_c(a(x_1)) \to q_c(x_1)\ \langle p \rangle$. Clearly $\mathsf{dom}([\![q_b]\!]_M) = T_{\{a,b\}}$ and $\mathsf{dom}([\![q_c]\!]_M) = T_{\{a,c\}}$. So, M is *not* without dead ends; for instance, the computation of M with input tree $a(a(c))$ that starts in initial state q_b is unsuccessful.

Now let B be the fta with two states p_b and p_c, both final, such that $\delta_B(a^n b) = p_b$ and $\delta_B(a^n c) = p_c$ for every $n \geq 0$ (where δ_B is the transition function of B), and let A' be the usual product of A and B (see the proof of Lemma 1). Moreover, let $M' = (Q, \Sigma, \Delta, I, R', A')$ be the LT^R transducer with the rules $q_b(b) \to b$ and $q_c(c) \to c$, and for every $p \in \{p_\text{e}, p_\text{o}\}$ the rules $q_b(a(x_1)) \to a(q_b(x_1))\ \langle (p, p_b) \rangle$ and $q_c(a(x_1)) \to q_c(x_1)\ \langle (p, p_c) \rangle$. It is easy to see that M' is equivalent to M and that M' is without dead ends. □

3.3. Properties

One of the two classes of queries that we are interested in, is DT^R. We will need the following two basic results. The first is from [22, Theorem 2.11], the second is proved in [15] (see also [16]).

Proposition 6. DT^R is closed under composition.

Proposition 7. It is decidable for two DT^R transducers whether they are equivalent.

As we will show in Section 5, determinacy is undecidable if the view transducers copy. Therefore, when we prove determinacy results in Section 7, we define views using linear transducers. To be precise,

the class of views that we are interested in, is fu-$(\text{ELT}^\text{R})^*$,

i.e., the class of all partial functions that are compositions of translations in ELT^{R}. This class properly contains fu-ELT^{R} by the proof of Theorem 5.2 of [12] (see also the proof of Theorem 34 of [14]). It will follow from Proposition 6 and Corollary 33 (in Section 6.3) that the class of views fu-$(\text{ELT}^{\text{R}})^*$ is properly contained in the class of queries DT^{R}.

From [11, Theorem 17] we recall a useful characterization of ELT^{R} as a class of so-called bimorphisms. We note that in [11] the class ELT^{R} is denoted l-$\text{XTOP}^{\text{R}}_{\text{ef}}$, where "ef" stands for "epsilon-free".

Proposition 8. ELT^{R} is the class of all tree translations $\{(f(r), g(r)) \mid r \in R\}$ where f is a linear nondeleting nonerasing tree homomorphism, g is a linear tree homomorphism, and R is a regular tree language.

From this proposition we obtain some basic properties of ELT^{R}. They are immediate from the fact that the class of regular tree languages is closed under inverse tree homomorphisms, intersection, and linear tree homomorphisms (noting that $\{(f(r), g(r)) \mid r \in R\} = f^{-1} \circ \mathsf{id}_R \circ g$).

Corollary 9. If $\tau \in \text{ELT}^{\text{R}}$ and R is a regular tree language, then $\tau(R)$ and $\tau^{-1}(R)$ are regular tree languages. In particular, $\mathsf{dom}(\tau)$ and $\mathsf{ran}(\tau)$ are regular tree languages.

For the next corollary we observe that if $\tau = \{(f(r), g(r)) \mid r \in R_\tau\}$, then its domain restriction to the set R is

$$\mathsf{id}_R \circ \tau = \{(f(r), g(r)) \mid r \in R_\tau \cap f^{-1}(R)\},$$

and its range restriction to R is

$$\tau \circ \mathsf{id}_R = \{(f(r), g(r)) \mid r \in R_\tau \cap g^{-1}(R)\}.$$

Corollary 10. If $\tau \in \text{ELT}^{\text{R}}$ and R is a regular tree language, then $\mathsf{id}_R \circ \tau$ and $\tau \circ \mathsf{id}_R$ are in ELT^{R}.

It is shown in [12, Theorem 4.8] that $\text{ET}^{\text{R}} = \text{T}^{\text{R}}$. The importance of extended top-down tree transducers is that in the linear case they are more powerful than ordinary top-down tree transducers, i.e., $\text{LT}^{\text{R}} \subsetneq \text{ELT}^{\text{R}}$, see Lemmas 4.6 and 4.7 of [12] (and Example 2). Here we show, using the construction in the proof of [12, Theorem 4.8] that ELT^{R} is included in $\text{T}^{\text{R}}_{\text{fc}}$ (as already observed before [11, Theorem 19]). The inclusion is proper by Corollary 9 because $\text{T}^{\text{R}}_{\text{fc}}$ contains translations that do not preserve regularity.

Lemma 11. $\text{ET}^{\text{R}} = \text{T}^{\text{R}}$ and $\text{ELT}^{\text{R}} \subsetneq \text{T}^{\text{R}}_{\text{fc}}$.

Proof. We use the construction in the proof of [12, Theorem 4.8], in our terminology, for an ET^{R} transducer $M = (Q, \Sigma, \Delta, I, R, A)$ with $A = (P, \Sigma, F, \delta)$. Let $C \subseteq T_\Sigma(X)$ be the finite set of contexts in the left-hand sides of the rules in R, and

let $\mathcal{V}$ be the set of all nodes of subtrees of those contexts, i.e., the union of all $V(C/v)$ with $C \in \mathcal{C}$ and $v \in V(C)$. For an X_k-context $C \in \mathcal{C}$ and $p_1, \ldots, p_k \in P$, let $A_{C,p_1,\ldots,p_k}$ be an fta over Σ such that

$$L(A_{C,p_1,\ldots,p_k}) = \{C[s_1, \ldots, s_k] \mid s_i \in T_\Sigma, \delta(s_i) = p_i \text{ for } i \in [k]\}.$$

Since $\mathcal{C}$ and P are finite, Lemma 1 shows that there exist ftas $A' = (P', \Sigma, F', \delta')$ and $A'_{C,p_1,\ldots,p_k} = (P', \Sigma, F_{C,p_1,\ldots,p_k}, \delta')$ such that $L(A') = L(A)$ and $L(A'_{C,p_1,\ldots,p_k}) = L(A_{C,p_1,\ldots,p_k})$. Note that these automata have the same set of states P' and the same transition function δ'.

We now construct the T^{R} transducer $M' = (Q \times \mathcal{V}, \Sigma, \Delta, I \times \{\varepsilon\}, R', A')$ with the following rules.

(i) If $q(C) \to \zeta\,\langle p_1, \ldots, p_k\rangle$ is a rule in R and $C = a(C_1, \ldots, C_m)$ with $a \in \Sigma^{(m)}$, $m \geq 0$, and $C_1, \ldots, C_m \in T_\Sigma(X_k)$, then R' contains all rules

$$\langle q, \varepsilon\rangle(a(x_1, \ldots, x_m)) \to \zeta'\,\langle p'_1, \ldots, p'_m\rangle$$

such that $\delta'(a, p'_1, \ldots, p'_m) \in F_{C,p_1,\ldots,p_k}$ and

$$\zeta' = \zeta[q'(x_i) \leftarrow \langle q', v\rangle(x_j) \mid q' \in Q, i \in [k], j \in [m], v \in V(C_j), \mathsf{lab}(C_j, v) = x_i].$$

Note that since C is an X_k-context, j and v are uniquely determined by i.

(ii) If $a \in \Sigma^{(m)}$, $m \geq 0$, and $\langle q, iv\rangle \in Q \times \mathcal{V}$ with $i \in [m]$ and $v \in \mathbb{N}_+^*$, then R' contains all rules

$$\langle q, iv\rangle(a(x_1, \ldots, x_m)) \to \langle q, v\rangle(x_i)\,\langle p'_1, \ldots, p'_m\rangle$$

with $p'_1, \ldots, p'_m \in P'$. Thus, the look-ahead is irrelevant for these rules.

It should be intuitively clear that M' is equivalent to M. In fact, it is straightforward to show by induction on the structure of $s \in T_\Sigma$ that $[\![\langle q, v\rangle]\!]_{M'}(s) = [\![q]\!]_M(s/v)$ if $v \in V(s)$ and $[\![\langle q, v\rangle]\!]_{M'}(s) = \emptyset$ otherwise. Hence $[\![\langle q, \varepsilon\rangle]\!]_{M'}(s) = [\![q]\!]_M(s)$ for every $q \in I$; since $L(A') = L(A)$, this shows that $[\![M']\!] = [\![M]\!]$.

It remains to prove that M' is finite-copying if M is linear. Let $K \in \mathbb{N}_+$ be such that if $q(C) \to \zeta\,\langle p_1, \ldots, p_k\rangle$ is a rule of M, then $k \leq K$. Consider $\langle q, v\rangle \in Q \times \mathcal{V}$, $p \in P'$, $s \in T_\Sigma(\{\Box\})$ with one occurrence of $\Box$, and $t \in [\![\langle q, v\rangle]\!]_{M'_p}(s)$. We claim that the number of occurrences of $\Box$ in t is

(1)	$\leq K$	if $v \in V(s)$ and $\Box$ occurs in s/v
(2)	$= 0$	if $v \in V(s)$ and $\Box$ does not occur in s/v, and
(3)	$= 1$	if $v \notin V(s)$ and v has a prefix w such that $\mathsf{lab}(s, w) = \Box$.

Note that if $v \notin V(s)$ and v has no prefix w with $\mathsf{lab}(s, w) = \Box$, then $[\![\langle q, v\rangle]\!]_{M'_p}(s) = \emptyset$ (by the rules of M' of type (ii)). Note also that (again by the rules of M' of type (ii)) if $v \in V(s)$, then $t \in [\![\langle q, \varepsilon\rangle]\!]_{M'_p}(s/v)$. That proves Claim (2). We prove Claims (1) and (3) by induction on the structure of s. If $s = \Box$ then $t = \Box$, and so the claims are true. Now let $s = a(s_1, \ldots, s_m)$ with $a \in \Sigma^{(m)}$, $m \geq 0$.

