PAPER Special Section on Formal Approach

Query Rewriting for Nondeterministic Tree Transducers

Kazuki MIYAHARA[†a)], *Student Member*, **Kenji HASHIMOTO**[††], *Member*, *and* **Hiroyuki SEKI**[†,††], *Fellow*

SUMMARY We consider the problem of deciding whether a query can be rewritten by a nondeterministic view. It is known that rewriting is decidable if views are given by single-valued non-copying devices such as compositions of single-valued extended linear top-down tree transducers with regular look-ahead, and queries are given by deterministic MSO tree transducers. In this paper, we extend the result to the case that views are given by nondeterministic devices that are not always single-valued. We define two variants of rewriting: universal preservation and existential preservation, and discuss the decidability of them.

key words: *tree transducers, query rewriting, query preservation*

1. Introduction

Almost every computer program runs along the structure of input data. The structure of input data would be sometimes modified due to changes or updates of its specifications. In the case, the program is needed to be rewritten for being adapted to the new structure of input data, one then may naturally ask the fundamental question—is it possible to rewrite the program automatically without changing its operation results? Some solutions to the automatic program rewriting have been presented in database theory, in which the problem has been formalized as *query rewriting* or query answering, which are motivated by data integration and query optimization (see, e.g., surveys [1], [2]). Let $\mathcal{Q}$, $\mathcal{Q}'$ be classes of queries and $\mathcal{V}$ be a class of views. Query rewriting for $\mathcal{Q}$ to $\mathcal{Q}'$ under $\mathcal{V}$ is the problem deciding whether, given a query $q \in \mathcal{Q}$ and a view $v \in \mathcal{V}$, a mapping $q' \in \mathcal{Q}'$ can be constructed from q and v such that $q = v \circ q'$, where $v \circ q'$ denotes the composition of v and q' defined as $(v \circ q')(t) = q'(v(t))$ for every input t. If such a mapping q' exists, we say that q can be rewritten in terms of v. Especially, the case when $\mathcal{Q}' = \mathcal{Q}$ is called *query preservation*. This case is important because it is not required to use a more expressive class than $\mathcal{Q}$ to rewrite the query. If q can be rewritten in terms of v and $\mathcal{Q}' = \mathcal{Q}$, we say that v preserves q.

Various techniques of query rewriting (and query preservation) have been developed in relational database theory. Query rewriting for semi-structured data, especially, tree-structured data such as XML documents has received attention recently due to the enormous success of the model on the Web (see, e.g., [3]–[10]). In this paper, we focus on query rewriting for tree-structured data.

The case when both views and queries are single-valued (or deterministic) tree transducers was studied in [7], [10]. Single-valued tree transducers output just one tree for each input tree (see [7], [8], [10]). Our main contribution is to extend their results to nondeterministic tree transducers as views, which transform each input tree to a *set* of output trees. To our knowledge, no previous work has considered the query rewriting for nondeterministic tree transducers (that are not always single-valued) as views.

Nondeterminism of tree transducers is required in some applications, e.g., probabilistic database (see a survey [11]) and natural language processing. In machine translation, each sentence in a source natural language can possibly be translated into more than one sentence in a target language (see, e.g., [12]–[14]). Thus we need nondeterminism in tree transducers that model syntax-based machine translations (see, e.g., [15], [16]). Bilingual documents are essentially required to construct statistical syntax-based translators. The translation accuracy of statistical syntax-based translators depends heavily on quality and quantity of the documents that are used to construct the translators (see, e.g., [17]). However, preparing huge and high-quality bilingual documents requires many efforts and costs in general. Query rewriting for nondeterministic tree transducers suggests a solution to this problem, which we call *rewriting-based construction* of machine translators. For instance, let q, v be English-to-Japanese and English-to-French machine translators, respectively, realized by nondeterministic tree transducers. One can construct by the rewriting-based construction French-to-Japanese translator q' from q and v, if q can be rewritten in terms of v (see Fig. 1). Advantages of the rewriting-based construction of translators are: (1) the translation accuracy of q' is guaranteed to be almost the same as q, because $q(t) = q'(v(t))$ holds for every input t by the definition of query rewriting, (2) any bilingual documents are not required. In the above example, French-to-Japanese bilingual documents are not needed to construct the French-to-Japanese translator q'. Our work is partially motivated to establish the rewriting-based construction of machine translators.

Our results in this paper contribute to the rewriting-based construction by extending the previous work [7], [10]

Manuscript received July 15, 2015.
Manuscript revised November 10, 2015.
Manuscript publicized May 2, 2016.

[†] The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

[††] The authors are with the Graduate School of Information Science, Nagoya University, Nagoya-shi, 464–8601 Japan.

a) E-mail: kazuki-mi@is.naist.jp

DOI: 10.1587/transinf.2015FOP0007

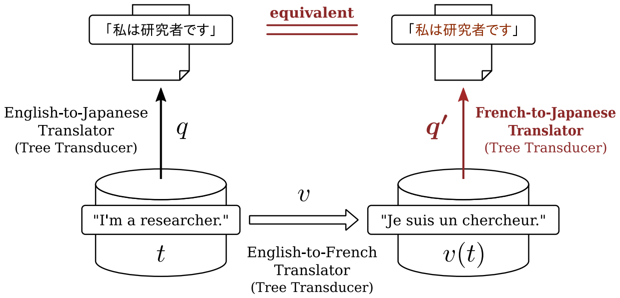

Fig. 1 An application of query rewriting to machine translations.

on the query rewriting from single-valued models to nondeterministic ones. More specifically, Hashimoto et al. showed in [7] that the query rewriting problem (deciding whether q can be rewritten in terms of v) is decidable when views are realized by single-valued extended linear bottom-up tree transducers and queries by single-valued bottom-up tree transducers. Benedikt et al. generalized in [10] the results of [7]. They showed that the problem is decidable when views are realized by compositions of single-valued extended linear top-down tree transducers with regular look-ahead and queries by deterministic MSO tree transducers. Note that the problem is undecidable even if the views can copy only once at each root of input trees [10]. Thus [7], [10] and we treat views that cannot copy[†]. Also, many applications of the query rewriting require the classes of q and q' coincide, so we focus on the query preservation.

As mentioned above, we generalize the results of [7] and [10] to that for nondeterministic views that are not always single-valued. We first define two variants of rewriting, which are natural extensions of query preservation for nondeterministic views: universal preservation and, its relaxed version, existential preservation.

Let $\mathcal{V}$, $\mathcal{Q}$ be classes of queries and views, respectively. Given a view $v \in \mathcal{V}$ and a query $q \in \mathcal{Q}$, v universally preserves q if there is a query $q' \in \mathcal{Q}$ such that for every input t to q and $\underline{\text{for every}}$ output $t' \in v(t)$, $q(t) = q'(t')$ holds; v existentially preserves q if there is a query q' such that for every input t to q $\underline{\text{there is}}$ an output $t' \in v(t)$ satisfying $q(t) = q'(t')$. Obviously, if v universally preserves q, then v existentially preserves q. Intuitively, if v universally preserves q then the result of q can be computed from any output of v. Whereas if v existentially preserves q then there exists at least one output of v from which the result of q can be computed. Existential preservation is useful in the case that the result of query is more important than that of view. For example, in the case of Fig. 1, v does not need to universally preserve q to construct q'. The French-to-Japanese translator q' can be constructed if at least one result of v preserves the result of q, that is, if v existentially preserves q then q' can be constructed.

