Available online at www.sciencedirect.com
ScienceDirect
Information Processing Letters 100 (2006) 206–212
Information Processing Letters
www.elsevier.com/locate/ipl

The equivalence problem for deterministic MSO tree transducers is decidable

Joost Engelfriet [a], Sebastian Maneth [b,*,1]

[a] *LIACS, Leiden University, The Netherlands*
[b] *Faculté I & C, EPFL, Switzerland*

Received 19 January 2006
Available online 14 August 2006
Communicated by L. Boasson

Abstract

It is decidable for deterministic MSO definable graph-to-string or graph-to-tree transducers whether they are equivalent on a context-free set of graphs.

Keywords: Formal languages; Automata theory; Equivalence problem; Transducer

1. Introduction

It is well known that the equivalence problem for nondeterministic (one-way) finite state transducers is undecidable, even when they cannot read or write the empty string [19]. In contrast, equivalence *is* decidable for deterministic finite state transducers, even for two-way transducers [20]. The question arises whether these results can be generalized from strings to transducers working on more complex structures like, e.g., trees or graphs. There is no accepted notion of finite state transducer working on graphs; instead, it is believed that transductions expressed in monadic second-order logic (MSO) are the natural counterpart of finite state transductions on graphs. The idea is to define an output graph by interpreting fixed MSO formulas on a given input graph. In fact, if the input and output graphs of such an MSO graph transducer are strings, then the resulting transductions (in the deterministic case) are precisely the deterministic two-way finite state transductions [9]. Hence, by the above, equivalence is decidable for deterministic MSO string transducers. A nondeterministic MSO graph transducer can easily simulate a nondeterministic finite state transducer that cannot read the empty string; hence, equivalence is undecidable. Actually, even for deterministic MSO graph transducers equivalence is undecidable. This is due to the fact that MSO is undecidable for graphs (Propositions 5.2.1 and 5.2.2 of [4]); cf. end of Section 3 for more details. The question remains whether deterministic MSO tree transducers have a decidable equivalence problem. Recently, these transducers have been characterized by cer-

* Corresponding author.
E-mail addresses: engelfri@liacs.nl (J. Engelfriet), sebastian.maneth@epfl.ch, sebastian.maneth@nicta.com.au (S. Maneth).
[1] Present address: National ICT Australia Ltd, Sydney, Australia and School of Computer Science and Engineering, University of New South Wales.

0020-0190/$ – see front matter
doi:10.1016/j.ipl.2006.05.015

tain attribute grammars [2] and macro tree transducers [10]. However, for both models it is unknown whether equivalence is decidable. Here we give an affirmative answer: equivalence of deterministic MSO tree transducers is decidable. This result has several applications; for instance, it implies that XML queries of linear size increase have decidable equivalence, by the results of [24,11,12,22], and [23]. Our proof generalizes the one of [20] (see also [21]): it is based on the fact that certain sets are semilinear. It proceeds roughly as follows: two transducers M_1, M_2 are equivalent if there is no input s and position n such that the symbol at position n of M_1's output on s is different from the symbol at position n of M_2's output on s. Hence, we must test whether there exists an n and distinct symbols a, b such that (n, n) is contained in the set $S^{a,b}$ of all pairs (i, j) where M_1's output at position i is a and M_2's output at position j is b, for some input s. The set $S^{a,b}$ is semilinear which implies that the existence of such an n is decidable. Semilinearity of $S^{a,b}$ is proved using known results from the theory of MSO graph transducers, by coding the pair (i, j) as a discrete graph with i a-labeled nodes and j b-labeled nodes.

This paper was presented at the 25th FSTTCS conference in Hyderabad, India [13].

2. Preliminaries

The reader is assumed to be familiar with MSO on graphs and with MSO graph transducers, see, e.g., the survey papers [4,3].

A graph alphabet is a pair (Σ, Γ) of alphabets of node and edge labels, respectively. A graph over (Σ, Γ) is a tuple (V, E, λ) where V is the finite set of nodes, $E \subseteq V \times \Gamma \times V$ is the set of edges, and $\lambda : V \to \Sigma$ is the node labeling function. The set of all graphs over (Σ, Γ) is denoted $\mathrm{GR}(\Sigma, \Gamma)$. The language $\mathrm{MSO}(\Sigma, \Gamma)$ of monadic second-order (MSO) formulas over (Σ, Γ) uses node variables $x, y, \ldots$ and node-set variables $X, Y, \ldots$; both can be quantified with $\exists$ and $\forall$. It has atomic formulas $\mathrm{lab}_\sigma(x)$ for $\sigma \in \Sigma$, denoting that x is labeled σ, $\mathrm{edg}_\gamma(x, y)$ for $\gamma \in \Gamma$, denoting that there is a γ-labeled edge from x to y, and $x \in X$ denoting that x is in X. For $g \in \mathrm{GR}(\Sigma, \Gamma)$ and a closed formula ψ in $\mathrm{MSO}(\Sigma, \Gamma)$ we write $g \models \psi$ if g satisfies ψ; similarly, if ψ has free variables x or x, y and u, v are nodes of g, then we write $(g, u) \models \psi$ or $(g, u, v) \models \psi$ if g satisfies ψ with $x = u$ or with $x = u$, $y = v$, respectively.