We first consider the case that $v \neq \varepsilon$, i.e., $v = iv'$ with $i \in [m]$ and $v' \in \mathbb{N}_+^*$. By a rule of type (ii), $t \in [\![\langle q, v'\rangle]\!]_{M'_p}(s_i)$. Clearly, if v and s satisfy the conditions of Claim (1), then so do v' and s_i, and hence Claim (1) is true by induction. The same holds for Claim (3).

It remains to consider the case that $v = \varepsilon$, for which we only have to prove Claim (1). Let $\langle q, \varepsilon\rangle(a(x_1, \ldots, x_m)) \to \zeta' \langle p'_1, \ldots, p'_m\rangle$ be the rule of M' of type (i) that is applied to s to obtain t, constructed from a rule $q(C) \to \zeta \langle p_1, \ldots, p_k\rangle$ of M. Let $\langle q'_1, v_1\rangle(x_{j_1}), \ldots, \langle q'_n, v_n\rangle(x_{j_n})$ be all occurrences of subtrees of ζ' in $(Q \times \mathcal{V})(X_m)$. Note that since M is linear, $n \leq k \leq K$. We also observe that all sequences $j_r v_r$, $r \in [n]$, are different and, in fact, no such sequence is a prefix of another one, because they are different leaves of C (with label in X_k). So, there are $t_1, \ldots, t_n$ such that $t = \zeta'[\langle q'_r, v_r\rangle(x_{j_r}) \leftarrow t_r \mid r \in [n]]$ and $t_r \in [\![\langle q'_r, v_r\rangle]\!]_{M'_p}(s_{j_r})$ for every $r \in [n]$. We now consider two subcases.

Case 1: There is no $r \in [n]$ such that $v_r \in V(s_{j_r})$ and $\Box$ occurs in s_{j_r}/v_r. Then, by induction, the number of occurrences of $\Box$ in t_r is ≤ 1 by Claims (2) and (3), and so the number of occurrences of $\Box$ in t is $\leq n \leq K$.

Case 2: There exists $r_0 \in [n]$ such that $v_{r_0} \in V(s_{j_{r_0}})$ and $\Box$ occurs in $s_{j_{r_0}}/v_{r_0}$. Then $\Box$ occurs in $s/j_{r_0}v_{r_0}$ and so we obtain by the above observation, for every $r \neq r_0$, that $\Box$ does *not* occur in s_{j_r}/v_r (if $v_r \in V(s_{j_r})$) and v_r has *no* prefix w such that $\mathsf{lab}(s_{j_r}, w) = \Box$ (if $v_r \notin V(s_{j_r})$); hence, by induction, $\Box$ does not occur in t_r by Claim (2). By induction and Claim (1), t_{r_0} has at most K occurrences of $\Box$, and hence so has t. $\Box$

Example 12. Consider the ELT$^{\mathrm{R}}$ transducer $M = (Q, \Sigma, \Delta, I, R, A)$ of Example 2. Let B be the fta with states p_a, p_b, p_e, all final, such that, for all $s_1, s_2, s_3 \in T_\Sigma$, $\delta_B(a(s_1, s_2, s_3)) = p_a$, $\delta_B(b(s_1, s_2)) = p_b$, and $\delta_B(e) = p_e$. Let A' be the usual product of A and B, with set of states P'. Then an equivalent $\mathrm{T}^{\mathrm{R}}_{\mathrm{fc}}$ transducer $M' = (Q', \Sigma, \Delta, I', R', A')$ has set of states $Q' = \{q_0, q, q_1, q_2\}$, with $I' = I = \{q_0\}$, where q_0 and q stand for $\langle q_0, \varepsilon\rangle$ and $\langle q, \varepsilon\rangle$, and q_i stands for $\langle q, i\rangle$. Moreover, M' has the rule

$$q_0(a(x_1, x_2, x_3)) \to a(q_1(x_1), q_2(x_1), q_1(x_2)) \langle (p, p_b), (p, p_b), (p, p_e)\rangle,$$

for all $p', p'' \in P'$ the rules

$$\begin{array}{lcl} q_1(b(x_1, x_2)) & \to & q(x_1) \langle p', p''\rangle, \\ q_2(b(x_1, x_2)) & \to & q(x_2) \langle p', p''\rangle, \\ q(b(x_1, x_2)) & \to & b(q(x_1), q(x_2)) \langle p', p''\rangle, \end{array}$$

and the rule $q(e) \to e$. $\Box$

Since our views are taken from the class fu-(ELT$^{\mathrm{R}}$)*, the next proposition is relevant.

Proposition 13. For an ELT$^{\mathrm{R}}$ transducer M it is decidable whether $[\![M]\!]$ is functional.

PROOF. By Lemma 11, $\text{ET}^{\text{R}} = \text{T}^{\text{R}}$. The result follows because functionality is decidable for T^{R} transducers by [15] (see the sentence after Theorem 8 of [15]). □

However, we do not know whether functionality is decidable for the composition of the translations realized by a given sequence of ELT^{R} transducers. On the other hand, we will show in Corollary 36 (in Section 6.3) that our class fu-$(\text{ELT}^{\text{R}})^*$ is in fact equal to the class (fu-$\text{ELT}^{\text{R}})^*$ of compositions of functional ELT^{R} translations. Thus, sequences of functional ELT^{R} transducers form an effective representation of the class of views fu-$(\text{ELT}^{\text{R}})^*$. We observe also that three such transducers suffice: it is shown in [14] that $(\text{ELT}^{\text{R}})^* = (\text{ELT}^{\text{R}})^3$.

We finally note that, by [14, Lemma 15], $(\text{ELT}^{\text{R}})^* = \text{ELT}^*$ and hence our class fu-$(\text{ELT}^{\text{R}})^*$ of views equals fu-ELT^*.

4. MSO Definable Tree Translations

A deterministic MSO graph transducer is a device that transforms graphs by using formulas from monadic second-order logic to define the output graph in terms of a finite number of copies of the input graph. For a formal definition see, e.g., [23, Chapter 7] where it is called a parameterless monadic second-order definition scheme. We say that it is a *deterministic* MSO *tree transducer* (DMSOT transducer) if it transforms trees into trees, see, e.g., [23, Chapter 8]. We note that the translation realized by a DMSOT transducer is functional. The class DMSOT of these translations is the second of the two classes of queries that we are interested in. It is investigated, e.g., in [24, 25, 26, 27].[1] It is incomparable with the class of queries DT^{R}.

We will only need the following three results on DMSOT. The first result is easy to prove, but also follows from [25, Theorem 7.1]. The second result is well known, see, e.g., [23, Theorem 7.14]. The third result is from [17, Corollary 10].

Proposition 14. $\text{DT}^{\text{R}}_{\text{fc}} \subseteq \text{DMSOT}$.

It is proved in [26, Theorem 7.4] that, in fact, $\text{DT}^{\text{R}}_{\text{fc}}$ is the intersection of our two classes of queries, i.e., $\text{DT}^{\text{R}}_{\text{fc}} = \text{DT}^{\text{R}} \cap \text{DMSOT}$.

Proposition 15. DMSOT is closed under composition.

It will follow from Propositions 14, 15 and Corollary 33 (in Section 6.3) that our class of views fu-$(\text{ELT}^{\text{R}})^*$ is properly contained in the class of queries DMSOT.

Proposition 16. It is decidable for two DMSOT transducers whether they are equivalent.

[1] In [24, 25, 26] the class contains total functions only, and is denoted MSO-TT or MSOTT.

5. Undecidability Results

The simplest kind of query is the identity on the set of inputs. Let ID denote the class of translations that are the identity on T_Σ, for any ranked alphabet Σ. As observed in [6], a function v is injective if and only if q is determined by v, where q is the identity on $\mathsf{dom}(v)$. Since the injectivity problem for tree homomorphisms is undecidable by [9], one obtains (as stated in [6, Theorem 17]) undecidability of the determinacy problem for ID under HOM, where HOM $\subseteq$ DT is the class of tree homomorphisms.

The tree homomorphisms used in the proof of [9] are not linear, and hence they are not even finite-copying. Here we show that determinacy is already undecidable for identity queries if the view makes at most one copy of the input tree. To be precise, we show that determinacy is undecidable for ID under views that are realized by what we will here call "total copy-once" DT transducers (t-DT$_{\text{co}}$ transducers). A DT transducer M is *total* if for each state $q \in Q$ and input symbol $a \in \Sigma^{(k)}$, $k \geq 0$, it has a rule with left-hand side $q(a(x_1, \ldots, x_k))$. For such a transducer, $\mathsf{dom}([\![M]\!]) = T_\Sigma$. It is *copy-once* if for every rule $q(a(x_1, \ldots, x_k)) \to \zeta$,

- the initial state q_0 of M does not occur in ζ,
- if $q = q_0$ then each variable x_i occurs at most twice in ζ, and
- if $q \neq q_0$ then ζ is linear.

Thus the transducer copies at most once, at the root of the input tree. The undecidability of injectivity for nontotal DT$_{\text{co}}$ transducers was proved by Ésik in [8], and in his PhD thesis (in Hungarian). Our proof for total DT$_{\text{co}}$ transducers is a slight variation of Ésik's proof.

Theorem 17. Determinacy for ID under t-DT$_{\text{co}}$ is undecidable.