To obtain the decidability of universal preservation for nondeterministic views, we first extend slightly the results in [20] on the equivalence problem for deterministic MSO tree transducers (see Theorem 9). Namely, we show that for a deterministic MSO tree transducer q_1 and a *nondeterministic* one q_2, the equivalence of q_1 and q_2 is decidable. Note that the equivalence is undecidable if q_1 and q_2 are both nondeterministic ones. Then, by adopting the proof strategy introduced in [10] that uses "uniformizers" and reduces the query preservation problem to the equivalence problem for deterministic MSO tree transducers, we obtained the desired result (Theorem 10), that is, we show that the problem is decidable when views are realized by compositions of nondeterministic extended linear top-down tree transducers with regular look-ahead and queries by deterministic MSO tree transducers. Summary of decidability results on query preservation is shown in Table 1, where B, DT, DMSOT, ELB, ELT stand for the classes of all transductions realized by bottom-up, deterministic top-down, deterministic MSO, extended linear bottom-up, and extended linear top-down tree transducers, respectively. "s-" is an abbreviation of single-valued, and the transducers in the classes with superscript R have regular look-ahead. The composition closure of a class X is denoted by X^*. See Sect. 2 for the definitions of them. As depicted in Table 1 there are class hierarchies of tree transductions such that s-ELB $\subsetneq$ $(\text{s-ELT}^{\text{R}})^*$ $\subsetneq$ $(\text{ELT}^{\text{R}})^*$ and s-B $\subsetneq$ DT^{R}. DMSOT is incomparable with DT^{R}.

Table 1 Decidability results on query preservation. Our result is indicated in bold. Incomparability is denoted by ⋈.

	single-valued				nondeterministic
Query \ View	s-ELB	⊊	$(\text{s-ELT}^{\text{R}})^*$	⊊	$(\text{ELT}^{\text{R}})^*$
s-B	decidable [7]				
⊊ DT^{R}			decidable [10]		
⋈ DMSOT			decidable [10]		**decidable** (Thm. 10)

Table 2 Summary of decidability results, where ∀ and ∃ stand for universal and existential preservation, respectively, "part" stands for the preservation for nondeterministic queries (see Sect. 4), and "sound" means that we give a sound algorithm of the problem for the classes in the line. Nondeterministic classes are indicated in bold.

Query	View	∀ or ∃	Result
DMSOT	**$(\text{ELT}^{\text{R}})^*$**	∀	decidable (Thm. 10)
DMSOT / DT^{R}	**finite-valued LB**	∃	sound (Thm. 13)
finite-valued LB	**$(\text{ELT}^{\text{R}})^*$**	∀ part	sound (Thm. 18)
finite-valued LB	**finite-valued LB**	∃ part	sound (Cor. 19)

Furthermore, we give an algorithm that is sound for existential preservation (see Theorem 13). Also, we give other algorithms that are sound for the related situations in which queries are expressed by nondeterministic tree transducers, specifically *finite*-valued bottom-up tree transducers (see Theorem 18 and Corollary 19). To show the results, we use the decomposition theorem for such transducers [21] (see Theorem 12). These results are summarized in Table 2. Note that finite-valued LB is a proper subclass of $(\text{ELT}^{\text{R}})^*$ and incomparable with DMSOT, namely, DMSOT ⋈ finite-valued LB $\subsetneq$ $(\text{ELT}^{\text{R}})^*$.

Related work. There have been interesting results on query rewriting, though not directly related to this paper. In [3], [18], it was shown that the query rewriting is undecid-

[†] Usually the non-copying property is called *linearity*.

able for the class of views and queries that are able to simulate first-order logic (FO) queries and projection queries, respectively, and even for views and queries expressed as unions of conjunctive queries (that are much weaker than FO queries), in the relational case. The problem was also shown to be undecidable in [3] for XSLT (or XQuery) as views and simple selection queries, in the XML context. In [4], views are defined as transformations that retrieve nodes selected by queries, such as Regular XPath and MSO queries in the context of unranked trees. Similarly, in [5] both queries and views are n-ary node-selecting queries represented by tree automata. In [8], a hybrid approach was adopted, that is, for views tree transducers were used and for queries n-ary node-selecting queries were adopted. In [8], query preservation was shown to be decidable if a given transducer is a single-valued extended linear top-down tree transducer with regular look-ahead and a given query is a run-based n-ary node-selecting query. The result in [19] is for deterministic tree transducers that require "origin" information.

Organization. This paper is organized as follows: Section 2 provides terminology and definitions of graphs, trees, tree automata, various tree transducers with their class hierarchies, and uniformizers. In Sect. 3, we show additional class hierarchies of tree transducers, which are needed to prove our main results. Section 4 defines universal and existential preservation for nondeterministic views. Furthermore, for nondeterministic queries two variants of query preservation are provided, which are analogue of existential preservation. Our main result on the decidability of universal preservation is presented in Sect. 5. The result is proved using a decidability result on the equivalence for deterministic and nondeterministic MSO tree transducers, which is included in the same section. A sound algorithm for existential preservation with single-valued queries is presented in Sect. 6, and others for nondeterministic queries are provided in Sect. 7. These sound algorithms employ a method that decomposes finite-valued tree transducers into a finite union of single-valued ones, and then the algorithms apply decision procedures of query preservation for single-valued tree transducers presented in the previous work [7], [10] to decomposed transducers. Section 6 also includes a solution for the problem of how to extend the domain of a view, which may be useful if a view does not preserve a query because of the difference between the domains of the view and the query. Section 8 concludes the paper and outlines future work.

2. Preliminaries

We denote the set of all nonnegative integers by $\mathbb{N}$. For $n \in \mathbb{N}$, the set $\{1, \ldots, n\}$ is denoted by $[n]$. A (ranked) alphabet is a finite set Σ of symbols with a mapping $\mathrm{rk} : \Sigma \to \mathbb{N}$, and let $\Sigma^{(n)}$ be the set $\{\, \sigma \in \Sigma \mid \mathrm{rk}(\sigma) = n \,\}$. For a binary relation $R \subseteq A \times B$, let $\mathrm{dom}(R) = \{\, a \in A \mid (a, b) \in R \,\}$, $\mathrm{ran}(R) = \{\, b \in B \mid (a, b) \in R \,\}$, $R^{-1} = \{\, (b, a) \in B \times A \mid (a, b) \in R \,\}$, and $R|_{A'} = \{\, (a, b) \in R \mid a \in A' \,\}$ for $A' \subseteq A$. The composition of relations $R_1 : A \to B$ and $R_2 : B \to C$, denoted by $R_1 \circ R_2$, is the relation $A \to C$ defined by $R_1 \circ R_2 = \{\, (a, c) \mid (a, b) \in R_1$ and $(b, c) \in R_2$ for some $b \in B \,\}$. For classes of binary relations $\mathcal{R}$, $\mathcal{S}$, we write $\mathcal{R} \circ \mathcal{S} = \{\, R \circ S \mid R \in \mathcal{R}, S \in \mathcal{S} \,\}$, $\mathcal{R}^* = \{\, R_1 \circ \cdots \circ R_n \mid n \geq 0, R_i \in \mathcal{R}\ (1 \leq i \leq n) \,\}$, and $\mathcal{R}^{-1} = \{\, R^{-1} \mid R \in \mathcal{R} \,\}$.

2.1 Graphs, Trees, Strings and MSO Graph Transducers

We basically follow the definitions of [20] and omit here formal definitions of monadic second-order (MSO) logic and MSO graph transducers [20], [22], [23] for the sake of simplicity.

A graph alphabet is a pair (Σ, Γ) where Σ and Γ are ranked alphabets of node labels and edge labels, respectively. A *graph over* (Σ, Γ) is a tuple (V, E, lab), with V a finite set of nodes, $E \subseteq V \times \Gamma \times V$ the set of labeled edges, and $\mathrm{lab} : V \to \Sigma$ the node-labeling function. The set of graphs over (Σ, Γ) is denoted by $G(\Sigma, \Gamma)$.

For an alphabet Δ and $a_1, \ldots, a_n \in \Delta$ $(n \geq 0)$, we identify the string $w = a_1, \ldots, a_n$ over Δ with the graph in $G(\{\#\}, \Delta)$ that has #-labeled nodes $v_1, \ldots, v_{n+1}$, and an a_i-labeled edge from v_i to v_{i+1} for $1 \leq i \leq n$.

Let Σ be a ranked alphabet and m be the maximal rank of symbols in Σ. The set of all trees over Σ is denoted by T_Σ. A tree whose root is labeled with $\sigma \in \Sigma^{(k)}$ and has subtrees $t_1, \ldots, t_k$ from left to right is denoted by $\sigma(t_1, \ldots, t_k)$. For $\sigma \in \Sigma^{(0)}$ we write $\sigma()$ as σ for simplicity.