Let (Σ_1, Γ_1), (Σ_2, Γ_2) be graph alphabets. A *deterministic MSO graph transducer M (from (Σ_1, Γ_1) to (Σ_2, Γ_2))* is a tuple $(C, \varphi_{\mathrm{dom}}, \Psi, X)$ where C is a finite set of *copy names*, $\varphi_{\mathrm{dom}} \in \mathrm{MSO}(\Sigma_1, \Gamma_1)$ is the closed *domain formula*, $\Psi = \{\psi_{c,\sigma}(x)\}_{c \in C, \sigma \in \Sigma_2}$ is a family of *node formulas*, i.e., MSO formulas $\psi_{c,\sigma}(x)$ over (Σ_1, Γ_1) with one free variable x, and $X = \{\chi_{c,c',\gamma}(x, y)\}_{c,c' \in C, \gamma \in \Gamma_2}$ is a family of *edge formulas*, i.e., MSO formulas $\chi_{c,c',\gamma}(x, y)$ over (Σ_1, Γ_1) with two free variables x, y.

The purpose of the node formulas of an MSO graph transducer is twofold: (1) they define which (copies of) nodes of the input graph are used, and they define their labels. As can be seen easily, it is no loss of generality to require that, for every $c \in C$, the formulas $\psi_{c,\sigma}(x)$ are mutually exclusive.

Given $g \in \mathrm{GR}(\Sigma_1, \Gamma_1)$, the graph $h = \tau_M(g) \in \mathrm{GR}(\Sigma_2, \Gamma_2)$ is defined if $g \models \varphi_{\mathrm{dom}}$, and then $V_h = \{(c, u) \mid c \in C, u \in V_g$, there is exactly one $\sigma \in \Sigma_2$ such that $(g, u) \models \psi_{c,\sigma}(x)\}$, $E_h = \{((c, u), \gamma, (c', u')) \mid (c, u), (c', u') \in V_h, \gamma \in \Gamma_2$, and $(g, u, u') \models \chi_{c,c',\gamma}(x, y)\}$, and $\lambda_h = \{((c, u), \sigma) \mid (c, u) \in V_h, \sigma \in \Sigma_2$, and $(g, u) \models \psi_{c,\sigma}(x)\}$. Hence, τ_M is a partial function from $\mathrm{GR}(\Sigma_1, \Gamma_1)$ to $\mathrm{GR}(\Sigma_2, \Gamma_2)$ with $\mathrm{dom}(\tau_M) = \{g \in \mathrm{GR}(\Sigma_1, \Gamma_1) \mid g \models \varphi_{\mathrm{dom}}\}$.

In the sequel we often identify a transducer M with its transduction τ_M, and simply write, e.g., $M(g)$ in place of $\tau_M(g)$.

A (*nondeterministic*) *MSO graph transducer* is obtained from a deterministic one by allowing all formulas to use fixed free node-set variables $Y_1, Y_2, \ldots$, called parameters. For each valuation of the parameters (by sets of nodes of the input graph) that satisfies the domain formula, the other formulas define the output graph as before. Hence each such valuation may lead to a different output graph for the given input graph. Thus, $\tau_M \subseteq \mathrm{GR}(\Sigma_1, \Gamma_1) \times \mathrm{GR}(\Sigma_2, \Gamma_2)$.

For an alphabet Δ and $a_1, \ldots, a_n \in \Delta$, $n \geqslant 0$, we identify the string $w = a_1 a_2 \ldots a_n$ with the graph in $\mathrm{GR}(\{\#\}, \Delta)$ that has #-labeled nodes $v_1, \ldots, v_{n+1}$ and, for $1 \leqslant i \leqslant n$, an a_i-labeled edge from v_i to v_{i+1}. For $1 \leqslant i \leqslant n$, we denote by w/i the ith letter a_i of w.

Let Σ be a ranked alphabet, i.e., an alphabet Σ together with a mapping $\mathrm{rank}_\Sigma : \Sigma \to \mathbb{N}$. Let m be the maximal rank of symbols in Σ. A *tree* (*over* Σ) is an acyclic, connected graph in $\mathrm{GR}(\Sigma, \{1, \ldots, m\})$, with exactly one node that has no incoming edges (the root), and, for $\sigma \in \Sigma$, every σ-labeled node has exactly $\mathrm{rank}_\Sigma(\sigma)$ outgoing edges, labeled $1, 2, \ldots, \mathrm{rank}_\Sigma(\sigma)$, respectively. The set of all trees over Σ is denoted by T_Σ.