Proof. We use an encoding of the Modified Post Correspondence Problem (MPCP). An instance of this problem consists of two sequences of strings $(\alpha_1, \ldots, \alpha_n)$ and $(\beta_1, \ldots, \beta_n)$, $n \geq 1$, and it has a solution if there exist numbers $i_1, \ldots, i_k \in [n]$, $k \geq 0$, such that the strings $\alpha_1\alpha_{i_1} \cdots \alpha_{i_k}$ and $\beta_1\beta_{i_1} \cdots \beta_{i_k}$ are equal. It is well known to be undecidable whether an MPCP instance has a solution. Let Ω be the ranked alphabet consisting of all symbols that appear in the strings α_i and β_i, where each symbol has rank one. For a string $\alpha = a_1 \cdots a_m$ and a tree t, we write $\alpha(t)$ to mean the tree $a_1(a_2(\cdots a_m(t)\cdots))$. We define the t-DT$_{\text{co}}$ transducer $M = (Q, \Sigma, \Delta, \{q_0\}, R)$ where $Q = \{q_0, q_a, q_b, q_{\text{id}}\}$, $\Sigma = \{i^{(1)} \mid i \in [n]\} \cup \{a^{(1)}, b^{(1)}, e^{(0)}\}$, and $\Delta = \Sigma \cup \Omega \cup \{f^{(2)}\}$. The state q_{id} realizes the identity on T_Σ; for every $\sigma \in \Sigma^{(k)}$, $k \geq 0$, it has the rule $q_{\text{id}}(\sigma(x_1, \ldots, x_k)) \to \sigma(q_{\text{id}}(x_1), \ldots, q_{\text{id}}(x_k))$ in R. For $i \in [n]$ we define the following rules in R.

$$
\begin{array}{lcl@{\qquad}lcl}
q_0(i(x_1)) & \to & i(q_{\text{id}}(x_1)) & q_a(i(x_1)) & \to & \alpha_i(q_a(x_1)) \\
q_0(e) & \to & e & q_b(i(x_1)) & \to & \beta_i(q_b(x_1)) \\
q_0(a(x_1)) & \to & f(q_{\text{id}}(x_1), \alpha_1(q_a(x_1))) & q_u(v(x_1)) & \to & e \quad \text{for all } u, v \in \{a, b\} \\
q_0(b(x_1)) & \to & f(q_{\text{id}}(x_1), \beta_1(q_b(x_1))) & q_u(e) & \to & e \quad \text{for all } u \in \{a, b\}
\end{array}
$$

The transducer M translates every input tree of which the root is not labeled by a or by b into itself. It translates every input tree $ai_1 \cdots i_k(s)$ where the root of $s \in T_\Sigma$ is not a number, into $f(i_1 \cdots i_k(s), \alpha_1\alpha_{i_1} \cdots \alpha_{i_k}(e))$, and similarly for b and β_{i_j}. This implies that the given MPCP instance has no solution if and only if $[\![M]\!]$ is injective if and only if the identity id_{T_Σ} is determined by $[\![M]\!]$, and hence it is undecidable whether id_{T_Σ} is determined by $[\![M]\!]$. □

Since, obviously, every $\mathrm{DT}_{\mathrm{co}}$ transducer is a $\mathrm{DT}_{\mathrm{fc}}$ transducer, and $\mathrm{DT}_{\mathrm{fc}}$ is (effectively) included in DMSOT by Proposition 14, Theorem 17 immediately gives undecidability of determinacy for ID under DMSOT, which slightly strengthens Theorem 19 of [6] (where the identities on all regular tree languages are allowed as query).

One often considers that a query q is determined by a *finite tuple of views* $\mathcal{F} = (v_1, \ldots, v_n)$. The extended definition states that q is determined by $\mathcal{F}$ if it is determined by the single view $\bar{v}$, where $\bar{v}(s) = (v_1(s), \ldots, v_n(s))$ for every input s. In this case one has undecidability even when the views are total deterministic finite-state string transformations. In fact, for a pair of views $\mathcal{F} = (v_1, v_2)$ consisting of two total deterministic finite-state string transducers v_1 and v_2, the proof is the same as for Theorem 17, but producing the ith branch of the output tree by the view v_i. Thus, in what follows we consider only a single noncopying view.

6. Inverses and Uniformizers of Linear Transducers

As Theorem 17 shows, determinacy cannot be decided under view transducers that copy, not even for a single initial copy at the root of the input tree. We therefore restrict our attention to linear view transducers. In the first two subsections of this section we consider arbitrary linear transducers. In the third subsection, and in Section 7, we restrict attention to functional linear translations.

Let τ and u be translations (i.e., binary relations). We say that u is a *uniformizer of* τ if $u \subseteq \tau$ and $\mathsf{dom}(u) = \mathsf{dom}(\tau)$. Note that, for technical reasons, in this definition we do not require u to be a function (as we did in the Introduction). We will, however, be mainly interested in functional uniformizers. The notion of uniformizer is relevant for deciding whether a query q is determined by a view v.

Lemma 18. Let q, v, u be partial functions such that u is a uniformizer of v^{-1}. Then, q is determined by v if and only if $q = v \circ u \circ q$.

Proof. If $q = v \circ u \circ q$, then $q = v \circ f$ where f is the partial function $u \circ q$. So, q is determined by v. Vice versa, if $q = v \circ f$ for a partial function f, then $v \circ u \circ q = v \circ u \circ v \circ f = v \circ f = q$ because $u \circ v$ is the identity on $\mathsf{ran}(v)$.[2] □

[2]We know that $u \subseteq v^{-1}$ and $\mathsf{dom}(u) = \mathsf{dom}(v^{-1})$. Since $\mathsf{ran}(u) \subseteq \mathsf{ran}(v^{-1}) = \mathsf{dom}(v)$, we have that $\mathsf{dom}(u \circ v) = \mathsf{dom}(u) = \mathsf{dom}(v^{-1}) = \mathsf{ran}(v)$. Also, if $(x, y) \in u \circ v$, then $(x, y) \in v^{-1} \circ v$ and so $x = y$ because v is functional. Hence $u \circ v$ is the identity on $\mathsf{ran}(v)$.

Example 19. Consider the view $v = [\![M]\!]$ where M is the ELT$^{\mathrm{R}}$ transducer of Example 2, i.e.,

$$v = \{(a(b(s_1, s_2), b(s_3, s_4), e)\,,\ a(s_1, s_2, s_3)) \mid s_1, s_2, s_3, s_4 \in T_{\{b,e\}}\}.$$

Let q be the query

$$q = \{(a(b(s_1, s_2), b(s_3, s_4), e)\,,\ b(s_1, s_1)) \mid s_1, s_2, s_3, s_4 \in T_{\{b,e\}}\}.$$

Using the construction in the proof of Lemma 11 it is easy to see that $q \in \mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}}$ and hence q is in both DT$^{\mathrm{R}}$ and DMSOT. Clearly, q is determined by v; for example, $q = v \circ f$ where

$$f = \{(a(s_1, s_2, s_3)\,,\ b(s_1, s_1)) \mid s_1, s_2, s_3 \in T_\Sigma\}.$$

A functional uniformizer of v^{-1} is

$$u = \{(a(s_1, s_2, s_3)\,,\ a(b(s_1, s_2), b(s_3, e), e)) \mid s_1, s_2, s_3 \in T_{\{b,e\}}\}.$$

Note that $u \in$ LT, as can easily be verified. Note also that $u \circ v = \mathsf{id}_{\mathsf{ran}(v)}$ where $\mathsf{ran}(v) = \{a(s_1, s_2, s_3) \mid s_1, s_2, s_3 \in T_{\{b,e\}}\}$. Obviously,

$$u \circ q = \{(a(s_1, s_2, s_3)\,,\ b(s_1, s_1)) \mid s_1, s_2, s_3 \in T_{\{b,e\}}\}$$

and $q = v \circ (u \circ q)$. □

6.1. Inverses

Because of Lemma 18, we are interested in uniformizers of inverse views. Thus, given an ELT$^{\mathrm{R}}$ transducer M, we would like to construct a transducer realizing its inverse $[\![M]\!]^{-1}$. Since M can translate the set of all input trees in T_Σ to a single output tree, a transducer realizing $[\![M]\!]^{-1}$ may need to translate a tree back to any tree in T_Σ. This is not possible by our extended top-down tree transducers because the height of an output tree is linearly bounded by the height of the input tree. The next, easy lemma "factors out" this problem by decomposing an ELT$^{\mathrm{R}}$ transducer into a component that can be inverted as an ELT transducer, and a component of a very simple form: a "projection".

Let n-ELT denote the class of *nondeleting nonerasing* ELT transducers: those in which every rule is of the form $q(C) \to \zeta$ such that each variable in X_k occurs in ζ and $\zeta \notin Q(\{x_1\})$. The phrase "nondeleting" indicates that we do not drop any input x_i, thus "deleting" an entire subtree from the input. The phrase "nonerasing" indicates that we do not have a rule such as $q(a(x_1, b)) \to q'(x_1)$, which "erases" the symbols a and b.

Let Δ be a ranked alphabet and let H be a ranked alphabet disjoint from Δ with $H^{(0)} = \emptyset$, i.e., each symbol of H has rank at least 1. The *projection from* $\Delta \cup H$ *to* Δ is the tree homomorphism $\pi_{\Delta,H} = \pi : T_{\Delta\cup H} \to T_\Delta$ such that $\pi_h = x_1$ for every

$h \in H$ and $\pi_d = d(x_1, \ldots, x_k)$ for every $d \in \Delta^{(k)}$.[3] We denote by PROJ the class of all projections.

Lemma 20. $\text{ELT}^{\text{R}} \subseteq \text{n-ELT} \circ \text{PROJ}$.

PROOF. It is proved in [10] (where n-ELT is denoted TID) that Proposition 8 holds for n-ELT when g is also required to be nondeleting and nonerasing. Thus, it suffices to prove that every linear tree homomorphism g can be decomposed into $g' \circ \pi$, where g' is a linear nondeleting nonerasing tree homomorphism and $\pi \in \text{PROJ}$. As shown in Lemma I-2-1-3-5 of [28] we can take the ranked alphabet $H = \{\#_n^{(n)} \mid 1 \leq n \leq m+1\}$, where m is the maximal rank of the symbols in Σ, and we can define $g'_a = \#_{n+1}(g_a, x_{i_1}, \ldots, x_{i_n})$ for every $a \in \Sigma^{(k)}$, $k \geq 0$, where $x_{i_1}, \ldots, x_{i_n}$ are the variables from X_k that do not occur in g_a. It should be clear that $g = g' \circ \pi_{\Delta,H}$. □

It is easy to see that n-ELT is closed under inverse. In fact, as shown in [10], one just "inverts" the rules of the given transducer.