For a deterministic MSO graph transducer M, we write $[\![M]\!]$ to denote the graph transduction $G(\Sigma_1, \Gamma_1) \to G(\Sigma_2, \Gamma_2)$ realized by M. Instead of $[\![M]\!](g)$ we write $M(g)$ by identifying a transducer M with its transduction $[\![M]\!]$ (and similarly for other transducers).

A (nondeterministic) MSO graph transducer M' is obtained from a deterministic one by allowing all formulas to have fixed free node-set variables called parameters. The transducer binds each parameter to a set of nodes of the input graph g satisfying the domain formula, then for each set of nodes, the node formulas and the edge formulas define the output graph as the deterministic one does. Thus, the graph transduction realized by M' is (not always a function but) a relation $[\![M']\!] \subseteq G(\Sigma_1, \Gamma_1) \times G(\Sigma_2, \Gamma_2)$.

An MSO graph transducer M is called an MSO graph-to-tree transducer if the range of M is a set of trees. For sets of graphs A and B, M is called an MSO A-to-B transducer if $\mathrm{dom}(M) \subseteq A$ and $\mathrm{ran}(M) \subseteq B$. In the case of $A = B$, M is called an MSO A transducer, e.g., an MSO tree transducer (abbreviated by MSOT transducer, and by DMSOT for deterministic one). The class of all transductions realized by MSOT transducers is denoted by MSOT, and similarly for other transducers.

The closure properties of MSO transducers under composition are shown, e.g., in Proposition 3.2(2) of [22] and Proposition 2 of [24].

Proposition 1. MSO graph transductions and (deterministic) MSO tree transductions are closed under composition.

2.2 Tree Automata and Tree Transducers

We omit here formal definitions of tree automata and several tree transducers (see, e.g., [13], [25]) for the sake of simplicity and instead provide their intuitive descriptions with their class hierarchies.

Deterministic bottom-up tree automata (TA) are a natural extension of deterministic (finite) word automata, which deal with ranked trees. A TA starts its computation at the leaves of an input tree and moves upward. The tree language accepted by a TA A is denoted by $L(A)$. A tree language L is *regular* if there exists a TA A such that $L = L(A)$.

Extended bottom-up tree transducers (EB transducers) are an extension of word transducers. An EB transducer M starts its transduction at the leaves of an input tree and moves upward. It can read a subtree of an input tree in one step. An EB transducer is *linear* if it cannot copy any subtrees of outputted trees during its computation. A *bottom-up tree transducer* (B transducer) is an EB transducer that can read only one symbol of an input tree in one step. We write (E)LB transducers to denote linear (E)B ones. In this paper, we use EB transducers that do not allow a transduction rule that does not read any subtree but only changes a state, which is called an input-ε rule (see, e.g., [7] and [25], in which the class of EB transducers with no input-ε rules is denoted by $\mathrm{XBOT}^{-\mathrm{e}}$).

Extended top-down tree transducers (ET transducers) are also an extension of word transducers. An ET transducer M starts its transduction at the root of an input tree and moves downward. M can read a subtree of an input tree in one step as with EB transducers. An ET transducer is *linear* if it cannot copy any subtrees of an input tree during its computation. A *top-down tree transducer* (T transducer) is an ET transducer that can read only one symbol of an input tree in one step. We write (E)LT transducers to denote linear (E)T ones. An *(extended) top-down tree transducer with regular look-ahead* ($(\mathrm{E})\mathrm{T}^{\mathrm{R}}$ transducer) M is an (E)T transducer that has a TA A. For a subtree t_s of an input tree, M first checks if $t_s \in L(A)$, and if so, transduces t_s. We write $(\mathrm{E})\mathrm{LT}^{\mathrm{R}}$ transducers to denote linear $(\mathrm{E})\mathrm{T}^{\mathrm{R}}$ ones. As with EB transducers, we use ET transducers that do not allow an input-ε rule (see, e.g., [13], [26]–[28], in which ET^{R} is denoted by $\mathrm{XTOP}^{\mathrm{R}}_{\mathrm{ef}}$ or e-$\mathrm{XTOP}^{\mathrm{R}}$).

For a tree transducer M and an input tree t, $\mathrm{val}_M(t) = |M(t)|$ denotes the number of different outputs of M for t, and let $\mathrm{val}(M) = \sup\{\, \mathrm{val}_M(t) \mid t \in T_\Sigma \,\}$, which is called *valuedness* of M. M is *finite-valued* if $\mathrm{val}(M) < \infty$. For any tree transducer M, we say that M is *single-valued (or functional)* if $\mathrm{val}(M) \le 1$, i.e., for each input t, there exists at most one output $M(t)$. We use the prefix 's' to denote the transducers, e.g., a single-valued EB transducer is denoted by an s-EB transducer. We also say that M is *finite-copying* (denoted with the subscript 'fc') if each subtree of the input tree is transduced by a bounded number of times (see [29], [30] for more formal definition of finite-copying). We denote by $[\![M]\!]$ the transduction realized by M that is the binary relation on trees. For transducers M_1 and M_2, we say that M_1 and M_2 are *equivalent* if $[\![M_1]\!] = [\![M_2]\!]$.

We will use the abbreviation such as EB, B for a class of tree transducers to denote the class of transductions realized by that class of tree transducers. By the definitions above the relations $\mathrm{B} \subseteq \mathrm{EB}$, $\mathrm{LB} \subseteq \mathrm{ELB}$, and $\mathrm{T} \subseteq \mathrm{T}^{\mathrm{R}} \subseteq \mathrm{ET}^{\mathrm{R}}$ hold. Additionally, as noted in Sect. 1, there are well-known class hierarchies of tree transducers such that $\text{s-ELB} \subsetneq (\text{s-ELT}^{\mathrm{R}})^* \subsetneq (\mathrm{ELT}^{\mathrm{R}})^*$, $\text{s-B} \subsetneq \mathrm{DT}^{\mathrm{R}} \bowtie \mathrm{DMSOT}$, and $\mathrm{DMSOT} \bowtie$ finite-valued $\mathrm{LB} \subsetneq (\mathrm{ELT}^{\mathrm{R}})^*$, where incomparability is denoted by $\bowtie$.

2.3 Uniformizers

We define uniformizers following [10]: Let R be a binary relation. A function u is a uniformizer of R if $u \subseteq R$ and $\mathrm{dom}(u) = \mathrm{dom}(R)$. For classes $\mathcal{T}$, $\mathcal{U}$ of transductions, 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 $u \in \mathcal{U}$. Benedikt et al. [10] showed that it is decidable whether v preserves q by reducing the problem to the equivalence of the query q and the composed mapping $v \circ u \circ q$, where u is a uniformizer of v^{-1}.

The following result is used later to prove our main result.

Theorem 2. [Theorem 11 of [10]] $((\mathrm{ELT}^{\mathrm{R}})^*)^{-1}$ has uniformizers in $\mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}}$.

3. Class Hierarchies of Tree Transducers

In the section, we show some class hierarchies of tree transducers that are needed in Sect. 5 and Sect. 7.

Theorem 3.

$$\begin{aligned}(\mathrm{ELT}^{\mathrm{R}})^* \circ \mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}} \circ \mathrm{DMSOT} &\subsetneq (\mathrm{ELT}^{\mathrm{R}})^* \circ \mathrm{DMSOT}\\ &\subsetneq (\mathrm{T}^{\mathrm{R}}_{\mathrm{fc}})^* \circ \mathrm{DMSOT}\\ &\subsetneq \mathrm{MSOT}.\end{aligned}$$

Proof. $\mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}} \subseteq \mathrm{DMSOT}$ by Theorem 7.4 of [32], and $\mathrm{ELT}^{\mathrm{R}} \subseteq \mathrm{T}^{\mathrm{R}}_{\mathrm{fc}}$ by the construction in the proof of Theorem 4.8 of [13], which states $\mathrm{ET}^{\mathrm{R}} = \mathrm{T}^{\mathrm{R}}$ (see also the paragraph under Corollary 18 of [26]), moreover,

$$\begin{aligned}\mathrm{T}^{\mathrm{R}}_{\mathrm{fc}} &\subseteq \mathrm{DBQREL} \circ \mathrm{T}_{\mathrm{fc}} \qquad (1)\\ &\subseteq \mathrm{DBQREL} \circ \mathrm{HOM}_{\mathrm{fc}} \circ \mathrm{LT} \qquad (2)\\ &\subseteq \mathrm{MSOT} \qquad (3)\end{aligned}$$

where DBQREL is the class of deterministic B transducers that can only relabel, and $\mathrm{HOM}_{\mathrm{fc}}$ is the class of finite-copying tree homomorphisms. (1) follows by the fact that $\mathrm{T}^{\mathrm{R}} \subseteq \mathrm{DBQREL} \circ \mathrm{T}$ (Theorem 2.6 [33]), and (2) follows by the construction in the proof of Lemma 3.6 of [34], which states $\mathrm{T} \subseteq \mathrm{HOM} \circ \mathrm{LT}$. It is not difficult to see that MSOT can simulate DBQREL, $\mathrm{HOM}_{\mathrm{fc}}$, and LT, then (3) follows by Proposition 1. □

We note that finite-valued $\mathrm{LB} \subsetneq (\mathrm{ELT}^{\mathrm{R}})^* \subsetneq \mathrm{MSOT}$.