Let M be an MSO graph transducer and let X, Y be sets of graphs. Then M is called an MSO X-to-Y transducer, if $\mathrm{dom}(M) \subseteq X$ and $\mathrm{range}(M) \subseteq Y$, and it is an MSO X transducer if additionally $Y = X$. Thus, as

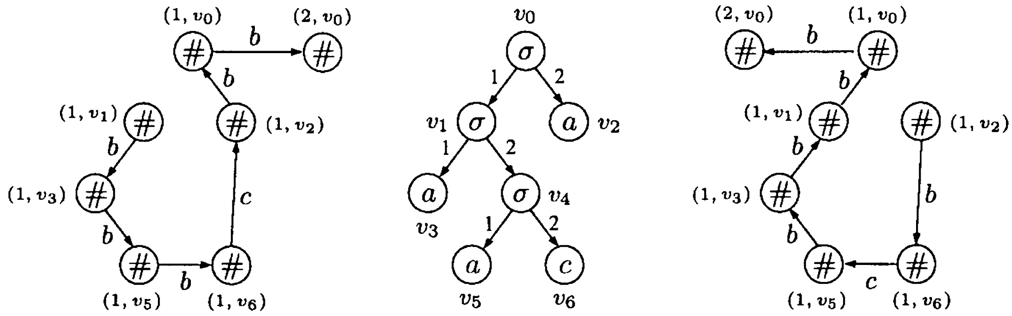

Fig. 1. The tree $s = \sigma(\sigma(a, \sigma(a, c), a))$ in the center, $M_3(s) = bbbcbb$ on the left, and $M_4(s) = bcbbbb$ on the right.

an example, an MSO tree-to-string transducer translates trees into strings.

Convention. All lemmas stated in this paper are *effective*.

Example 1. (i) Let Σ be the ranked alphabet consisting of the binary symbol σ and the nullary symbol a. Consider the deterministic MSO tree-to-string transducers M_1, M_2 that translate trees s over Σ into strings b^n, where $n = i + j + k - 1$, i is the number of binary nodes on the left-most path in s, j is the number of leaves in s, and k is the number of binary nodes on the right-most path in s. Let us denote n by outer(s). Roughly, the transducer M_1 realizes the translation by doing a depth-first left-to-right traversal through s, while M_2 does a depth-first right-to-left traversal. Let $M_1 = (\{1,2\}, \varphi_{\text{tree}}, \{\psi_{1,\#}(x), \psi_{2,\#}(x)\}, \{\chi_{c,c',b}(x,y)\}_{c,c'\in\{1,2\}})$ where φ_{tree} is an MSO formula expressing that a graph is a tree, and

$$\begin{aligned}
\psi_{1,\#}(x) &\equiv \text{lab}_a(x) \vee (\exists y)\big(\text{root}(y) \\
&\qquad \wedge \big(\text{edg}_1^*(y,x) \vee \text{edg}_2^*(y,x)\big)\big),\\
\psi_{2,\#}(x) &\equiv \text{root}(x),\\
\chi_{1,1,b}(x,y) &\equiv \neg\,\text{root}(x) \wedge \big(\text{edg}_1(x,y) \\
&\qquad \vee \text{yield}(x,y) \vee \text{edg}_2(y,x)\big),\\
\chi_{1,2,b}(x,y) &\equiv \text{root}(x),\\
\chi_{2,1,b}(x,y) &\equiv \chi_{2,2,b}(x,y) \equiv \text{false}.
\end{aligned}$$

Here, root(y) denotes that y is the root node, yield(x, y) denotes that x is a leaf and y is the next leaf in pre-order, and $\text{edg}_1^*(x,y)$ denotes the transitive closure of edg_1 (similarly edg_2^*); all these can easily be expressed in MSO. The transducer M_2 is defined as M_1 with the only difference that $\chi_{1,1,b}(x,y) \equiv \neg\,\text{root}(x) \wedge (\text{edg}_2(x,y) \vee \text{yield}(x,y) \vee \text{edg}_1(y,x))$.

(ii) Next, consider the transducers M_3, M_4, obtained from M_1, M_2 as follows. We now take input trees over Σ' that additionally contains the nullary symbol c which is translated into c. Hence, M_3 becomes