Lemma 21. $\text{n-ELT}^{-1} \subseteq \text{n-ELT}$.

PROOF. A rule of an n-ELT transducer M is of the form

$$q(C) \to C'[q_1(x_1), \ldots, q_k(x_k)]$$

where C and C' are X_k-contexts $\neq x_1$ (and $q, q_1, \ldots, q_k \in Q$). An n-ELT transducer realizing $[\![M]\!]^{-1}$ is obtained from M by changing each such rule into the rule $q(C') \to C[q_1(x_1), \ldots, q_k(x_k)]$. □

The previous two lemmas give us the next corollary.

Corollary 22. $(\text{ELT}^{\text{R}})^{-1} \subseteq \text{PROJ}^{-1} \circ \text{ELT}$.

Example 23. Let M be the ELT^{R} transducer of Example 2. Then $[\![M]\!] = [\![M']\!] \circ \pi_{\Delta,H}$ where $H = \{\#_2^{(2)}\}$ and M' is the ELT transducer with the rule

$$q_0(a(b(x_1, x_2), b(x_3, x_4), e)) \to \#_2(a(q(x_1), q(x_2), q(x_3)), q(x_4))$$

and the rules $q(b(x_1, x_2)) \to b(q(x_1), q(x_2))$ and $q(e) \to e$. Then we obtain that $[\![M]\!]^{-1} = \pi^{-1}_{\Delta,H} \circ [\![M'']\!]$ where M'' is the ELT transducer with the rule

$$q_0(\#_2(a(x_1, x_2, x_3), x_4)) \to a(b(q(x_1), q(x_2)), b(q(x_3), q(x_4)), e)$$

and the same rules $q(b(x_1, x_2)) \to b(q(x_1), q(x_2))$ and $q(e) \to e$. □

[3] Thus, $\pi(h(s_1, \ldots, s_k)) = \pi(s_1)$ for $h \in H^{(k)}$ and $\pi(d(s_1, \ldots, s_k)) = d(\pi(s_1), \ldots, \pi(s_k))$ for $d \in \Delta^{(k)}$, for all $k \geq 0$ and $s_1, \ldots, s_k \in T_{\Delta \cup H}$.

6.2. Uniformizers

We say that the sequence $\tau_1, \ldots, \tau_n$ of translations is *compatible* if $\mathsf{ran}(\tau_i) \subseteq \mathsf{dom}(\tau_{i+1})$ for every $i \in [n-1]$.

Lemma 24. If $u_1, \ldots, u_n$ are uniformizers of $\tau_1, \ldots, \tau_n$, respectively, and the sequence $\tau_1, \ldots, \tau_n$ is compatible, then $u_1 \circ \cdots \circ u_n$ is a uniformizer of $\tau_1 \circ \cdots \circ \tau_n$.

Proof. Obviously $u_1 \circ \cdots \circ u_n \subseteq \tau_1 \circ \cdots \circ \tau_n$. Since the sequence $\tau_1, \ldots, \tau_n$ is compatible, $\mathsf{dom}(\tau_1 \circ \cdots \circ \tau_n) = \mathsf{dom}(\tau_1)$. Since $\mathsf{ran}(u_i) \subseteq \mathsf{ran}(\tau_i) \subseteq \mathsf{dom}(\tau_{i+1}) = \mathsf{dom}(u_{i+1})$, the sequence $u_1, \ldots, u_n$ is also compatible and so $\mathsf{dom}(u_1 \circ \cdots \circ u_n) = \mathsf{dom}(u_1)$. This shows that $\mathsf{dom}(u_1 \circ \cdots \circ u_n) = \mathsf{dom}(\tau_1 \circ \cdots \circ \tau_n)$. □

For classes $\mathcal{T}, \mathcal{U}$ of translations we say that $\mathcal{T}$ *has uniformizers in* $\mathcal{U}$ if for every $\tau \in \mathcal{T}$ we can construct a uniformizer u of τ such that $u \in \mathcal{U}$.

Our goal in this subsection is to show that $(\textsc{elt}^{\mathrm{R}})^*$ and $((\textsc{elt}^{\mathrm{R}})^*)^{-1}$ have uniformizers in $(\textsc{dt}^{\mathrm{R}}_{\mathrm{fc}})^*$. We do this by decomposing into compatible translations, constructing uniformizers in $\textsc{dt}^{\mathrm{R}}_{\mathrm{fc}}$ for them, and then obtaining a uniformizer in $(\textsc{dt}^{\mathrm{R}}_{\mathrm{fc}})^*$ by Lemma 24. A similar idea was used in [29] to obtain uniformizers in $\textsc{dt}^{\mathrm{R}}$ for compositions of top-down and bottom-up tree translations. Note that $\textsc{dt}^{\mathrm{R}}$ is closed under composition by Proposition 6; in fact, $\textsc{dt}^{\mathrm{R}}_{\mathrm{fc}}$ is also closed under composition (see, e.g., [30, Theorem 5.4]), but this fact will not be needed in this paper.

We start with a slight generalization of the Lemma in [29] (which says that $\textsc{t}$ has uniformizers in $\textsc{dt}^{\mathrm{R}}$).

Lemma 25. $\textsc{t}^{\mathrm{R}}$ has uniformizers in $\textsc{dt}^{\mathrm{R}}$, and $\textsc{t}^{\mathrm{R}}_{\mathrm{fc}}$ has uniformizers in $\textsc{dt}^{\mathrm{R}}_{\mathrm{fc}}$.

Proof. Let $M = (Q, \Sigma, \Delta, I, R, A)$ be a $\textsc{t}^{\mathrm{R}}$ transducer. Obviously, we may assume that I is a singleton (as required for determinism).[4] By Lemma 4 we may also assume that M is without dead ends. Suppose that M has two rules

$$q(a(x_1, \ldots, x_k)) \to \zeta_i \langle p_1, \ldots, p_k \rangle,$$

for $i = 1, 2$ with $\zeta_1 \neq \zeta_2$, and let M' be obtained from M by removing one of these rules. Since M is without dead ends, $[\![\overline{q}]\!]_{M'}$ is a uniformizer of $[\![\overline{q}]\!]_M$ for every $\overline{q} \in Q$.[5] Hence $[\![M']\!]$ is a uniformizer of $[\![M]\!]$, and M' is still without dead ends. Repeating this process, one finally obtains a $\textsc{dt}^{\mathrm{R}}$ transducer M'' such that $[\![M'']\!]$ is a uniformizer of $[\![M]\!]$. The construction obviously preserves the finite-copying property. □

The next result is immediate from Lemmas 11 and 25.

[4] Introduce a new state q_0 (which is the unique new initial state), and for every rule $q(C) \to \zeta \langle p_1, \ldots, p_k \rangle$ where q is an old initial state, add the rule $q_0(C) \to \zeta \langle p_1, \ldots, p_k \rangle$.

[5] Recall that, by definition, a uniformizer need not be functional.

Corollary 26. ELT^{R} has uniformizers in $\text{DT}^{\text{R}}_{\text{fc}}$.

Theorem 27. $(\text{ELT}^{\text{R}})^*$ has uniformizers in $(\text{DT}^{\text{R}}_{\text{fc}})^*$.

Proof. Let $\tau_1, \dots, \tau_n$ be ELT^{R} translations. We define ELT^{R} translations $\rho_1, \dots, \rho_n$ such that $\rho_1 \circ \cdots \circ \rho_n = \tau_1 \circ \cdots \circ \tau_n$ and the sequence $\rho_1, \dots, \rho_n$ is compatible. We let $\rho_n = \tau_n$, and for $i \in [n-1]$ we let ρ_i be the range restriction $\tau_i \circ \mathsf{id}_R$ of τ_i to $R = \mathsf{dom}(\rho_{i+1})$. The language R is regular by Corollary 9, and $\tau_i \circ \mathsf{id}_R$ is in ELT^{R} by Corollary 10. We obtain uniformizers in $\text{DT}^{\text{R}}_{\text{fc}}$ for the ρ_i by Corollary 26, and a uniformizer for $\rho_1 \circ \cdots \circ \rho_n$ in $(\text{DT}^{\text{R}}_{\text{fc}})^*$ by Lemma 24. □

We now turn to the inverses of the translations in $(\text{ELT}^{\text{R}})^*$. Since the relation "is a uniformizer of" is transitive, it suffices to show that $((\text{ELT}^{\text{R}})^*)^{-1}$ has uniformizers in $(\text{ELT}^{\text{R}})^*$.

An FTA *transducer* is an fta A, seen as a tree transducer realizing the translation $[\![A]\!] = \mathsf{id}_{L(A)}$, which is the identity on $L(A)$; composing a tree translation τ with $[\![A]\!]$ amounts to restricting the range of τ to $L(A)$: $\tau \circ [\![A]\!] = \{(s,t) \in \tau \mid t \in L(A)\}$.

Lemma 28. $\text{PROJ}^{-1} \circ \text{FTA}$ has uniformizers in LT.