Proposition 4. $\text{s-LB} \subsetneq \mathrm{DMSOT}$.

Proof. LB $\subsetneq$ ELB by the definition of ELB, ELB = ELT$^{\mathrm{R}}$ by Theorem 1 of [10], s-ELT$^{\mathrm{R}}$ $\subsetneq$ DT$^{\mathrm{R}}_{\mathrm{fc}}$ by Corollary 13 of [10], and DT$^{\mathrm{R}}_{\mathrm{fc}}$ $\subseteq$ DMSOT by Theorem 7.4 of [32]. □

4. Query Preservations

Let $\mathcal{V}$ and Q be a class of single-valued views and a class of single-valued queries for some tree-structured data. For a query $q \in Q$ and a view $v \in \mathcal{V}$, we say that *v preserves q* if there exists a query $q' \in Q$ such that $q(t) = (v \circ q')(t)$ for all tree t.

As mentioned in Sect. 1, previous studies on query preservation for tree transducers focus on single-valued (or deterministic) views. In contrast, nondeterministic views output a set of trees for each input tree. We consider two definitions of query preservation for nondeterministic views that are not always single-valued: Let $\mathcal{V}$ be a class of nondeterministic views and Q be a class of single-valued queries. Given a view $v \in \mathcal{V}$ and a query $q \in Q$ such that $\mathrm{dom}(q) \subseteq \mathrm{dom}(v)$, we say that *$v$ universally preserves q* ($\forall$-preserves q for short) if

$$\exists q' \in Q : (\ \mathrm{ran}(v) \subseteq \mathrm{dom}(q') \\ \wedge \forall t \in \mathrm{dom}(q) : q(t) = (v \circ q')(t)\ ).$$

The above definition coincides with the definition of the query preservation if v is single-valued. We also say that *v existentially preserves q* ($\exists$-preserves q for short) if

$$\exists q' \in Q.\ \forall t \in \mathrm{dom}(q).\ \exists t' \in v(t) :\ q(t) = q'(t').$$

This condition is equivalent to the following one: There exists a uniformizer u of v such that

$$\exists q' \in Q.\ \forall t \in \mathrm{dom}(q) :\ q(t) = (u \circ q')(t).$$

By definition, if q is universally preserved by v, then q is also existentially preserved by v.

Furthermore, for a class Q of nondeterministic queries we define additional two variants of query preservation, which are analogue of existential preservation. We say that *v $\forall$-preserves a part of q* if

$$\exists q' \in Q : (\ \mathrm{ran}(v) \subseteq \mathrm{dom}(q') \\ \wedge \forall t \in \mathrm{dom}(q).\ \exists t_q \in q(t) : t_q = (v \circ q')(t)\ ).$$

Similarly, we say that *v $\exists$-preserves a part of q* if

$$\exists q' \in Q.\ \forall t \in \mathrm{dom}(q).\ \exists t_q \in q(t).\ \exists t' \in v(t) :\ t_q = q'(t').$$

This condition is equivalent to the following one: There exists a uniformizer u of v such that

$$\exists q' \in Q.\ \forall t \in \mathrm{dom}(q).\ \exists t_q \in q(t) :\ t_q = (u \circ q')(t).$$

These two variants of query preservation restrict implicitly the class of q' to a single-valued one. Algorithms that are sound for the variants are presented in Sect. 7.

5. Universal Preservation

The main result of this section is Theorem 10: For $v \in$ (ELT$^{\mathrm{R}}$)* as a view and $q \in$ DMSOT as a query, it is decidable whether v $\forall$-preserves q. To obtain the result, we adopt the proof strategy taken by Benedikt et al. [10]. In [10], in order to show the decidability of the query preservation for views realized by (s-ELT$^{\mathrm{R}}$)* transducers and queries by DMSOT transducers, Benedikt et al. reduced this problem to the equivalence problem of two DMSOT transducers, which is known to be decidable [20]. According to the strategy we reduce the problem to the equivalence of MSOT transducers M_1 and M_2, where M_1 is deterministic and M_2 is *nondeterministic*. The previous result [20] is for the case that M_1 and M_2 are both deterministic MSOT transducers (see Theorem 8 below). We slightly extend the previous result (see Theorem 9). Note that if M_1 and M_2 are both nondeterministic MSOT transducers, the equivalence for them is known to be undecidable due to the negative result for nondeterministic word transducers [31], which are strictly less expressive than MSOT transducers.

To explain our result, let us summarize the decision procedure of [20] for the equivalence of DMSOT transducers. It can be shown that DMSOT transducers M_1 and M_2 are *not* equivalent if and only if there exist an input t and a position n such that the symbol at position n of $M_1(t)$ is different from the symbol at position n of $M_2(t)$. Hence, roughly speaking, the procedure tests whether a position n and distinct symbols a, b exist such that the pair (n, n) is contained in the set $S^{a,b}$ of all pairs (i, j) where a is the symbol at position i of $M_1(t)$ and b is the symbol at position j of $M_2(t)$, for some input t. In [20] the set $S^{a,b}$ is shown to be *semilinear* (defined below) and then the existence of a pair (n, n) in $S^{a,b}$ is decidable (as stated in Lemma 6). The set $S^{a,b}$ is constructed using *Parikh mapping* (also defined below).

Additional definitions are needed to describe the decision procedure precisely. For a string w, we denote by w/i the i-th letter of w. The Parikh mapping for graphs is the function $\mathrm{Par} : G(\Sigma, \Gamma) \to \mathbb{N}^k$ defined as $\mathrm{Par}(g) = (n_1, \ldots, n_k)$ where g is a graph over (Σ, Γ) with $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ and n_i is the number of σ_i-labeled nodes in g for $i \in [k]$. Similarly, the Parikh mapping for strings over $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ is the function $\mathrm{Par} : \Sigma^* \to \mathbb{N}^k$ defined as $\mathrm{Par}(w) = (n_1, \ldots, n_k)$ where n_i is the number of occurrences of σ_i in w for $i \in [k]$. A *discrete graph* (abbreviated as *dgraph*) is a graph that has no edges. Let dgr be a function $\Sigma^* \to G(\Sigma, \emptyset)$ such that $\mathrm{Par}(w) = \mathrm{Par}(\mathrm{dgr}(w))$ for any string $w \in \Sigma^*$. For a set G of graphs, we denote by $\mathrm{Par}(G)$ the set $\{\mathrm{Par}(g) \mid g \in G\}$. Similarly, for a string language L, let $\mathrm{Par}(L) = \{\mathrm{Par}(w) \mid w \in L\}$. A set $S \subseteq \mathbb{N}^k$ is *semilinear* if there exists a regular string language R such that $S = \mathrm{Par}(R)$. The set G is *Parikh* if $\mathrm{Par}(G)$ is semilinear. A set of graphs is *VR* if it is generated by a context-free vertex replacement graph grammar (or a C-edNCE or an S-HH grammar, see, e.g., [22], [23]). It should be noted that the set of all trees and the set of all strings are VR. The following two lemmas state useful properties of semilinear sets.