$$\begin{aligned}
\psi_{1,\#}(x) &\equiv \text{lab}_a(x) \vee \text{lab}_c(x) \vee (\exists y)\big(\text{root}(y) \\
&\qquad \wedge \big(\text{edg}_1^*(y,x) \vee \text{edg}_2^*(y,x)\big)\big),\\
\psi_{2,\#}(x) &\equiv \text{root}(x),\\
\chi_{1,1,b}(x,y) &\equiv \neg\,\text{root}(x) \wedge \neg\,\text{lab}_c(x) \wedge \big(\text{edg}_1(x,y) \\
&\qquad \vee \text{yield}(x,y) \vee \text{edg}_2(y,x)\big),\\
\chi_{1,1,c}(x,y) &\equiv \neg\,\text{root}(x) \wedge \text{lab}_c(x) \wedge \big(\text{edg}_1(x,y) \\
&\qquad \vee \text{yield}(x,y) \vee \text{edg}_2(y,x)\big),\\
\chi_{1,2,b}(x,y) &\equiv \text{root}(x) \wedge \neg\,\text{lab}_c(x),\\
\chi_{1,2,c}(x,y) &\equiv \text{root}(x) \wedge \text{lab}_c(x),\\
\chi_{2,1,b}(x,y) &\equiv \chi_{2,2,b}(x,y) \equiv \chi_{2,1,c}(x,y)\\
&\equiv \chi_{2,2,c}(x,y) \equiv \text{false}.
\end{aligned}$$

Consider the input tree $s = \sigma(\sigma(a, \sigma(a, c), a))$. Then, as can be seen in Fig. 1, M_3 generates as output the string $M_3(s) = bbbcbb$ and M_4 generates $M_4(s) = bcbbbb$.

The following lemma contains a basic fact about MSO definable graph transductions; see, e.g., Proposition 3.2 in [3].

Lemma 2. *The* (*deterministic*) *MSO graph transductions are closed under composition.*

Notation. Let $M_1; M_2$ denote a transducer M for which $\tau_M = \tau_{M_2} \circ \tau_{M_1}$; note that M is deterministic, if M_1 and M_2 are. By Lemma 2, $M_1; M_2$ effectively exists.

A *discrete graph* (*dgraph*, for short) is a graph without edges. Let g be a dgraph over $(\Sigma, \emptyset)$ with $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$. Define Par($g$) as the vector $(n_1, \ldots, n_k)$ in $\mathbb{N}^k$ such that, for $1 \leqslant i \leqslant k$, n_i is the number of σ_i-labeled nodes in g. Similarly, for a string $w \in \Sigma^*$, Par(w) is the vector in $\mathbb{N}^k$ such that the ith component is the number of σ_i's in w. We denote by $\mathrm{dgr}(w)$ the (unique) dgraph g such that $\mathrm{Par}(g) = \mathrm{Par}(w)$. For a set S of dgraphs or strings, $\mathrm{Par}(S)$ is the set of all $\mathrm{Par}(g)$ for $g \in S$. A set $P \subseteq \mathbb{N}^k$ is *semilinear* if there exists a regular language R such that $P = \mathrm{Par}(R)$, cf. [25,18,17] for the more usual definition of semilinearity. The set S is *Parikh* if $\mathrm{Par}(S)$ is semilinear. Note that since $\mathrm{Par}(R) = \emptyset$ iff $R = \emptyset$, emptiness of semilinear sets is decidable.

A set of graphs is NR if it is generated by a context-free node replacement graph grammar, see, e.g., [8,3]; it is also called C-edNCE or VR. Such grammars have productions of the form $X \to (g, C)$ where X is a nonterminal, g is a graph, and C is a finite set of connection instructions. The application of $X \to (g, C)$ to an X-labeled node v of a sentential form works as follows. First v (and edges to and from it) is removed, g is disjointly added, and then edges from the former neighbors of v to nodes in g are established, according to the connection instructions. A connection instruction is of the form $(\sigma, \beta, \gamma, x, d)$ where σ is a node label, β, γ are edge labels, x is a node of g, and $d \in \{\mathrm{in}, \mathrm{out}\}$. For $d = \mathrm{out}$ it means that if there was a β-labeled edge from v to a σ-labeled neighbor w of v, then a γ-labeled edge from x to w is generated. Similarly, for "in", edges from a w to x are generated. The derivation relation of the grammar should satisfy a certain confluence requirement, see Definition 4.6 of [8].

Lemma 3. (Theorem 7.1 of [3].) *The images of NR sets of graphs under MSO graph-to-dgraph transductions are Parikh.*

In fact, the class of NR sets of graphs is closed under MSO graph transductions (see Theorem 4.2(3) of [3], or Section 5 of [8]) and NR sets of graphs are Parikh (see Proposition 4.11 of [8]).

A useful property of semilinear sets is their (effective) closure under intersection. It implies the following lemma.

Lemma 4. *It is decidable for a semilinear set $S \subseteq \mathbb{N}^2$ whether there exists an $n \in \mathbb{N}$ such that $(n, n) \in S$.*

Proof. Let $P = \{(n, n) \mid n \in \mathbb{N}\} = \mathrm{Par}((ab)^*)$. The lemma holds because $S \cap P$ is semilinear [18,17] and semilinear sets have a decidable emptiness problem. □

Note that Lemma 4 can alternatively be proved without using the fact that semilinear sets are closed under intersection: let R be a regular language with $\mathrm{Par}(R) = S$ and let C be the context-free language containing all strings over $\{a, b\}$ that have an equal number of a's and b's. Then $C' = R \cap C$ is context-free and hence $\mathrm{Par}(C')$ is semilinear by Parikh's theorem [25].