Proof. Let $\tau = \pi^{-1} \circ [\![A]\!]$ where $\pi \in \text{PROJ}$ and A is an FTA transducer. Thus, $\pi = \pi_{\Sigma,H}$ for disjoint ranked alphabets Σ and H such that $H^{(0)} = \emptyset$, and A is an fta $(Q, \Sigma \cup H, F, \delta)$. Let C be the set of all X_1-contexts C over $\Sigma \cup H$ such that the left-most leaf of C has label x_1 and all the ancestors of this leaf have labels in H. Note that $\pi(C[t]) = \pi(t)$ for every $C \in C$ and $t \in T_{\Sigma \cup H}$. For every $a \in \Sigma^{(k)}$ and $q, q_1, \dots, q_k \in Q$, let $C(a, q, q_1, \dots, q_k)$ be the set of all $C \in C$ such that $\delta(C[a(x_1, \dots, x_k)], q_1, \dots, q_k) = q$.[6] Let $C_0(a, q, q_1, \dots, q_k)$ be one (fixed) such C – since the set $C(a, q, q_1, \dots, q_k)$ is effectively regular,[7] one can always compute such an element C if the set is nonempty. If there does not exist such a C then $C_0(a, q, q_1, \dots, q_k)$ is undefined.

We now construct an LT transducer M such that $[\![M]\!]$ is a uniformizer of τ. We define $M = (Q, \Sigma, \Sigma \cup H, F, R')$ where R' consists of all rules

$$q(a(x_1, \dots, x_k)) \to C_0(a, q, q_1, \dots, q_k)[a(q_1(x_1), \dots, q_k(x_k))]$$

such that $C_0(a, q, q_1, \dots, q_k)$ is defined. Intuitively, for $s \in T_\Sigma$, M simulates top-down the state behavior of A on some tree t in $\pi^{-1}(s)$ and, at each node of s, outputs a context in C on which A has the same state behavior as on the context in C that is "above" the corresponding node in t. Formally, the correctness of the construction follows from the following two claims, in which $q \in Q$, $s \in T_\Sigma$, and $t \in T_{\Sigma \cup H}$.

Claim 1. If $t \in [\![q]\!]_M(s)$, then $\pi(t) = s$ and $\delta(t) = q$.

[6] Recall again that the transition function δ was extended to contexts at the end of Section 2.

[7] It is the intersection of the regular tree language C with the tree language accepted by the fta $(Q, \Sigma \cup H \cup \{x_1\}, \{q\}, \delta')$ where δ' extends δ with $\delta'(x_1) = \delta(a, q_1, \dots, q_k)$.

The proof is by structural induction on s. If $s = a(s_1, \ldots, s_k)$, then, by the above rule, $t = C[a(t_1, \ldots, t_k)]$ where $C = C_0(a, q, q_1, \ldots, q_k)$ and $t_i \in [\![q_i]\!]_M(s_i)$ for every $i \in [k]$. By induction, $\pi(t_i) = s_i$ and $\delta(t_i) = q_i$. Then $\pi(t) = \pi(a(t_1, \ldots, t_k)) = a(\pi(t_1), \ldots, \pi(t_k)) = s$ and $\delta(t) = \delta(C[a(x_1, \ldots, x_k)], q_1, \ldots, q_k) = q$.

Claim 2. If $\delta(t) = q$, then $\pi(t) \in \mathsf{dom}([\![q]\!]_M)$.

The proof is by induction on the size of t. Clearly, t is of the form $C[a(t_1, \ldots, t_k)]$ for (unique) $C \in C$, $k \geq 0$, $a \in \Sigma^{(k)}$ and $t_1, \ldots, t_k \in T_{\Sigma \cup H}$. Let $q_i = \delta(t_i)$ and $s_i = \pi(t_i)$ for $i \in [k]$. By induction, $s_i \in \mathsf{dom}([\![q_i]\!]_M)$, i.e., there exists $t'_i \in [\![q_i]\!]_M(s_i)$. Let $s = a(s_1, \ldots, s_k)$, so $\pi(t) = s$. Then, by the above rule, $C_0(a, q, q_1, \ldots, q_k)[a(t'_1, \ldots, t'_k)]$ is in $[\![q]\!]_M(s)$, and so $s \in \mathsf{dom}([\![q]\!]_M)$. Note that $C_0(a, q, q_1, \ldots, q_k)$ is defined because $C \in C(a, q, q_1, \ldots, q_k)$: $\delta(C[a(x_1, \ldots, x_k)], q_1, \ldots, q_k) = \delta(t) = q$.

Taking $q \in F$, Claim 1 shows that $[\![M]\!] \subseteq \tau$, and Claim 2 shows that $\mathsf{dom}([\![M]\!])$ equals $\mathsf{dom}(\tau)$ (in fact, if $s \in \mathsf{dom}(\tau)$ then there exists t such that $s = \pi(t)$ and $\delta(t) = q \in F$, and so $s \in \mathsf{dom}([\![q]\!]_M) \subseteq \mathsf{dom}([\![M]\!])$). Hence $[\![M]\!]$ is a uniformizer of τ. □

Lemma 29. $(\text{ELT}^{\text{R}})^{-1}$ has uniformizers in ELT^2.

Proof. Let $\tau \in (\text{ELT}^{\text{R}})^{-1}$. By Corollary 22, $\tau = \tau_1 \circ \tau_2$ with $\tau_1 \in \text{PROJ}^{-1}$ and $\tau_2 \in \text{ELT}$. By Corollary 9 we can construct an fta A such that $L(A) = \mathsf{dom}(\tau_2)$. Then $\tau = \tau_1 \circ [\![A]\!] \circ \tau_2$ and the translations $\tau_1 \circ [\![A]\!], \tau_2$ are compatible (by definition of A). For $\tau_1 \circ [\![A]\!] \in \text{PROJ}^{-1} \circ \text{FTA}$ we obtain, by Lemma 28, a uniformizer u_1 in ELT. Then $u_1 \circ \tau_2$ is a uniformizer in ELT^2 for τ by Lemma 24. □

Theorem 30. $((\text{ELT}^{\text{R}})^*)^{-1}$ has uniformizers in $(\text{ELT}^{\text{R}})^*$.

Proof. The proof is analogous to the one of Theorem 27. Let $\tau_1, \ldots, \tau_n$ be ELT^{R} translations. We define ELT^{R} translations ρ_i such that $\rho_1^{-1} \circ \cdots \circ \rho_n^{-1} = \tau_1^{-1} \circ \cdots \circ \tau_n^{-1}$ and the sequence $\rho_1^{-1}, \ldots, \rho_n^{-1}$ is compatible, i.e., $\mathsf{ran}(\rho_i^{-1}) \subseteq \mathsf{dom}(\rho_{i+1}^{-1})$. We let $\rho_n = \tau_n$, and for $i \in [n-1]$ we let ρ_i be the domain restriction $\mathsf{id}_R \circ \tau_i$ of τ_i to $R = \mathsf{ran}(\rho_{i+1})$. The language R is regular by Corollary 9, and $\mathsf{id}_R \circ \tau_i$ is in ELT^{R} by Corollary 10. We obtain uniformizers in ELT^2 for the ρ_i^{-1} by Lemma 29, and a uniformizer in $(\text{ELT}^{\text{R}})^*$ for $\rho_1^{-1} \circ \cdots \circ \rho_n^{-1}$ by Lemma 24. □

The next corollary is immediate from Theorems 27 and 30.

Corollary 31. $((\text{ELT}^{\text{R}})^*)^{-1}$ has uniformizers in $(\text{DT}^{\text{R}}_{\text{fc}})^*$.

Example 32. We continue Example 23. Recall that M'' is the ELT transducer with the rule

$$q_0(\#_2(a(x_1, x_2, x_3), x_4)) \to a(b(q(x_1), q(x_2)), b(q(x_3), q(x_4)), e)$$

and the rules $q(b(x_1, x_2)) \to b(q(x_1), q(x_2))$ and $q(e) \to e$. The domain of $[\![M'']\!]$ is

$$\mathsf{dom}([\![M'']\!]) = \{\#_2(a(s_1, s_2, s_3), s_4) \mid s_1, s_2, s_3, s_4 \in T_{\{b,e\}}\}.$$

Clearly, $\mathsf{dom}([\![M'']\!]) = L(A)$ where $A = (Q, \Delta \cup H, F, \delta)$ with $Q = \{q, q_a, q_f, \overline{q}\}$, $F = \{q_f\}$, and $\delta(e) = q$, $\delta(b, q, q) = q$, $\delta(a, q, q, q) = q_a$, $\delta(\#_2, q_a, q) = q_f$ and $\delta(\dots) = \overline{q}$ in the remaining cases. With the terminology of the proof of Lemma 28, it is easy to see that $C(a, q_f, q, q, q) = \{\#_2(x_1, s) \mid s \in T_{\{b,e\}}\}$ and so we can take $C_0(a, q_f, q, q, q) = \#_2(x_1, e)$. Also, $C_0(b, q, q, q) = C_0(e, q) = x_1$. From this we obtain that a uniformizer u_1 of $\pi^{-1}_{\Delta,H} \circ [\![A]\!]$ is realized by the (deterministic) LT transducer with the rule

$$q_f(a(x_1, x_2, x_3)) \to \#_2(a(q(x_1), q(x_2), q(x_3)), e)$$

and the rules $q(b(x_1, x_2)) \to b(q(x_1), q(x_2))$ and $q(e) \to e$. Then a (functional) uniformizer u of $[\![M]\!]^{-1} = \pi^{-1}_{\Delta,H} \circ [\![M'']\!]$ in ELT^2 is $u = u_1 \circ [\![M'']\!]$. Composing the translations of the two transducers, it is clear that

$$u = \{(a(s_1, s_2, s_3)\,,\, a(b(s_1, s_2), b(s_3, e), e)) \mid s_1, s_2, s_3 \in T_{\{b,e\}}\}.$$

Thus, for the view $v = [\![M]\!]$, u is a uniformizer of v^{-1}, as already observed in Example 19. □

6.3. The Class of Views

As observed before, our class of views is fu-$(\text{ELT}^{\text{R}})^*$. It is immediate from Theorem 27 that every such view is in $(\text{DT}^{\text{R}}_{\text{fc}})^*$, as stated in the next corollary.[8]

Corollary 33. fu-$(\text{ELT}^{\text{R}})^* \subsetneq (\text{DT}^{\text{R}}_{\text{fc}})^*$.