Lemma 5. [Theorem 7.1 of [22], Lemma 3 of [20]] The images of VR sets of graphs under MSO graph-to-dgraph transductions are Parikh.

Lemma 6. [Lemma 4 of [20]] It is decidable for a semi-linear set $S \subseteq \mathbb{N}^2$ whether there exists an $n \in \mathbb{N}$ such that $(n, n) \in S$.

Let us refer to the important lemma of [20].

Lemma 7. [Lemma 7 of [20]] Let a, b be distinct symbols and let M_1, M_2 be MSO graph-to-string transducers. There exists an MSO graph-to-dgraph transducer $M^{a,b}$ such that for every graph g,

$$M^{a,b}(g) = \{\, \mathrm{dgr}(a^m b^n) \mid \exists w_1 \in M_1(g),\ w_2 \in M_2(g) : \\ w_1/m = a \ \text{ and } \ w_2/n = b \,\}.$$

The decision procedure of [20] consists of three steps, in which the above three lemmas are used:

Step 1. Let M_1, M_2 be deterministic MSO tree transducers. Construct a deterministic MSO tree-to-string transducer W that "flattens" input trees to strings, then compose them with M_1, M_2, i.e., construct M_1' and M_2' with $[\![M_1']\!] = [\![M_1]\!] \circ [\![W]\!]$ and $[\![M_2']\!] = [\![M_2]\!] \circ [\![W]\!]$, respectively. Since DMSOT transductions are closed under composition (see Proposition 1), M_1' and M_2' are obtained as DMSOT transducers.

Step 2. Let $N^\$$ be a deterministic MSO string transducer that reads a given input string $w \in \Delta_1 \cup \Delta_2$ and outputs the string $w\$$, where $\$$ is a symbol not in $\Delta_1 \cup \Delta_2$. For $i \in [2]$, let $M_i^\$$ be a deterministic MSO tree-to-string transducer with $[\![M_i^\$]\!] = [\![M_i']\!] \circ [\![N^\$]\!]$. Now we can say that $[\![M_1]\!] \neq [\![M_2]\!]$ if and only if $[\![M_1^\$]\!] \neq [\![M_2^\$]\!]$[†].

Step 3. Let D_1 and D_2 be the domains of $M_1^\$$ and $M_2^\$$, respectively. Clearly, if $D_1 \neq D_2$ then $[\![M_1^\$]\!] \neq [\![M_2^\$]\!]$. It is decidable whether D_1 coincides with D_2 because the domain of every (D)MSOT transducer is regular (see, e.g., Theorem 5.82 of [23]). Then, assume $D_1 = D_2$ here and denote the domains of them by D. $[\![M_1^\$]\!] \neq [\![M_2^\$]\!]$ if and only if there exist $a \in \Delta_1 \cup \{\$\}$, $b \in \Delta_2 \cup \{\$\}$, $n \in \mathbb{N}$, and $t \in D$ such that $a \neq b$, $M_1^\$(t)/n = a$, and $M_2^\$(t)/n = b$. Let $M^{a,b}$ be the transducer of Lemma 7 for a, b, $M_1^\$$, and $M_2^\$$. $[\![M_1^\$]\!] \neq [\![M_2^\$]\!]$ iff $\mathrm{dgr}(a^n b^n)$ is in $M^{a,b}(D)$ for some $n \in \mathbb{N}$. Note that $\mathrm{dgr}(a^n b^n)$ is in $M^{a,b}(D)$ iff (n, n) is in $\mathrm{Par}(M^{a,b}(D))$. By Lemma 5 the set of vectors $\mathrm{Par}(M^{a,b}(D))$ is Parikh because D is regular and thus VR. By Lemma 6 it can be decided whether (n, n) is in $\mathrm{Par}(M^{a,b}(D))$ for some $n \in \mathbb{N}$. This proves the main result of [20].

Theorem 8. [Corollary 10 of [20]] It is decidable whether two DMSOT transducers are equivalent.

We now extend Theorem 8 slightly to show the decidability of the universal preservation for nondeterministic views.

Theorem 9. It is decidable whether a deterministic MSOT transducer M_1 and a nondeterministic MSOT transducer M_2 are equivalent.

Proof. Recall that the domain of every (D)MSOT is regular, so we can decide whether the domain of M_1 coincides with that of M_2. Let us assume $\mathrm{dom}(M_1) = \mathrm{dom}(M_2) = D$. Since Lemmas 5–7 described above are not restricted to deterministic MSOT transducers, the same idea for DMSOT can be applied to our case. Thus we can state that $M_1 \in$ DMSOT and $M_2 \in$ MSOT are not equivalent if and only if there exist $a \in \Delta_1 \cup \{\$\}$, $b \in \Delta_2 \cup \{\$\}$, $n \in \mathbb{N}$, $t \in D$, and $w \in M_2^\$(t)$ such that $a \neq b$, $M_1^\$(t)/n = a$, and $w/n = b$. The nondeterminism of M_2 makes differences in Steps 1–3, in which nondeterministic MSOT transducers M_2' and $M_2^\$$ such that $[\![M_2']\!] = [\![M_2]\!] \circ [\![W]\!]$ and $[\![M_2^\$]\!] = [\![M_2']\!] \circ [\![N^\$]\!]$ are constructed instead of deterministic ones. Recall that MSOT transductions are closed under composition (Proposition 1), i.e., MSOT $\circ$ MSOT $\subseteq$ MSOT, hence MSOT $\circ$ DMSOT $\subseteq$ MSOT. For an input tree t, $a \in \Delta_1 \cup \{\$\}$, and $b \in \Delta_2 \cup \{\$\}$, assume $M_1^\$(t) = w_1$ and $w_2 \in M_2^\$(t)$ with $i \in \mathbb{N}$, $w_1/i = a$, $w_2/i = b$, $a \neq b$. In the case, obviously $M_1^\$$ and $M_2^\$$ are not equivalent. Let $M^{a,b}$ be an MSOT transducer in Step 3, which can be (effectively) constructed because Lemma 7 is not restricted to DMSOT transducers. By the assumption, $\mathrm{dgr}(a^i b^i) \in M^{a,b}(t)$, and so $\mathrm{Par}(M^{a,b}(t))$ contains $(i, i) \in \mathbb{N}^2$. Conversely, for distinct symbols $c \in \Delta_1 \cup \{\$\}$ and $d \in \Delta_2 \cup \{\$\}$, assume $\mathrm{Par}(M^{c,d}(D))$ includes $(j, j) \in \mathbb{N}^2$ with $j \in \mathbb{N}$. By the assumption, there exist a tree $t \in D$, $w_1 \in \Delta_1^*$, $w_2 \in \Delta_2^*$ such that $M_1^\$(t) = w_1\$$, $w_2\$ \in M_2^\$(t)$, $w_1\$/j = c$, and $w_2\$/j = d$. Hence M_1 and M_2 are not equivalent. It is decidable whether there exists $(i, i) \in \mathbb{N}^2$ in $\mathrm{Par}(M^{a,b}(t))$ because Lemmas 5 and 6 are not restricted to DMSOT transducers. □

We are now ready to describe our main result.

Theorem 10. For $v \in (\mathrm{ELT}^{\mathrm{R}})^*$ as a view and $q \in$ DMSOT as a query, it is decidable whether v $\forall$-preserves q.