3. Main result

The main result of this paper (Theorem 8) is that it is decidable for deterministic MSO graph-to-string or graph-to-tree transducers whether they are equivalent on an NR set of graphs, cf. the Abstract.

Recall from the Preliminaries that for a string w, w/i denotes the ith letter of w.

Lemma 5. *Let Δ be an alphabet and $a \in \Delta$. There exists an MSO string-to-dgraph transducer N^a_Δ such that for every $w \in \Delta^*$,*

$$N^a_\Delta(w) = \{\mathrm{dgr}(a^n) \mid w/n = a\}.$$

Proof. The transducer N^a_Δ uses one parameter Y_1 to nondeterministically choose a node v that has an outgoing a-labeled edge (if there is one). It copies v and all input nodes to the left of v, and labels them a. There are no edge formulas because dgraphs have no edges. Define $N^a_\Delta = (\{1\}, \varphi_{\mathrm{dom}}(Y_1), \psi_{1,a}(x, Y_1), \emptyset)$ with

$$\varphi_{\mathrm{dom}}(Y_1) \equiv \varphi_{\mathrm{string}} \wedge \mathrm{singleton}(Y_1) \wedge (\exists x)(\exists y)\big(\mathrm{edg}_a(x, y) \wedge x \in Y_1\big),$$

$$\psi_{1,a}(x, Y_1) \equiv (\exists y)(x \preccurlyeq y \wedge y \in Y_1)$$

where $\varphi_{\mathrm{string}}$ expresses that a graph is a string, $\mathrm{singleton}(Y_1)$ expresses that Y_1 is a singleton, and $x \preccurlyeq y$ that there is a path from x to y. □

We denote the disjoint union of graphs h_1 and h_2 by $h_1 \uplus h_2$.

Lemma 6. *Let M_1, M_2 be MSO graph transducers. There exists an MSO graph transducer M, denoted $M_1 \uplus M_2$, such that for every graph g,*

$$M(g) = \{h_1 \uplus h_2 \mid h_1 \in M_1(g),\ h_2 \in M_2(g)\}.$$

Proof. Let $M_1 = (C_1, \varphi_1, \Psi_1, X_1)$ and $M_2 = (C_2, \varphi_2, \Psi_2, X_2)$. We may assume w.l.o.g. that C_1 is disjoint from C_2 and that the parameters of M_1 are disjoint from those of M_2. Then $M = (C_1 \cup C_2, \varphi_1 \wedge \varphi_2, \Psi_1 \cup \Psi_2, X_1 \cup X_2 \cup X)$ realizes the desired transduction, where all edge formulas in X are set to false, i.e., $\chi_{c,c',\gamma}(x, y) \equiv \mathrm{false}$ for all $(c, c') \in (C_1 \times C_2) \cup (C_2 \times C_1)$. □

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

$$M^{a,b}(g) = \{\mathrm{dgr}(a^m b^n) \mid \exists h_1 \in M_1(g),\ h_2 \in M_2(g):\ h_1/m = a \text{ and } h_2/n = b\}.$$

Proof. Let M_i be from (Σ_i, Γ_i) to $(\{\#\}, \Delta_i)$ for $i \in \{1,2\}$. If $a \notin \Delta_1$ or $b \notin \Delta_2$ then let $M^{a,b} = (\emptyset, \text{false}, \emptyset, \emptyset)$. Otherwise define $M^{a,b} = (M_1; N^a_{\Delta_1}) \uplus (M_2; N^b_{\Delta_2})$ according to Lemmas 2, 5, and 6. □

For a relation $R \subseteq A \times B$ and a set $D \subseteq A$, denote by $R|_D$ the restriction of R to D, i.e., $R|_D = \{(a,b) \in R \mid a \in D\}$.

Theorem 8. *It is decidable for deterministic MSO graph-to-string or graph-to-tree transducers M_1, M_2 and an NR set D of graphs whether $\tau_{M_1}|_D = \tau_{M_2}|_D$.*

Proof. We start with the graph-to-string case. For $i \in \{1,2\}$ let $D_i = \mathrm{dom}(M_i) \cap D$. We first show that it is decidable whether $D_1 = D_2$. Clearly, $D_1 = D_2$ if and only if $\mathrm{Par}(E(D)) = \emptyset$, where E is the deterministic MSO graph-to-dgraph transducer that removes the edges of all graphs in the symmetric difference of $\mathrm{dom}(M_1)$ and $\mathrm{dom}(M_2)$: $E = (\{1\}, \neg(\varphi_1 \leftrightarrow \varphi_2), \{\psi_{1,\sigma}(x)\}_{\sigma\in\Sigma}, \emptyset)$ where φ_i is the domain formula of M_i for $i \in \{1,2\}$, Σ is the node alphabet of D, and $\psi_{1,\sigma}(x) = \mathrm{lab}_\sigma(x)$ for $\sigma \in \Sigma$. By Lemma 3, $\mathrm{Par}(E(D))$ is effectively semilinear, and hence its emptiness can be decided. If $D_1 \neq D_2$ then we are finished and know that $\tau_{M_1}|_D \neq \tau_{M_2}|_D$. Assume now that $D_1 = D_2$.