In view of Theorem 27 and Corollary 31, a natural question to ask now is whether $(\text{ELT}^{\text{R}})^*$ and $((\text{ELT}^{\text{R}})^*)^{-1}$ have uniformizers in fu-$(\text{ELT}^{\text{R}})^*$. We prove a slightly stronger statement. We first strengthen Corollary 26.

Lemma 34. ELT^{R} has uniformizers in fu-ELT^{R}.

Proof. Let $M = (Q, \Sigma, \Delta, I, R, A)$ be an ELT^{R} transducer, with $A = (P, \Sigma, F, \delta)$. The proof is similar to the one of Lemma 25, but more complicated. As in that proof, we may assume that I is a singleton and, by Lemma 4, that M is without dead ends.

We will construct an ELT^{R} transducer M' that simulates M and, at each moment of a computation, applies exactly one of the applicable rules of M. Since M is without dead ends, this ensures that $[\![M']\!]$ is a functional uniformizer of $[\![M]\!]$. We fix an arbitrary linear order on the rules of M. The transducer M' will always pick the first applicable rule of M, in this order.

Let $C \subseteq T_\Sigma(X)$ be the finite set of contexts in the left-hand sides of the rules of M. As in the proof of Lemma 11, for an X_k-context $C \in C$ and $p_1, \dots, p_k \in P$, let $A_{C,p_1,\dots,p_k}$ be an fta over Σ such that

$$L(A_{C,p_1,\dots,p_k}) = \{C[s_1, \dots, s_k] \mid s_i \in T_\Sigma, \delta(s_i) = p_i \text{ for } i \in [k]\}.$$

[8] The inclusion is proper by Corollary 9, because $\text{DT}^{\text{R}}_{\text{fc}}$ contains translations that do not preserve regularity.

It follows from Lemma 1 that there exist ftas $A' = (P', \Sigma, F', \delta')$ and $A'_{C,p_1,\dots,p_k} = (P', \Sigma, F_{C,p_1,\dots,p_k}, \delta')$ such that $L(A') = L(A)$ and $L(A'_{C,p_1,\dots,p_k}) = L(A_{C,p_1,\dots,p_k})$. These automata have the same set of states P' and the same transition function δ'.

We now define the ELT$^{\mathrm{R}}$ transducer $M' = (Q, \Sigma, \Delta, I, R', A')$ with the following set of rules R'. A rule $q(C) \to \zeta\,\langle p'_1, \dots, p'_k\rangle$ is in R' if and only if there is a rule $q(C) \to \zeta\,\langle p_1, \dots, p_k\rangle$ in R such that

- $\delta'(C, p'_1, \dots, p'_k) \in F_{C,p_1,\dots,p_k}$,[9] and
- there is no smaller (in the given order) rule $\overline{q}(\overline{C}) \to \overline{\zeta}\,\langle \overline{p}_1, \dots, \overline{p}_m\rangle$ in R such that $\overline{q} = q$ and $\delta'(C, p'_1, \dots, p'_k) \in F_{\overline{C},\overline{p}_1,\dots,\overline{p}_m}$.

It should be clear that M' satisfies the requirements. Formally, it is straightforward to show for every $q \in Q$ and $s \in T_\Sigma$ that $[\![q]\!]_{M'}(s) \subseteq [\![q]\!]_M(s)$ and that $[\![q]\!]_{M'}(s)$ is a singleton if $[\![q]\!]_M(s) \neq \emptyset$ (by induction on the size of s). Since I is a singleton, this shows that $[\![M']\!]$ is a functional uniformizer of $[\![M]\!]$. □

The next corollary strengthens Theorem 27. Its proof is the same as the one of Theorem 27, using Lemma 34 instead of Corollary 26.

Corollary 35. (ELT$^{\mathrm{R}}$)* has uniformizers in (fu-ELT$^{\mathrm{R}}$)*.

As a direct consequence of Corollary 35 we obtain that our class of views equals (fu-ELT$^{\mathrm{R}}$)*, as already observed in Section 3.

Corollary 36. fu-(ELT$^{\mathrm{R}}$)* = (fu-ELT$^{\mathrm{R}}$)*.

Note that, since (ELT$^{\mathrm{R}}$)* = (ELT$^{\mathrm{R}}$)^3 by [14] and, clearly, (ELT$^{\mathrm{R}}$)^3 has uniformizers in (fu-ELT$^{\mathrm{R}}$)^3, we have that fu-(ELT$^{\mathrm{R}}$)* = (fu-ELT$^{\mathrm{R}}$)^3.

Similarly to Corollary 35, we strengthen Corollary 31 in the next corollary. It follows immediately from Theorem 30 and Corollary 35.

Corollary 37. ((ELT$^{\mathrm{R}}$)*)$^{-1}$ has uniformizers in (fu-ELT$^{\mathrm{R}}$)*.

7. Decidability of Determinacy and Rewriting

Consider a query q, a view v, and a uniformizer u of v^{-1}, each of them being a partial function. By Lemma 18, q is determined by v if and only if $q = v \circ u \circ q$. For queries in DT$^{\mathrm{R}}$ or DMSOT, equivalence is decidable and they are effectively closed under left composition with DT$^{\mathrm{R}}_{\mathrm{fc}}$. Thus, if v and u are in (DT$^{\mathrm{R}}_{\mathrm{fc}}$)*, then we can decide determinacy. This holds for the views in the class (ELT$^{\mathrm{R}}$)*. In fact, if v is in this class, then v^{-1} has a uniformizer u in (DT$^{\mathrm{R}}_{\mathrm{fc}}$)* by Corollary 31, and v itself is in (DT$^{\mathrm{R}}_{\mathrm{fc}}$)* by Corollary 33.

The main results of this paper are presented in the next three theorems.

[9] Recall again that the transition function was extended to contexts at the end of Section 2.

Theorem 38. Determinacy is decidable for DT^{R} and for DMSOT under fu-$(\text{ELT}^{\text{R}})^*$.

Proof. Let $v \in$ fu-$(\text{ELT}^{\text{R}})^*$. By Corollary 33, v is in $(\text{DT}^{\text{R}}_{\text{fc}})^*$, effectively.[10] By Corollary 31, one can construct (transducers for) a uniformizer $u \in (\text{DT}^{\text{R}}_{\text{fc}})^*$ of v^{-1}. If a query q is a DT^{R} (DMSOT) translation, then so are $u \circ q$ and $v \circ u \circ q$, by Propositions 6, 14 and 15, effectively. We can decide if $q = v \circ u \circ q$ (i.e., if q is determined by v by Lemma 18) because equivalence is decidable for DT^{R} and DMSOT transducers by Propositions 7 and 16. □

The proof of Theorem 38 also proves the next theorem (taking $f = u \circ q$).

Theorem 39. Let $\mathcal{V} =$ fu-$(\text{ELT}^{\text{R}})^*$, $v \in \mathcal{V}$, and let q be a DT^{R} (DMSOT) translation such that q is determined by v. A DT^{R} (DMSOT) translation f can be constructed such that $q = v \circ f$. That is, DT^{R} and DMSOT are complete for $\mathcal{V}$-to-DT^{R} and $\mathcal{V}$-to-DMSOT rewritings, respectively.

As defined in [3], a class $\mathcal{Q}$ of queries (and views) *admits view-based rewritings* if $\mathcal{Q}$ is complete for $\mathcal{Q}$-to-$\mathcal{Q}$ rewritings, i.e., for every $q \in \mathcal{Q}$ and $v \in \mathcal{Q}$ such that q is determined by v, there is an $f \in \mathcal{Q}$ with $q = v \circ f$. Intuitively this means that the query language $\mathcal{Q}$ has the nice property that it need not be extended to allow rewriting of queries with respect to views.[11] In the next theorem we show that, considered as a class of (linear) queries, the class fu-$(\text{ELT}^{\text{R}})^*$ admits view-based rewritings. Recall that fu-$(\text{ELT}^{\text{R}})^*$ is a subclass of both classes of queries DT^{R} and DMSOT (effectively), by Corollary 33 and Propositions 6, 14, and 15.

Theorem 40. Let $\mathcal{V} =$ fu-$(\text{ELT}^{\text{R}})^*$. Determinacy is decidable for $\mathcal{V}$ under $\mathcal{V}$. Moreover, $\mathcal{V}$ is complete for $\mathcal{V}$-to-$\mathcal{V}$ rewritings.

Proof. The first statement is a special case of Theorem 38. To prove the second statement, let $q, v \in \mathcal{V}$ such that q is determined by v. By Corollary 37, v^{-1} has a uniformizer $u \in \mathcal{V}$. Then, by Lemma 18, $q = v \circ f$ with $f = u \circ q \in \mathcal{V}$. □

7.1. Weakly Determined Queries

A query q is determined by a view v if there exists a partial function f such that $q = v \circ f$. For practical purposes, this could be weakened to $q \subseteq v \circ f$. For a given input s, one first checks if $s \in \mathsf{dom}(q)$, and if so, obtains $q(s)$ as $f(v(s))$. We say that *q is weakly determined by v* if there exists a partial function f such that $q \subseteq v \circ f$. As an example consider $q = \{(1,1)\}$ and $v = \{(1,1),(2,1)\}$. Then q is *not* determined by v, but is weakly determined by v. Let $\mathcal{Q}, \mathcal{V}, \mathcal{R}$ be classes

[10] This means that if $v = [\![M_1]\!] \circ \cdots \circ [\![M_k]\!]$ for given ELT^{R} transducers $M_1, \ldots, M_k$, then one can construct $\text{DT}^{\text{R}}_{\text{fc}}$ transducers $M'_1, \ldots, M'_n$ such that $v = [\![M'_1]\!] \circ \cdots \circ [\![M'_n]\!]$.