Proof. By Theorem 2 we can construct a $\mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}}$ transducer that realizes a uniformizer u of v^{-1}. Let $v \in (\mathrm{ELT}^{\mathrm{R}})^*$, $q \in$ DMSOT. We show that v $\forall$-preserves q if and only if $q = v \circ u \circ q$. The right-to-left direction is obvious. For the other direction, assume that v $\forall$-preserves q and let $\tilde{q}$ be a query in DMSOT such that $q = v \circ \tilde{q}$. Then, $v \circ u \circ q = v \circ u \circ v \circ \tilde{q} = v \circ \tilde{q} = q$ holds. Precisely, let $t \in \mathrm{dom}(q)$ and then,

$$\begin{aligned}
&q(u(v(t))) \\
&= \{\, q(t'') \mid t' \in v(t),\ t'' \in u(t') \,\} \\
&= \{\, \tilde{q}(v(t'')) \mid t' \in v(t),\ t'' \in u(t') \,\} \ \ (\because q = v \circ \tilde{q}) \\
&= \{\, \tilde{q}(t') \mid t' \in v(t) \,\} \ (\because v \circ \tilde{q} \text{ is single-valued, } t' \in v(t'')) \\
&= \tilde{q}(v(t)) \\
&= q(t).
\end{aligned}$$

It follows that $u \circ q \in$ DMSOT by Proposition 1 and Theorem 7.4 of [32] that states $\mathrm{DT}^{\mathrm{R}}_{\mathrm{fc}} \circ$ DMSOT $\subseteq$ DMSOT, and

[†] If we do not compose $N^\$$ with M_i', then the following procedure answers "M_1 and M_2 are equivalent" incorrectly if one of them always outputs "prefix strings" of the other, i.e., for every input tree t and let $w = M_1'(t)$, there exists a string $z \neq \varepsilon$ such that $M_2'(t) = w \cdot z$, where $\cdot$ is concatenation of strings.

$v \circ u \circ q \in$ MSOT by Theorem 3. We can decide the equivalence of q and $v \circ u \circ q$ by Theorem 9. □

6. Existential Preservation

Even when v does not universally preserve q, it is still too early to give up on preserving information of q. In the case, we would like to know whether v existentially preserves q. There is a simple relation between the existential preservation and the universal preservation.

Proposition 11. Let v, v' be (nondeterministic) views and q be a single-valued query. If $v' \subseteq v$ and v' $\forall$-preserves q, then v $\exists$-preserves q.

Based on Proposition 11, we give a decidable sufficient condition of the existential preservation. We only focus on *finite*-valued views, because we use a decomposition theorem for finite-valued B transducers (recall that the results in the previous section focus not only on finite-valued views but also on "infinite"-valued ones). Note that it is decidable in deterministic polynomial time whether a B transducer is finite-valued or not (Theorem 6.9 of [35]).

The following theorem states that every finite-valued B transducer can be effectively decomposed into a finite union of single-valued ones.

Theorem 12. [Theorem 6.2 of [21]] For every finite-valued B transducer M, there exist single-valued ones $M_1, \ldots, M_K$ such that $[\![M]\!] = [\![M_1]\!] \cup \cdots \cup [\![M_K]\!]$, where $K \leq 2^{2^{P(|M|)}}$, $|M_j| \leq 2^{2^{P(|M|)}}$, $j \in [K]$, for some polynomial P independent of M[†].

By Proposition 11 and Theorem 12, we obtain a decidable sufficient condition of the existential preservation for views realized by finite-valued LB transducers and queries realized by DMSOT (or DT^R) transducers as follows. Notice that, in the following theorem, each v_j is single-valued, hence we just say "v_j preserves" instead of "v_j $\forall(\exists)$-preserves."

Theorem 13. Let v be a finite-valued LB transduction and q be a DMSOT (or DT^R) transduction. Let $v_1, \ldots, v_K$ be s-LB transductions such that $v = v_1 \cup \cdots \cup v_K$. Then, v $\exists$-preserves q if v_j preserves q for some $j \in [K]$. The latter condition is decidable.

Proof. By Proposition 11, v $\exists$-preserves q if v_j preserves q for some j. Every s-LB transduction is in $(\text{s-ELT}^\text{R})^*$. Thus the preservation is decidable by Theorem 15 of [10], which states that it is decidable for $\bar{v} \in (\text{s-ELT}^\text{R})^*$ and $\bar{q} \in$ DMSOT (or DT^R), whether $\bar{v}$ preserves $\bar{q}$ (see Table 1). □

Let $\exists$-Pres be the algorithm that decides the above sufficient condition by using Theorem 12. We are not sure whether $\exists$-Pres (and the other algorithms that are sound in the rest of this paper) is complete for existential preservation, because even when $\exists$-Pres answers "no," another s-LB transduction v' may exist such that $v' \subsetneq v$ and v' preserves q. The problem of deciding whether such v' exists seems to be hard, because one is required to prove a given query q is not preserved by v_i ($i \in [K]$) for *every* possible way of decomposing a finite-valued tree transduction v into single-valued ones $v_1, \ldots, v_K$.

Extending the domain of a view. Let v be a view given by a finite-valued LB transducer and q be a query given by a DMSOT transducer. When $\exists$-Pres answers "no," there remains a possibility that v existentially preserves q. When decomposing a view v into $v_1, \ldots, v_K$ by Theorem 12, the domain of the resulting transduction v_i may be a proper subset of the domain of the original transduction v. Consider the case when $\text{dom}(q) \supsetneq \text{dom}(v_i)$ for each $i \in [K]$. In the case, by the definition of query preservation, every v_i does not preserve q. Still, there may be $S \subsetneq [K]$ such that the union v' of transductions $\forall$-preserves q where $v' = \cup_{j \in S} v_j$. If so, we can conclude that $\cup_{j \in [K]} v_j$ $\exists$-preserves q. Hence the following result holds as a corollary of Theorem 10. Note that every union of s-LB transductions is an LB transduction.

Corollary 14. Let $v_1, \ldots, v_K \in$ s-LB and $q \in$ DMSOT. It is decidable whether there is $S \subsetneq [K]$ such that v' $\forall$-preserves q where $v' = \cup_{j \in S} v_j$. If such an S exists, $\cup_{j \in [K]} v_j$ $\exists$-preserves q (by Proposition 11).

Let q be a query given by an s-B transducer. Since s-B is incomparable with DMSOT, we cannot apply Theorem 10 directly to obtain a decidable sufficient condition similar to the one stated in Corollary 14 for $q \in$ s-B. For the case, in order to construct an appropriate view $v' (\subseteq v)$ from $v_1, \ldots, v_K$ obtained by decomposing v by Theorem 12, each component v_j of v' is required not to be *joinable* with another. For views v_i, v_j and a query q, we say that v_i and v_j are joinable against q if there exists a pair of trees (t_1, t_2) such that $q(t_1) \neq q(t_2)$ and $v_i(t_1) = v_j(t_2)$. To show why transductions must not be joinable, let us suppose that v_i and v_j are joinable against q with a pair (t_1, t_2) and $v' = v_i \cup v_j$ (at least) existentially preserves q, then there exists q' in the same class as q's (s-B) such that for all $t \in \text{dom}(q)$ there exists $t' \in v'(t)$ satisfying $q(t) = q'(t')$. However, $q'(t'_{1,2})$ for $t'_{1,2} = v_i(t_1) = v_j(t_2)$ has to be $\{q(t_1), q(t_2)\}$ but q' is single-valued, which is a contradiction. Thus, each component v_j of v' is required not to be joinable against q with another. We show that joinability is decidable by using the single-valuedness test for EB transducers with "grafting" presented in [7].

Lemma 15. Let $v_1, v_2 \in$ s-LB and $q \in$ s-B. It is decidable whether v_1 and v_2 are joinable against q.

Proof. Construct ELB transducers with grafting ($\text{ELB}^{+\text{g}}$ for short) that realize v_1^{-1} and v_2^{-1}, respectively. The detailed construction is given in [7] (see Lemma 6 in it). Next, construct a TA A that accepts the intersection of the ranges of v_1 and v_2, i.e., $L(A) = \text{ran}(v_1) \cap \text{ran}(v_2)$. A can be effectively constructed by the fact that the range of every LB transducer

[†] Here the size of a B transducer M is denoted by $|M|$, which is the sum of the sizes of all its rules (see [21] for more formal definition).

is effectively regular (see, e.g., Corollary 3.11 of [34]) and by the construction of product automaton. After that, construct an ELB^{+g} transducer that realizes $w = v_1^{-1} \cup v_2^{-1}|_{L(A)}$, which is the union of v_1^{-1} and v_2^{-1}, and the domain of which is restricted to $L(A)$. Finally, decide whether $q' = w \circ q$ is single-valued or not, which is decidable by Lemmas 7–8 of [7]. It is not difficult to show that v_1 and v_2 are joinable against q if and only if q' is *not* single-valued. The left-to-right direction holds obviously by the definition of joinability. Conversely, if q' is not single-valued, there exists a tree $t \in \mathrm{dom}(q') \subseteq \mathrm{ran}(v_1) \cap \mathrm{ran}(v_2)$ such that $|q'(t)| \geq 2$. Hence, there exist $t'_1, t'_2 \in q'(t)$ with $t'_1 \neq t'_2$. Since q is single-valued, there exist t_1, t_2 ($t_1 \neq t_2$) such that $q(t_1) = t'_1 \neq t'_2 = q(t_2)$ and $v_1(t_1) = v_2(t_2)$. Thus v_1 and v_2 are joinable against q. □

Lemma 16. Let $v_1, \ldots, v_K \in$ s-LB and $q \in$ s-B. The union v' ∀-preserves q if and only if (1) each component v_j of v' is not joinable against q with another, (2) v_j preserves $q|_{\mathrm{dom}(v_j)}$, and (3) $\mathrm{dom}(q) \subseteq \mathrm{dom}(v')$.

