INFORMATIQUE THÉORIQUE ET APPLICATIONS

YVES ANDRÉ
MAX DAUCHET

Decidability of equivalence for a class of non-deterministic tree transducers

Informatique théorique et applications, tome 28, nº 5 (1994), p. 447-463.
<http://www.numdam.org/item?id=ITA_1994__28_5_447_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 28, n° 5, 1994, p. 447 à 463)

DECIDABILITY OF EQUIVALENCE FOR A CLASS OF NON-DETERMINISTIC TREE TRANSDUCERS (*)

by Yves ANDRÉ (1) and Max DAUCHET (1)

Communicated by A. ARNOLD

Abstract. – *In this paper, we consider non-deterministic tree transducers in the letter to letter case, that is to say tree transducers for which trees which appear in the rules are reduced to one letter in the right-hand side as in the left one. We establish the decidability of equivalence for linear and non-deleting top-down transducers. These results are valid in the bottom-up case.*

Résumé. – *Nous considérons des transducteurs non déterministes d'arbres dans le cas lettre à lettre. Nous établissons la décidabilité de l'équivalence pour les transducteurs descendants linéaires et complets. Ces résultats s'étendent au cas des transducteurs ascendants.*

1. INTRODUCTION

Tree transducers which are a generalization of rational transformations in the word case (*see* [1], [3] for a synthesis), were introduced by W. C. Rounds [15] and J. W. Thatcher [17]. They have been widely studied. The authors have chosen either the algebraic point of view ([2], [9], [4]), or the machine point of view ([7], [8], [16]). Naturally, the question arises whether or not the results obtained for transformations in the word case can be transferred to tree transducers. The situation is different. For instance, we have to distinguish two main classes of tree transducers: top-down transducers which process

(*) Received November 1992; Revised July 1993. This work is supported in part by GDR "Mathematiques et Informatique", by project "Modèles logiques de la programmation" of the PRC "Informatique" and by ESPRIT Basic Research Action 6317 ASMICS 2.

(1) L.I.F.L., U.R.A. 369 C.N.R.S. University of Lille 1, 59655 Villeneuve d'Ascq Cedex, France.
e-mail:{andre, dauchet} @ lifl.lifl.fr

0988-3754/94/05/$ 4.00/© AFCET-Gauthier-Villars

the input trees from the root to the leaves and bottom-up transducers for which, on the contrary, the computations begin at the leaves and finish at the root. In 1975, J. Engelfriet proposed a comparison between these classes of tree transformations [7].

In this paper, we investigate the equivalence problem for a particular class of tree transducers. Two transducers are called *equivalent* if and only if they define the same transformations, that is to say if every input tree has the same set of output trees in both transducers.

In the word case, equivalence is undecidable in the non-deterministic case (Griffiths 1968) and it is decidable in the deterministic one (Bird 1973, Valiant 1974) (*see* [10]).

For trees, equivalence is in general undecidable in the non-deterministic case and it is decidable for deterministic transducers in the bottom-up case (K. Zachar 1978, [19]) and in the top-down one (Z. Esik 1979 [11]). More recently, in 1990, H. Seidl showed that equivalence is decidable for finite-valued bottom-up finite state transducers [16].

Linear and non-deleting letter to letter transducers (in the non-deterministic case) are studied here. Informally, these transducers only modify the label of the nodes of the trees and for every node can make a permutation (called here *torsion*) of the subtrees (precise definitions can be found in section 2).

First, *torsion-free* letter to letter top-down tree transducers are introduced and investigated (section 3). These transducers, which are only *relabelings*, preserve the skeleton of the trees. Using a classical coding (couples of trees are encoded in trees by "superposition"), we can associate a recognizable forest with the tree transformation we consider and so we easily prove that equivalence is decidable. Then, we show that the equivalence problem for linear and non-deleting letter to letter top-down transducers can be reduced to the equivalence problem for relabelings (section 4). The main problem, which is illustrated in the following example, is: even if T and T' are equivalent transducers, for some trees, computations with the same torsions cannot be realized in T and in T'.