[11] Similarly, in the other direction, it is nice when $\mathcal{Q}$ is closed under composition: for every view $v \in \mathcal{Q}$ and every query $q \in \mathcal{Q}$ "on that view", the "viewless" query $v \circ q$ is in $\mathcal{Q}$. In this case one could say that $\mathcal{Q}$ admits removal of views. This holds for DT^{R}, DMSOT and fu-$(\text{ELT}^{\text{R}})^*$.

of partial functions. We say that $\mathcal{R}$ is *complete for weak $\mathcal{V}$-to-$\mathcal{Q}$ rewritings*, if for every $q \in \mathcal{Q}$ and $v \in \mathcal{V}$ such that q is weakly determined by v, there is an $f \in \mathcal{R}$ with $q \subseteq v \circ f$.

We now show that our main results also hold for weak determinacy and weak rewritings. Clearly, q is weakly determined by v if and only if q is determined by the domain restriction $\mathsf{id}_R \circ v$ of v to $R = \mathsf{dom}(q)$, with the same functions f involved. For $q \in \text{DT}^\text{R}$, $\mathsf{dom}(q)$ is effectively regular by [22, Corollary 2.7]; the same holds for $q \in$ DMSOT because $\mathsf{dom}(q)$ is MSO definable by definition, and hence regular by [31, 32]. And if $v \in \text{fu-}(\text{ELT}^\text{R})^*$, then $\mathsf{id}_R \circ v$ is in $\text{fu-}(\text{ELT}^\text{R})^*$ for every regular tree language R, by Corollary 10. This shows that Theorems 38 and 39 also hold for weak determinacy.

Corollary 41. Let $\mathcal{V} = \text{fu-}(\text{ELT}^\text{R})^*$. Weak determinacy is decidable for DT^R and for DMSOT under $\mathcal{V}$. The classes DT^R, DMSOT, and $\mathcal{V}$ are complete for weak $\mathcal{V}$-to-DT^R, weak $\mathcal{V}$-to-DMSOT, and weak $\mathcal{V}$-to-$\mathcal{V}$ rewritings, respectively.

7.2. Extended Bottom-Up Tree Transducers

Finally, we compare our results with those in [6, 7]. The views in that paper are (partial functions) realized by extended linear *bottom-up* tree transducers, and the queries are realized by (ordinary) *bottom-up* tree transducers.

An *extended bottom-up tree transducer* (EB transducer, for short) is a tuple $M = (Q, \Sigma, \Delta, F, R)$ where Q is a ranked alphabet of states all of rank 1, Σ and Δ are ranked alphabets of input and output symbols, respectively, $F \subseteq Q$ is a set of final states, and R is a finite set of rules of the form

$$C[q_1(x_1), \ldots, q_k(x_k)] \to q(\zeta)$$

where $k \geq 0$, $C \neq x_1$ is an X_k-context over Σ, $q_1, \ldots, q_k, q \in Q$, and $\zeta \in T_\Delta(X_k)$. For every state $q \in Q$ we define the *q-translation* $[\![q]\!]_M \subseteq T_\Sigma \times T_\Delta$ as follows. For an input tree $s \in T_\Sigma$, the q-translation $[\![q]\!]_M(s)$ of s is the smallest set of trees $T \subseteq T_\Delta$ such that for every rule $C[q_1(x_1), \ldots, q_k(x_k)] \to q(\zeta)$ and all $s_1, \ldots, s_k \in T_\Sigma$, if $s = C[s_1, \ldots, s_k]$ and $t_i \in [\![q_i]\!]_M(s_i)$ for every $i \in [k]$, then T contains the tree $\zeta[x_i \leftarrow t_i \mid i \in [k]]$. The *translation realized by M* is $[\![M]\!] = \cup_{q \in F} [\![q]\!]_M$.

The transducer M is *linear* (an ELB transducer), if the right-hand side $q(\zeta)$ of each rule is linear, i.e., each variable x_i occurs at most once in ζ.

The transducer M is an (ordinary, not extended) *bottom-up tree transducer* (B transducer) if the left-hand side context C of each of its rules contains exactly one symbol in Σ, i.e., each rule is of the form $a(q_1(x_1), \ldots, q_k(x_k)) \to q(\zeta)$ with $a \in \Sigma^{(k)}$ and $k \geq 0$.

Extended bottom-up tree transducers are studied in, e.g., [10, 13, 6]. In [13, 6, 7], the left-hand side of a rule is allowed to be of the form $q_1(x_1)$ with $q_1 \in Q$, and the corresponding (larger) class of translations is denoted XBOT. It is easy to show that such rules can effectively be removed from a transducer M when it is known

that $[\![M]\!]$ is a partial function (as is the case for views and queries), in such a way that linearity of M is preserved.[12]

In the next proposition we show that for linear extended transducers, top-down (with look-ahead) gives the same translations as bottom-up, just as for nonextended transducers (see [22, Theorem 2.8]). This result was already pointed out below Proposition 5 in [13] (see also [33, Theorem 3.1]). Based on Proposition 8, its proof is a standard one, relating bimorphisms to finite-state transducers (see [34] and [10]).

Proposition 42. $\text{ELT}^{\text{R}} = \text{ELB}$.

Proof. Let BIM denote the class of all tree translations $\{(f(r), g(r)) \mid r \in R\}$ where f is a linear nondeleting nonerasing tree homomorphism, g is a linear tree homomorphism, and R is a regular tree language. In accordance with Proposition 8, we show that BIM = ELB.

BIM $\subseteq$ ELB: Let $\tau = \{(f(r), g(r)) \mid r \in R_\tau\}$ be a BIM translation, where $f : T_\Omega \to T_\Sigma$, $g : T_\Omega \to T_\Delta$, and $R_\tau \subseteq T_\Omega$. Let $A = (Q, \Omega, F, \delta)$ be an fta accepting R_τ. We construct the ELB transducer $M = (Q, \Sigma, \Delta, F, R)$ with the following rules. Let $\omega \in \Omega^{(k)}$, $k \geq 0$, and let $q, q_1, \ldots, q_k \in Q$ such that $\delta(\omega, q_1, \ldots, q_k) = q$. Then R contains the rule $C[q_1(x_1), \ldots, q_k(x_k)] \to q(\zeta)$, where $C = f_\omega$ and $\zeta = g_\omega$. Note that since f is linear and nondeleting, C is an X_k-context, and since f is nonerasing, $C \neq x_1$. Also, ζ is linear because g is.

It should be clear that $[\![M]\!] = \tau$. Formally, it is straightforward to show, for every $q \in Q$, $s \in T_\Sigma$ and $t \in T_\Delta$, that $t \in [\![q]\!]_M(s)$ if and only if there exists $r \in T_\Omega$ such that $\delta(r) = q$, $f(r) = s$ and $g(r) = t$. The proof in the if direction is by induction on the structure of r, and the one in the only-if direction by induction on the size of s.

ELB $\subseteq$ BIM: This inclusion holds by the proof of Lemma 6 of [13]. For completeness' sake we also prove it here. Let $M = (Q, \Sigma, \Delta, F, R)$ be an ELB transducer. We turn the set of rules R into a ranked alphabet Ω by defining the rank of a rule $\omega : C[q_1(x_1), \ldots, q_k(x_k)] \to q(\zeta)$ to be k. We define the regular tree language $D = L(A)$ where $A = (Q \cup \{\bot\}, \Omega, F, \delta)$ is the fta such that, for the rule ω above, $\delta(\omega, q_1, \ldots, q_k) = q$ and $\delta(\omega, q'_1, \ldots, q'_k) = \bot$ for all $q'_1, \ldots, q'_k \in Q \cup \{\bot\}$ with $(q'_1, \ldots, q'_k) \neq (q_1, \ldots, q_k)$. Intuitively, D can be viewed as the set of derivation trees of M. Finally we define the tree homomorphisms f and g such that, for the rule ω above, $f_\omega = C$ and $g_\omega = \zeta$.

Let τ be the BIM translation $\{(f(r), g(r)) \mid r \in D\}$, and let the ELB transducer $M' = (Q \cup \{\bot\}, \Sigma, \Delta, F, R')$ be associated with τ as in the proof of BIM $\subseteq$ ELB above (with $R_\tau = D$). So, $[\![M']\!] = \tau$. Since $R \subseteq R'$ and the rules in $R' - R$ all lead to the nonfinal state $\bot$, it is obvious that $[\![M']\!] = [\![M]\!]$, and hence $\tau = [\![M]\!]$. □

[12] First, repeatedly add rules as follows: if $C[q_1(x_1), \ldots, q_k(x_k)] \to q(\zeta)$ and $q(x_1) \to q'(\zeta')$ are rules, then add the rule $C[q_1(x_1), \ldots, q_k(x_k)] \to q'(\zeta'[x_1 \leftarrow \zeta])$. Second, remove all rules with left-hand side in $Q(X)$.

Since the class fu-B of functional bottom-up tree translations is included in DT^{R} by [29], it is immediate from Theorem 38 and Proposition 42 that determinacy is decidable for fu-B under fu-ELB, as proved in Theorem 16 of [6] (Theorem 2 of [7]). In Theorem 21 of [6] (Theorem 3 of [7]) it is shown to be decidable for $q \in$ fu-B and $v \in$ fu-ELB whether there exists $f \in$ fu-B such that $q = v \circ f$. Theorem 39 shows that such an f can always be found in DT^{R}. Note that Theorem 38 extends [6, Theorem 16] in two ways: fu-B $\subsetneq \text{DT}^{\text{R}}$ (see [22, Section 3]) and fu-ELB = fu-$\text{ELT}^{\text{R}} \subsetneq$ fu-$(\text{ELT}^{\text{R}})^*$ (see the discussion after Proposition 7). In particular, the class of queries DT^{R} is much larger than fu-B because (as observed in [22, Section 3]) a DT^{R} transducer has the ability to copy an input subtree and to continue the translation of these copies in different states, whereas a B transducer can only put identical copies in the output tree (see also [35, Example 2.2]). Of course, the class of queries $\text{DT}^{\text{R}} \cup$ DMSOT is even far larger, because DMSOT contains tree translations that do not preserve the ancestor relation between nodes (such as taking the yield of the input tree, viewing that yield as a tree with symbols of rank 1 and 0 only).