Proof. As described in the beginning of this subsection, if (1) does not hold, v' does not ∃-preserve q, and hence v' also does not ∀-preserve q. On the other side, suppose (1), (2), and (3) hold. For simplicity, suppose $K = 2$. By (2), let q'_1, q'_2 be s-LB transductions such that $q|_{\mathrm{dom}(v_i)} = v_i \circ q'_i$ ($i \in [2]$). For any $t_{12} \in \mathrm{dom}(v_1) \cap \mathrm{dom}(v_2)$, $q(t_{12}) = (v_1 \circ q'_1)(t_{12}) = (v_2 \circ q'_2)(t_{12})$. Hence for any $t \in \mathrm{dom}(v_1) \cup \mathrm{dom}(v_2)$, $q(t) = (v_1 \cup v_2) \circ (q'_1 \cup q'_2)(t)$ holds. Also, $q'_1 \cup q'_2$ is single-valued by (1). Therefore, $v_1 \cup v_2$ ∀-preserves $q|_{\mathrm{dom}(v_1) \cup \mathrm{dom}(v_2)}$. Generally, $v_1 \cup \cdots \cup v_K$ ∀-preserves $q|_{\mathrm{dom}(v_1) \cup \cdots \cup \mathrm{dom}(v_K)}$, therefore v' ∀-preserves q by (3). □

By Lemmas 15, 16, we obtain the following result for queries realized by s-B transducers.

Theorem 17. Let $v_1, \ldots, v_K \in$ s-LB and $q \in$ s-B. It is decidable whether there is $S \subseteq [K]$ such that v' ∀-preserves q where $v' = \cup_{j \in S}\, v_j$. If such an S exists, $\cup_{j \in [K]}\, v_j$ ∃-preserves q (by Proposition 11).

Proof. It suffices to show that the three conditions (1)–(3) in Lemma 16 are decidable. (1) is decidable by Lemma 15 and (2) is also decidable [7] (see Table 1). (3) is decidable due to the regularity of $\mathrm{dom}(q)$ and $\mathrm{dom}(v')$. □

7. Nondeterministic Queries

Let v be a view and q be a nondeterministic query. In this section, we show two algorithms that are sound for the problem of deciding whether v universally or existentially preserves *a part of* q (see Sect. 4 for definition).

We can adopt the idea of Theorem 13 to obtain an algorithm called ∀-PresPart that is sound for query preservation for finite-valued queries. ∀-PresPart is almost the same as ∃-Pres (see the previous section), but it decomposes a given query q into $q_1, \ldots, q_K$ instead of a given view v, after that for each $i \in [K]$ it tests whether v ∀-preserves q_i.

Theorem 18. For $v \in (\mathrm{ELT}^{\mathrm{R}})^*$ as a view and a finite-valued LB transduction q as a query, the algorithm ∀-PresPart is sound for the problem of deciding whether v ∀-preserves a part of q.

Proof. Suppose $q_1, \ldots, q_K \in$ s-LB that are obtained by decomposing q. By Proposition 4 and by Theorem 10, it is decidable whether v ∀-preserves q_i for each $i \in [K]$. Our algorithm ∀-PresPart answers "yes" (that means v ∀-preserves a part of q) if and only if there exists $j \in [K]$ such that v ∀-preserves q_j. □

By Theorems 13, 18, and by Theorem 21 of [7] that states query preservation is decidable for a view realized by an s-ELB transducer and a query realized by an s-B transducer (see Table 1), we also obtain an algorithm (called ∃-PresPart) for the case when views and queries are both finite-valued. We do not describe explicitly the algorithm ∃-PresPart here because it can easily be obtained by modifying slightly the algorithm ∀-PresPart.

Corollary 19. For a view v (resp. query q) realized by a finite-valued LB (resp. B) transducer, the algorithm ∃-PresPart is sound for the problem of deciding whether v ∃-preserves a part of q.

By the way, in [10] the "weak condition" is considered. For a view $v \in \mathcal{V}$ and a query $q \in \mathcal{Q}$, v (∀- or ∃-) preserves q if there exists a mapping $q' \in \mathcal{Q}$ such that (1) $\mathrm{dom}(v \circ q') = \mathrm{dom}(q)$ and (2) $(v \circ q')(t) = q(t)$ for every $t \in \mathrm{dom}(q)$. For some situation in practice, the condition (1) is weakened to $\mathrm{dom}(v \circ q') \supseteq \mathrm{dom}(q)$. The above results (Theorems, Lemmas, and Corollaries 10–19) can be applied to the weak condition by restricting $\mathrm{dom}(v)$ to $\mathrm{dom}(q) \cap \mathrm{dom}(v)$[†].

8. Conclusions and Open Problems

We have defined two kinds of query preservation problem for nondeterministic views and queries on ranked trees: universal preservation and existential preservation. We have proved that the universal preservation problem is decidable for compositions of extended linear top-down tree transducers with regular look-ahead as views and deterministic MSO tree transducers as queries (see Theorem 10). To obtain the result we have slightly generalized the result [20] of the equivalence problem for deterministic MSO tree transducers (see Theorem 9). Moreover, we have shown an algorithm that is sound for the existential preservation for finite-valued linear bottom-up tree transducers as views and deterministic MSO tree transducers as queries (see Theorem 13), and also shown additional algorithms that are sound for the problem for nondeterministic queries realized by finite-valued (linear) bottom-up tree transducers (see Theorem 18 and Corollary 19). We would like to know whether (1) a sound

[†] It is known that $\mathrm{dom}(q)$ and $\mathrm{dom}(v)$ are regular for every q realized by a DMSOT or an LB transducer and every v realized by $(\mathrm{ELT}^{\mathrm{R}})^*$, hence $\mathrm{dom}(q) \cap \mathrm{dom}(v)$ is also effectively regular.

and *complete* algorithm exists for the existential preservation, and (2) our results can be extended to more expressive classes of tree transducers such as macro tree transducers (see, e.g., [32], [36], [37]). Obtaining a positive solution for the question (1) seems difficult, because one is required to prove a given query q is not preserved by v_i ($i \in [K]$) for *every* possible way of decomposing a finite-valued tree transduction v into single-valued ones $v_1, \ldots, v_K$.

As mentioned in Theorem 12, finite-valued bottom-up tree transducers can be effectively decomposed into a finite number of single-valued ones of *double*-exponential order of the size of the original transducers. Whereas, in the word case, k-valued (word) transducers can be effectively decomposed into k single-valued (unambiguous) ones [38], [39] of *single*-exponential size [40]. Can k-valued tree transducers decomposed into k single-valued ones of single-exponential size? It is an important problem that remains open for twenty years.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Numbers 26870270 and 23300008.

References

[1] A.Y. Halevy, "Answering queries using views: A survey," The VLDB Journal, vol.10, no.4, pp.270–294, 2001.

[2] A.Y. Halevy, A. Rajaraman, and J. Ordille, "Data integration: The teenage years," Proc. 32nd International Conference on Very Large Data Bases, pp.9–16, 2006.

[3] W. Fan and P. Bohannon, "Information preserving XML schema embedding," ACM Transactions on Database Systems, vol.33, no.1, pp.4:1–4:44, 2008.