Let M_i have output edge alphabet Δ_i, for $i \in \{1,2\}$, and let \$ be a symbol not in $\Delta = \Delta_1 \cup \Delta_2$. We define deterministic MSO graph-to-string transducers $M_i^\$ = M_i; N$ such that $M_i^\$(g) = M_i(g)\$$ for all $g \in \mathrm{dom}(M_i)$. Here N is the deterministic MSO string transducer $(C, \text{true}, \{\psi_{1,\#}(x), \psi_{2,\#}(x)\}, \{\chi_{c,c',\delta}(x,y)\}_{c,c'\in C,\, \delta\in\Delta\cup\{\$\}})$ such that

$$C = \{1,2\},\ \psi_{1,\#}(x) \equiv \text{true},$$

$$\psi_{2,\#}(x) \equiv \chi_{1,2,\$}(x,y) \equiv \neg(\exists z) \bigvee_{\delta\in\Delta} \mathrm{edg}_\delta(x,z)$$

and, for $\delta \in \Delta$, $\chi_{1,1,\delta}(x,y) = \mathrm{edg}_\delta(x,y)$; all other edge formulas are set to false.

Since now all output strings end on the special marker \$, $\tau_{M_1}|_D \neq \tau_{M_2}|_D$ iff

$$\exists a\ \exists b\colon \big(d(a,b) \wedge \exists n\ \exists g\colon \big(g \in D_1 \wedge M_1^\$(g)/n = a \wedge M_2^\$(g)/n = b\big)\big)$$

where $d(a,b)$ denotes the statement $a, b \in (\Delta \cup \{\$\}) \wedge a \neq b$. For given a, b, let $M^{a,b}$ be the transducer of Lemma 7 for a, b, $M_1^\$$, $M_2^\$$. Then the statement displayed above holds if and only if

$$\exists a\ \exists b\colon \big(d(a,b) \wedge \exists n\colon \mathrm{dgr}(a^n b^n) \in M^{a,b}(D)\big)$$
$$\text{iff } \exists a\ \exists b\colon \big(d(a,b) \wedge \underbrace{\exists n\colon (n,n) \in \mathrm{Par}\big(M^{a,b}(D)\big)}_{P(a,b)}\big)$$

By Lemma 3, $\mathrm{Par}(M^{a,b}(D))$ is effectively semilinear. By Lemma 4 this means that $P(a,b)$ is decidable. Since there are only finitely many a, b with $d(a,b)$, the statement is decidable.

We now reduce the graph-to-tree case to the graph-to-string case. Let Δ be a ranked alphabet and let m be the maximal rank of its elements. There is a deterministic MSO tree-to-string transducer M_Δ that translates every tree t over Δ into the string $\mathrm{pre}(t)$ of its node labels in pre-order. Clearly, if we associate with a deterministic MSO graph-to-tree transducer M (from (Σ, Γ) to $(\Delta, \{1,\dots,m\})$) the deterministic MSO graph-to-string transducer $\widehat{M} = M; M_\Delta$, then M_1 is equivalent to M_2 on D if and only if $\widehat{M}_1$ is equivalent to $\widehat{M}_2$ on D. Let $M_\Delta = (\{1,2\}, \text{true}, \{\psi_{1,\#}(x), \psi_{2,\#}(x)\}, \{\chi_{c,c',\delta}(x,y)\}_{c,c'\in\{1,2\},\delta\in\Delta})$ with $\psi_{1,\#}(x) \equiv \text{true}$, $\psi_{2,\#}(x) \equiv \mathrm{root}(x)$, where $\mathrm{root}(x)$ expresses that x is the root node. Further, for $\delta \in \Delta$, $\chi_{1,1,\delta}(x,y) \equiv \mathrm{lab}_\delta(x) \wedge \pi(x,y)$ and $\chi_{1,2,\delta}(x,y) \equiv \mathrm{lab}_\delta(x) \wedge \mathrm{root}(y) \wedge \neg(\exists z)\pi(x,z)$ where $\pi(x,y)$ expresses that y is the successor of x in the pre-order. □

Note that it is essential in Theorem 8 that the transductions are restricted to an NR set of graphs. The set of all graphs (over a given alphabet) is *not* NR. In fact, equivalence of deterministic MSO graph (or graph-to-string, or graph-to-tree) transducers is undecidable when taking all graphs as input. This follows from the fact that, as mentioned in the Introduction, MSO is undecidable for graphs (Propositions 5.2.1 and 5.2.2 of [4]); i.e., given an MSO formula ϕ, it is undecidable whether ϕ holds for all graphs. It is easy to construct MSO graph-to-string transducers M_1, M_2 that take as input the disjoint union h of an arbitrary graph g and the string \$, such that M_1 translates h into \$ if g satisfies ϕ (and is undefined otherwise), and M_2 translates h into \$. Clearly, M_1 is equivalent to M_2 if and only if ϕ holds for all graphs g. The proof for the graph-to-tree case is analogous.