Example: Let T and T' be two linear and non-deleting letter to letter top-down tree transducers defined by:

$$
\begin{aligned}
T:\ & q(\sigma(x, y)) \to \delta(q_1(x), q_2(y)) & & \\
& q_1(a(x)) \to a(q_1(x)) & \qquad & q_1(a) \to a \\
& q_2(\alpha(x)) \to \alpha(q_2(x)) & \qquad & q_2(\alpha) \to \alpha
\end{aligned}
$$

$$
\begin{array}{lll}
T' : & k(\sigma(x, y)) \to \delta(k'_1(x), k_2(y)) & k(\sigma(x, y)) \to \delta(k_1(x), k'_2(y)) \\
& k(\sigma(x, y)) \to \delta(k_3(y), k_4(x)) & \\
& k_1(a(x)) \to a(k_1(x)) & k_1(a) \to a \\
& k_2(\alpha(x)) \to \alpha(k_2(x)) & k_2(\alpha) \to \alpha \\
& k'_1(a(x)) \to a(k_{11}(x)) & k'_2(\alpha(x)) \to \alpha(k_{22}(x)) \\
& k_{11}(a(x)) \to a(k_{11}(x)) & k_{11}(a) \to a \\
& k_{22}(\alpha(x)) \to \alpha(k_{22}(x)) & k_{22}(\alpha) \to \alpha \\
& k_3(\alpha) \to a & k_4(a) \to \alpha
\end{array}
$$

T and T' are equivalent transducers because they realize the same transformations : $\hat{T} = \hat{T}' = \{(\sigma(a^n, \alpha^m), \delta(a^n, \alpha^m)), n, m \in \mathbb{N}\}$. But for $(\sigma(a, \alpha), \delta(a, \alpha))$ different torsions are used in the first step of the computations: for T, the rule used initially is $q(\sigma(x, y)) \to \delta(q_1(x), q_2(y))$ when, for T', $k(\sigma(x, y)) \to \delta(k_3(y), k_4(x))$ is used.

The key part of our proof consists in showing that this phenomenon is of "bounded depth" (lemmas 4.1, 4.2). So, we can encode the transducers we consider in torsion-free transducers. For technical reasons, infinitary transducers (that is to say transducers for which from each state an infinite number of trees can be transformed) are first studied (section 4.3). The results we obtain are valid in the general case (section 4.4). Finally, we extend the previous result to bottom-up transducers (section 4.5).

2. PRELIMINARIES

Main definitions and results about tree transducers can be found in J. Engelfriet's papers ([7], [8], [10]) and in the book of F. Gecseg and M. Steinby [13]. In this section, we just give basic definitions and properties used in the paper.

2.1. Trees

A ranked alphabet is a pair (Σ, ρ) where Σ is a finite alphabet and ρ is a mapping from Σ to $\mathbb{N}$. Usually, we will write Σ for short. For any σ in Σ, $\rho(\sigma)$ is called the *rank* of σ. The subset Σ_m of Σ is the set of letters of *rank* m.

For $p \geqq 1$, we denote by X_p the set $\{x_1, \cdots, x_p\}$ of variables. X_0 is the empty set.

Given a ranked alphabet Σ and a set X_p of variables, the set of all trees over Σ and indexed by X_p, denoted by $T_\Sigma(X_p)$, is inductively defined by $X_p \subseteqq T_\Sigma(X_p)$ and if $\sigma \in \Sigma_n$ and $t_1, \cdots, t_n \in T_\Sigma(X_p)$ then $\sigma(t_1, \cdots, t_n) \in T_\Sigma(X_p)$. For short, $T_\Sigma(X_0)$ is written T_Σ.

The *depth* of a tree $t \in T_\Sigma(X_p)$, denoted by $\pi(t)$, is defined by $\pi(t) = 0$ if $t \in \Sigma_0$ or $t \in X_p$ and $\pi(t) = 1 + \max\{\pi(t_1), \cdots, \pi(t_n)\}$ if $t = \sigma(t_1, \cdots, t_n)$.