[4] B. Groz, S. Staworko, A.-C. Caron, Y. Roos, and S. Tison, "Static analysis of XML security views and query rewriting," Information and Computation, vol.238, pp.2–29, 2014.

[5] L.V.S. Lakshmanan and A. Thomo, "View-based tree-language rewritings for XML," in Foundations of Information and Knowledge Systems, ed. C. Beierle and C. Meghini, LNCS 8367, pp.270–289, 2014.

[6] M. Bojańczyk, L.A. Kołodziejczyk, and F. Mopturlak, "Solutions in XML data exchange," Journal of Computer and System Sciences, vol.79, no.6, pp.785–815, 2013.

[7] K. Hashimoto, R. Sawada, Y. Ishihara, H. Seki, and T. Fujiwara, "Determinacy and subsumption for single-valued bottom-up tree transducers," in Language and Automata Theory and Applications, ed. A.H. Dediu, C. Martín-Vide, and B. Truthe, LNCS 7810, pp.335–346, 2013.

[8] K. Miyahara, K. Hashimoto, and H. Seki, "Node query preservation for deterministic linear top-down tree transducers," IEICE Trans. Inf. & Syst., vol.E98-D, no.3, pp.512–523, 2015.

[9] M. Arenas and L. Libkin, "XML data exchange: Consistency and query answering," Journal of the ACM, vol.55, no.2, pp.7:1–7:72, 2008.

[10] M. Benedikt, J. Engelfriet, and S. Maneth, "Determinacy and rewriting of top-down and MSO tree transformations," in Mathematical Foundations of Computer Science, ed. K. Chatterjee and J. Sgall, LNCS 8087, pp.146–158, 2013.

[11] N. Dalvi, C. Ré, and D. Suciu, "Probabilistic databases: Diamonds in the dirt," Communications of the ACM, vol.52, no.7, pp.86–94, 2009.

[12] K. Knight and J. Graehl, "An overview of probabilistic tree transducers for natural language processing," in Computational Linguistics and Intelligent Text Processing, ed. A. Gelbukh, LNCS 3406, pp.1–24, 2005.

[13] A. Maletti, J. Graehl, M. Hopkins, and K. Knight, "The power of extended top-down tree transducers," SIAM Journal on Computing, vol.39, no.2, pp.410–430, 2009.

[14] A. Maletti, "Survey: Weighted extended top-down tree transducers part II—application in machine translation," Fundamenta Informaticae, vol.112, no.2-3, pp.239–261, 2011.

[15] F. Braune, N. Seemann, D. Quernheim, and A. Maletti, "Shallow local multi bottom-up tree transducers in statistical machine translation," Proc. 51st Annual Meeting of the Association for Computational Linguistics, vol.1, pp.811–821, 2013.

[16] N. Seemann, F. Braune, and A. Maletti, "String-to-tree multi bottom-up tree transducers," Proc. 53rd Annual Meeting of the Association for Computational Linguistics, vol.1, pp.815–824, 2015.

[17] T. Brants, A.C. Popat, P. Xu, F.J. Och, and J. Dean, "Large language models in machine translation," Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp.858–867, 2007.

[18] A. Nash, L. Segoufin, and V. Vianu, "Views and queries: Determinacy and rewriting," ACM Transactions on Database Systems, vol.35, no.3, pp.21:1–21:41, 2010.

[19] E. Filiot, S. Maneth, P.-A. Reynier, and J.-M. Talbot, "Decision problems of tree transducers with origin," in Automata, Languages, and Programming, ed. M.M. Halldórsson, K. Iwama, N. Kobayashi, and B. Speckmann, LNCS 9135, pp.209–221, 2015.

[20] J. Engelfriet and S. Maneth, "The equivalence problem for deterministic MSO tree transducers is decidable," Information Processing Letters, vol.100, no.5, pp.206–212, 2006.

[21] H. Seidl, "Equivalence of finite-valued tree transducers is decidable," Mathematical Systems Theory, vol.27, no.4, pp.285–346, 1994.

[22] B. Courcelle, "Monadic second-order definable graph transductions: a survey," Theoretical Computer Science, vol.126, no.1, pp.53–75, 1994.

[23] B. Courcelle and J. Engelfriet, Graph Structure and Monadic Second-Order Logic, a Language-Theoretic Approach, Encyclopedia of Mathematics and its applications, vol.138, Cambridge University Press, 2012.

[24] R. Bloem and J. Engelfriet, "A comparison of tree transductions defined by monadic second order logic and by attribute grammars," Journal of Computer and System Sciences, vol.61, no.1, pp.1–50, 2000.

[25] J. Engelfriet, E. Lilin, and A. Maletti, "Extended multi bottom-up tree transducers," Acta Informatica, vol.46, no.8, pp.561–590, 2009.

[26] A. Maletti, "Compositions of extended top-down tree transducers," Information and Computation, vol.206, no.9-10, pp.1187–1196, 2008.

[27] A. Arnold and M. Dauchet, "Bi-transductions de forêts," Proc. 3rd Conference on Automata, Languages, and Programming, pp.74–86, 1976.

[28] Z. Fülöp, A. Maletti, and H. Vogler, "Weighted extended tree transducers," Fundamenta Informaticae, vol.111, no.2, pp.163–202, 2011.

[29] A.V. Aho and J.D. Ullman, "Translations on a context free grammar," Information and Control, vol.19, no.5, pp.439–475, 1971.

[30] J. Engelfriet, G. Rozenberg, and G. Slutzki, "Tree transducers, L systems, and two-way machines," Journal of Computer and System Sciences, vol.20, no.2, pp.150–202, 1980.

[31] T.V. Griffiths, "The unsolvability of the equivalence problem for Λ-free nondeterministic generalized machines," Journal of the ACM, vol.15, no.3, pp.409–413, 1968.

[32] J. Engelfriet and S. Maneth, "Macro tree transducers, attribute grammars, and MSO definable tree translations," Information and Com-

putation, vol.154, no.1, pp.34–91, 1999.

[33] J. Engelfriet, "Top-down tree transducers with regular look-ahead," Mathematical Systems Theory, vol.10, no.1, pp.289–303, 1976.

[34] J. Engelfriet, "Bottom-up and top-down tree transformations—a comparison," Mathematical Systems Theory, vol.9, no.2, pp.198–231, 1975.

[35] H. Seidl, "Single-valuedness of tree transducers is decidable in polynomial time," Theoretical Computer Science, vol.106, no.1, pp.135–181, 1992.

[36] J. Engelfriet and H. Vogler, "Macro tree transducers," Journal of Computer and System Sciences, vol.31, no.1, pp.71–146, 1985.

[37] J. Engelfriet and S. Maneth, "Macro tree translations of linear size increase are MSO definable," SIAM Journal on Computing, vol.32, no.4, pp.950–1006, 2003.

[38] M.P. Schützenberger, "Sur les relations rationnelles entre monoïdes libres," Theoretical Computer Science, vol.3, no.2, pp.243–259, 1976.

[39] A. Weber, "Decomposing a *k*-valued transducer into *k* unambiguous ones," Theoretical Informatics and Applications, vol.30, no.5, pp.379–413, 1996.

[40] J. Sakarovitch and R. de Souza, "Lexicographic decomposition of *k*-valued transducers," Theory of Computing Systems, vol.47, no.3, pp.758–785, 2010.

Kazuki Miyahara received the B.E. degree from Nagaoka University of Technology and obtained his M.E. in Information Science from Nara Institute of Science and Technology in 2011 and 2013, respectively. He is currently a doctoral student of Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include signal processing and formal language theory.

Kenji Hashimoto received the Ph.D. degree in Information and Computer Sciences from Osaka University in 2009. From 2009 to 2013, he was an Assistant Professor of Nara Institute of Science and Technology. Since Oct. 2013, he has been an Assistant Professor of Graduate School of Information Science, Nagoya University. His research interests include formal language, database theory, and information security.

Hiroyuki Seki received his Ph.D. degree from Osaka University in 1987. He was an Assistant Professor, and later, an Associate Professor from 1987 to 1994. In 1994, he joined Nara Institute of Science and Technology, where he was a Professor during 1996 to 2013. Currently, he is a Professor in Nagoya University. His current research interests include formal language theory and formal approach to software development.