Example 9. (i) Let M_1, M_2 (and Σ, Σ') be as in Example 1(i). We now want to follow the proof of Theorem 8 to test whether M_1 is equivalent to M_2. We take $D =$

T_Σ. First, construct $M_1^\$$ and $M_2^\$$. Then construct $M^{b,\$}$ and $M^{\$,b}$. Clearly, for every tree s over Σ, $M^{b,\$}(s) = (\mathrm{dgr}(b^m\$^n) \mid 1 \leqslant m \leqslant \mathrm{outer}(s),\ n = \mathrm{outer}(s)+1\}$ and $M^{\$,b}(s) = \{\mathrm{dgr}(\$^m b^n) \mid m = \mathrm{outer}(s)+1,\ 1 \leqslant n \leqslant \mathrm{outer}(s)\}$. The sets $M^{b,\$}(D)$ and $M^{\$,b}(D)$ are Parikh. Clearly, $\mathrm{Par}(M^{b,\$}(T_\Sigma)) = \{(m,n) \mid 1 \leqslant m < n,\ n = 2$ or $n = 4$ or $n \geqslant 6\}$ which implies that its intersection with $P = \{(n,n) \mid n \in \mathbb{N}\}$ is empty. Similarly, $\mathrm{Par}(M^{b,\$}(T_\Sigma)) \cap P = \emptyset$, which proves that indeed M_1 is equivalent to M_2.

(ii) Consider now the transducers M_3, M_4 of Example 1(ii). If we again follow the proof of Theorem 8, then it turns out that the set $\mathrm{Par}(M^{b,c}(T_{\Sigma'})$ contains, e.g., the pair $(2,2)$ and hence M_3 is *not* equivalent to M_4. To see this, consider the input tree $s = \sigma(\sigma(a, \sigma(a,c), a)))$. Then $M_3(s) = bbbcbb$ and $M_4(s) = bcbbbb$, and therefore $\mathrm{dgr}(bbcc) \in M^{b,c}(s)$.

4. String and tree transductions

Clearly, Theorem 8 also holds if we restrict the input graphs to strings or trees. In particular, deterministic MSO X-to-Y transducers have decidable equivalence for all $X, Y \in \{\text{string}, \text{tree}\}$, because the set of all strings and the set of all trees (over given alphabets) are NR. For string transducers this reproves the decidability result of [20] (through [9]). For trees we obtain the following new decidability result (see the Title of this paper).

Corollary 10. *The equivalence problem is decidable for deterministic MSO tree transducers.*

Of course, even stronger statements hold; namely, given an NR set D of strings or trees, it is decidable if two deterministic MSO X-to-Y transducers are equivalent when restricted to D. For string transducers this means the following.

Corollary 11. *It is decidable whether two deterministic two-way finite state transducers are equivalent on an NR set of strings.*

As discussed in Section 6 of [8], the NR sets of strings are the same as the ranges of deterministic tree-walking tree-to-string transducers. They properly contain, for instance, the context-free languages and the ranges of deterministic two-way finite state transducers. Since the NR sets of strings form a full AFL of Parikh languages, Corollary 11 is in fact a special case of the general decidability result for deterministic two-way finite state transducers in Theorem 5 of [21]. It is incomparable to the decidability of equivalence of two such transducers on an NPDT0L language [6].

Our results also imply that equivalence of deterministic tree-walking tree-to-string transducers [1] is decidable: by Lemma 4.4 and Theorem 4.7 of [14] such transducers can be simulated by deterministic finite-copying top-down tree-to-string transducers with regular look-ahead, which, in turn, realize exactly the same translations as deterministic MSO tree-to-string transducers by Theorem 7.7 of [10].

Corollary 12. *It is decidable whether two deterministic tree-walking tree-to-string transducers are equivalent.*

The two statements of the next corollary follow from the characterizations of deterministic MSO tree transductions in [2] and [12], respectively. An attributed tree transducer is essentially an attribute grammar which translates trees over a ranked alphabet (instead of derivation trees of a context-free grammar) into trees; the only allowed semantic operation is top-concatenation of trees. An attribute grammar (or attributed tree transducer) is said to be single-use restricted if each attribute is used at most once in every local set of semantic rules. The single-use restriction implies that the corresponding dependency graphs are trees (instead of DAGs, for unrestricted grammars/transducers). A tree transducer is of linear size increase if the size of the output tree is at most linear in the size of the input tree; it is decidable whether a deterministic macro tree transducer is of linear size increase [12]. For more information on tree transducers see [16].