8. Conclusion

We have shown that it is decidable whether a query q is determined by a view v when v is in fu-$(\text{ELT}^{\text{R}})^*$ and q is in DT^{R} or DMSOT. And if so, then q can be rewritten into a query f in the same class as q such that $q = v \circ f$. The main aim of this paper was to show that the proof of this result is quite straightforward, as it can be based on theoretical results that are either well known or easy to understand. Thus, we do not know whether the result has any practical relevance. In fact, we did not even investigate the complexity of our determinacy algorithm. A rough estimation shows that it works in 5-fold exponential time for the case of a fu-ELT^{R} view v and a DT^{R} query q. A double exponential blow-up is due to Lemmas 28 and 29 that construct two $\text{DT}^{\text{R}}_{\text{fc}}$ transducers for the uniformizer u of v^{-1}, because the involved FTA transducer can be of exponential size. Another exponential blow-up is due to Proposition 6: the composed transducer for $v \circ u \circ q$ may need exponentially many look-ahead states. Finally, the equivalence algorithm of Proposition 7 takes double exponential time (see [16]). It would be interesting to find special cases of lower complexity. As shown in [7], the determinacy algorithm of [6, 7] for a fu-ELB view v and a fu-B query q works in coNEXPTIME.

A tree transducer that is more expressive than both the DT^{R} transducer and the DMSOT transducer is the deterministic macro tree transducer (DMT transducer), see [25, Theorem 7.1]. Can our results be extended to the class of queries DMT? As should be clear from the beginning of Section 7, determinacy is decidable for $\mathcal{Q}$ under fu-$(\text{ELT}^{\text{R}})^*$, whenever $\mathcal{Q}$ is a class of deterministic transducers such that

- $\text{DT}^{\text{R}}_{\text{fc}} \circ \mathcal{Q} \subseteq \mathcal{Q}$, effectively, and
- it is decidable for two $\mathcal{Q}$ transducers whether they are equivalent.

It follows from Theorems 6.15 and 7.6 of [36], that the first statement holds for Q = DMT. However, decidability of equivalence of DMT transducers is a long-standing open problem, see [37]. It was recently shown in [38] that equivalence of $y\text{DT}^{\text{R}}$ transducers is decidable. These are DT^{R} transducers that output the yield of the output tree instead of the output tree itself. They can be viewed as a special case of DMT transducers. It follows from Proposition 6 that the first statement holds for $Q = y\text{DT}^{\text{R}}$. Hence determinacy is decidable for $y\text{DT}^{\text{R}}$ under fu-$(\text{ELT}^{\text{R}})^*$. This shows the robustness of our approach.

We finally note that it is interesting and practically important (for XML) to study determinacy for unranked tree transducers, e.g., those of [39].

Acknowledgements

We thank the reviewers for their critical comments which helped to improve the quality of the paper, and in particular one of the reviewers for suggesting a shorter proof of Lemma 20.

References

[1] L. Segoufin, V. Vianu, Views and queries: determinacy and rewriting, in: C. Li (Ed.), Proc. PODS 2005, ACM Press, 2005, pp. 49–60.

[2] A. Nash, L. Segoufin, V. Vianu, Views and queries: Determinacy and rewriting, ACM Trans. Database Syst. 35 (3), article 21.

[3] M. Marx, Queries determined by views: pack your views, in: L. Libkin (Ed.), Proc. PODS 2007, ACM Press, 2007, pp. 23–30.

[4] F. N. Afrati, Determinacy and query rewriting for conjunctive queries and views, Theor. Comput. Sci. 412 (11) (2011) 1005–1021.

[5] D. Pasailă, Conjunctive queries determinacy and rewriting, in: T. Milo (Ed.), Proc. ICDT 2011, ACM Press, 2011, pp. 220–231.

[6] K. Hashimoto, R. Sawada, Y. Ishihara, H. Seki, T. Fujiwara, Determinacy and subsumption for single-valued bottom-up tree transducers, in: A.-H. Dediu, C. Martin-Vide, B. Truthe (Eds.), Proc. LATA 2013, Vol. 7810 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 335–346.

[7] K. Hashimoto, R. Sawada, Y. Ishihara, H. Seki, T. Fujiwara, Determinacy and subsumption of single-valued bottom-up tree transducers, IEICE Transactions 99-D (3) (2016) 575–587.

[8] Z. Ésik, On decidability of injectivity of tree transformations, in: Les arbres en algèbre et en programmation, Lille, 1978, pp. 107–133.

[9] Z. Fülöp, P. Gyenizse, On injectivity of deterministic top-down tree transducers, Inf. Proc. Lett. 48 (4) (1993) 183–188.

[10] A. Arnold, M. Dauchet, Bi-transductions de forêts, in: S. Michaelson, R. Milner (Eds.), Proc. ICALP 1976, Edinburgh University Press, 1976, pp. 74–86.

[11] A. Maletti, Compositions of extended top-down tree transducers, Inf. Comput. 206 (9-10) (2008) 1187–1196.

[12] A. Maletti, J. Graehl, M. Hopkins, K. Knight, The power of extended top-down tree transducers, SIAM J. Comput. 39 (2) (2009) 410–430.

[13] J. Engelfriet, E. Lilin, A. Maletti, Extended multi bottom-up tree transducers, Acta Inf. 46 (8) (2009) 561–590.

[14] J. Engelfriet, Z. Fülöp, A. Maletti, Composition closure of linear extended top-down tree transducers, Theory Comput. Syst. Doi:10.1007/s00224-015-9660-2.

[15] Z. Ésik, Decidability results concerning tree transducers I, Acta Cybern. 5 (1980) 1–20.

[16] J. Engelfriet, S. Maneth, H. Seidl, Deciding equivalence of top-down XML transformations in polynomial time, J. Comput. Syst. Sci. 75 (5) (2009) 271–286.

[17] J. Engelfriet, S. Maneth, The equivalence problem for deterministic MSO tree transducers is decidable, Inf. Proc. Lett. 100 (5) (2006) 206–212.

[18] M. Benedikt, J. Engelfriet, S. Maneth, Determinacy and rewriting of top-down and MSO tree transformations, in: K. Chatterjee, J. Sgall (Eds.), Proc. MFCS 2013, Vol. 8087 of Lecture Notes in Computer Science, Springer-Verlag, 2013, pp. 146–158.

[19] B. Groz, XML security views: Queries, updates, and schemas, Ph.D. thesis, Université Lille 1 (2012).

[20] B. Groz, S. Staworko, A. Caron, Y. Roos, S. Tison, Static analysis of XML security views and query rewriting, Inf. Comput. 238 (2014) 2–29.

[21] H. Seidl, Equivalence of finite-valued tree transducers is decidable, Math. Systems Theory 27 (4) (1994) 285–346.

[22] J. Engelfriet, Top-down tree transducers with regular look-ahead, Math. Systems Theory 10 (1977) 289–303.

[23] B. Courcelle, J. Engelfriet, Graph Structure and Monadic Second-Order Logic – a Language-Theoretic Approach, Cambridge University Press, 2012.

[24] R. Bloem, J. Engelfriet, A comparison of tree transductions defined by monadic second order logic and by attribute grammars, J. Comput. Syst. Sci. 61 (1) (2000) 1–50.

[25] J. Engelfriet, S. Maneth, Macro tree transducers, attribute grammars, and MSO definable tree translations, Inf. Comput. 154 (1) (1999) 34–91.

[26] J. Engelfriet, S. Maneth, Macro tree translations of linear size increase are MSO definable, SIAM J. Comput. 32 (4) (2003) 950–1006.

[27] R. Alur, L. D'Antoni, Streaming tree transducers, in: A. Czumaj, K. Mehlhorn, A. Pitts, R. Wattenhofer (Eds.), Proc. ICALP 2012, Part II, Vol. 7392 of Lecture Notes in Computer Science, Springer-Verlag, 2012, pp. 42–53.

[28] M. Dauchet, Transductions de forêts, bimorphismes de magmoïdes, Ph.D. thesis, Université de Lille 1 (1977).

[29] J. Engelfriet, On tree transducers for partial functions, Inf. Proc. Lett. 7 (4) (1978) 170–172.

[30] J. Engelfriet, G. Rozenberg, G. Slutzki, Tree transducers, L systems, and two-way machines, J. Comput. Syst. Sci. 20 (2) (1980) 150–202.

[31] J. Doner, Tree acceptors and some of their applications, J. Comput. Syst. Sci. 4 (5) (1970) 406–451.

[32] J. W. Thatcher, J. B. Wright, Generalized finite automata theory with an application to a decision problem of second-order logic, Math. Systems Theory 2 (1) (1968) 57–81.

[33] Z. Fülöp, A. Maletti, H. Vogler, Weighted extended tree transducers, Fundam. Inform. 111 (2) (2011) 163–202.

[34] M. Nivat, Transduction des langages de Chomsky, Annales de l'Institut Fourier 18 (1968) 339–456.

[35] J. Engelfriet, Bottom-up and top-down tree transformations - a comparison, Math. Systems Theory 9 (3) (1975) 198–231.

[36] J. Engelfriet, H. Vogler, Macro tree transducers, J. Comput. Syst. Sci. 31 (1) (1985) 71–146.

[37] S. Maneth, Equivalence problems for tree transducers: a brief survey, in: Z. Ésik, Z. Fülöp (Eds.), Proc. AFL 2014, Vol. 151 of EPTCS, 2014, pp. 74–93.

[38] H. Seidl, S. Maneth, G. Kemper, Equivalence of deterministic top-down tree-to-string transducers is decidable, in: V. Guruswami (Ed.), Proc. FOCS 2015, IEEE Computer Society, 2015, pp. 943–962.

[39] T. Perst, H. Seidl, Macro forest transducers, Inf. Proc. Lett. 89 (3) (2004) 141–149.