Corollary 13. *The equivalence problem is decidable*

(1) *for single-use restricted attributed tree transducers and*
(2) *for deterministic macro tree transducers of linear size increase.*

This result is incomparable with the decidability of the equivalence problem for

(1) deterministic bottom-up tree transducers [26],
(2) deterministic top-down tree transducers [15], and
(3) deterministic nonnested separated attributed/macro tree transducers [5].

Note that (2) is included in the more general (3), and that all transducers mentioned in (1)–(3) can be simulated by deterministic macro tree transducers. It remains open since 1979 [7] whether the equivalence problem is decidable for attributed tree transducers and for deterministic macro tree transducers.

In both [24] and [23] formal models of XML queries are introduced (the pebble tree transducer and the transformation language TL, respectively). In the deterministic case these can be simulated by compositions of deterministic macro tree transducers [11,23]. Hence we call such compositions *deterministic XML queries*. If they are of linear size increase, then they are MSO definable [22].

Corollary 14. *The equivalence problem is decidable for deterministic XML queries of linear size increase.*

References

[1] A.V. Aho, J.D. Ullman, Translations on a context-free grammar, Inform. and Control 19 (1971) 439–475.

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

[3] B. Courcelle, Monadic second-order definable graph transductions: a survey, Theoret. Comput. Sci. 126 (1994) 53–75.

[4] B. Courcelle, The expression of graph properties and graph transformations in monadic second-order logic, in: G. Rozenberg (Ed.), Handbook of Graph Grammars and Computing by Graph Transformation, vol. 1, World Scientific, 1997, pp. 313–400.

[5] B. Courcelle, P. Franchi-Zannettacci, On the equivalence problem for attribute systems, Inform. and Control 52 (1982) 275–305.

[6] K. Culik II, J. Karhumäki, The equivalence problem for single-valued two-way transducers (on NPDT0L languages) is decidable, SIAM J. Comput. 16 (1987) 221–230.

[7] J. Engelfriet, Some open questions and recent results on tree transducers and tree languages, in: Formal Language Theory; Perspectives and Open Problems, Academic Press, 1980.

[8] J. Engelfriet, Context-free graph grammars, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, vol. 3, Springer-Verlag, 1997, pp. 125–213.

[9] J. Engelfriet, H.J. Hoogeboom, MSO definable string transductions and two-way finite-state transducers, ACM Trans. Comput. Logic 2 (2001) 216–254.

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

[11] J. Engelfriet, S. Maneth, A comparison of pebble tree transducers with macro tree transducers, Acta Informatica 39 (2003) 613–698.

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

[13] J. Engelfriet, S. Maneth, The equivalence problem for deterministic MSO tree transducers is decidable, in: Proc. FSTTCS 2005, in: Lecture Notes in Comput. Sci., vol. 3821, Springer-Verlag, 2005, pp. 495–504.

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

[15] Z. Ésik, Decidability results concerning tree transducers II, Acta Cybernetica (1980).

[16] Z. Fülöp, H. Vogler, Syntax-Directed Semantics—Formal Models Based on Tree Transducers, W. Brauer, G. Rozenberg, A. Salomaa (Eds.), EATCS Monographs in Theoretical Computer Science, Springer-Verlag, 1998.

[17] S. Ginsburg, The Mathematical Theory of Context-Free Languages, McGraw-Hill, 1966.

[18] S. Ginsburg, E.H. Spanier, Bounded Algol-like languages, Trans. Amer. Math. Soc. 113 (1964) 333–368.

[19] T.V. Griffiths, The unsolvability of the equivalence problem for Λ-free nondeterministic generalized machines, J. ACM 15 (1968) 409–413.

[20] E.M. Gurari, The equivalence problem for deterministic two-way sequential transducers is decidable, SIAM J. Comput. 11 (1982) 448–452.

[21] O.H. Ibarra, 2DST mappings on languages and related problems, Theoret. Comput. Sci. 19 (1982) 219–227.

[22] S. Maneth, The macro tree transducer hierarchy collapses for functions of linear size increase, in: Proc. FSTTCS 2003, in: Lecture Notes in Comput. Sci., vol. 2556, Springer-Verlag, 2003, pp. 326–337.

[23] S. Maneth, A. Berlea, T. Perst, H. Seidl, XML type checking with macro tree transducers, in: Proc. PODS 2005, ACM Press, 2005, pp. 283–294.

[24] T. Milo, D. Suciu, V. Vianu, Typechecking for XML transformers, J. Comput. Syst. Sci. 66 (2003) 66–97.

[25] R. Parikh, On context-free languages, J. ACM 13 (1966) 570–581.

[26] Z. Zachar, The solvability of the equivalence problem for deterministic frontier-to-root tree transducers, Acta Cybernetica 4 (1980) 167–177.