For any $p \in \mathbb{N}\,(p \geqq 1)$, $[p]$ denotes the set $\{1, \cdots, p\}$.

A torsion θ from $[p]$ to $[q]$ is a mapping from $[p]$ to $[q]$. We denote it by $\langle q;\, \theta(1), \cdots, \theta(p)\rangle$. Especially, $id_{[n]}$ will denote the identity on $[n]$.

2.2. Letter to letter top-down tree transducers

DEFINITIONS: A *top-down tree transducer* is a 5-tuple $T = \langle \Sigma, \Delta, Q, I, R\rangle$ where Σ and Δ are ranked alphabets of respectively input and output symbols, Q is a finite set of states, I is a subset of Q of initial states and R is a finite set of rewriting rules of the form $q(\sigma(x_1, \cdots, x_n)) \to \tau(q_1(x_{\theta(1)}), \cdots, q_p(x_{\theta(p)}))$ with $\sigma \in \Sigma$, $\tau \in T_\Delta(X_p)$, $q, q_1, \cdots, q_p$ states of Q, and θ mapping from $[p]$ to $[n]$ (if $n = 0$ we have a rule of the form $q(\sigma) \to \tau$). θ is called a *torsion*.

A top-down transducer is *torsion-free* if, for every rule, the torsion θ is the identity.

A top-down transducer is *letter to letter* if, for every rule, τ belongs to Δ.

$t \mapsto t'$ if and only if there exist $t_0 \in T_\Sigma(X_1)$, $\sigma \in \Sigma_n$, $t_1, \cdots, t_n \in T_\Sigma$, $\delta \in \Delta_m$, $q, q_1, \cdots, q_m \in Q$, a rule $q(\sigma(x_1, \cdots, x_n)) \to \delta(q_1(x_{\theta(1)}), \cdots, q_m(x_{\theta(m)}))$ in R and $t = t_0(q(\sigma(t_1, \cdots, t_n)))$, $t' = t_0(\delta(q_1(t_{\theta(1)}), \cdots, q_m(t_{\theta(m)})))$. $\overset{*}{\mapsto}$ denotes the reflexive and transitive closure of $\mapsto$.

For any state q in Q, $\hat{T}_q$ denotes the transformation realized from state q. Formally, $\hat{T}_q = \{(t, u) \in T_\Sigma \times T_\Delta / q(t) \overset{*}{\mapsto} u\}$.

$\hat{T}$ denotes the tree transformation associated with T : $\hat{T} = \bigcup_{q \in I} \hat{T}_q$.

The *domain* of a tree transformation $\hat{T}$, denoted by $\mathrm{dom}(\hat{T})$, is the set $\{t \in T_\Sigma / \exists\, u \in T_\Delta,\ (t, u) \in \hat{T}\}$.

The *range* of a tree transformation $\hat{T}$, denoted by $\mathrm{im}(\hat{T})$, is the set $\{u \in T_\Delta / \exists\, t \in T_\Sigma,\ (t, u) \in \hat{T}\}$.

A state of a transducer is *infinitary* (resp. *finitary*) if and only if an infinite (resp. a finite) number of trees is transformed from this state. A transducer for which all states are infinitary is said to be *infinitary*.

A top-down tree transducer *is deterministic* if and only if the set of initial states is a singleton and there are no two rules with the same left-hand side. A transducer is *linear* (respectively *non-deleting, torsion-free*) if and if for each rule the torsion θ is injective (respectively surjective, the identity).

Two transducers T and T' are *equivalent* if and only if the tree transformations $\hat{T}$ and $\hat{T}'$ associated with these transducers are equal.

PROPERTY 2.1: *For every non-deleting letter to letter top-down tree transducer* T, *for every* $(t, u) \in \hat{T}$, $\pi(t) = \pi(u)$.

Notations: A torsion-free to letter transducer is also called a *relabeling*. **T-LAB** denotes the class of all top-down relabelings. **LCT-LL** (resp. **LCB-LL**) denotes the class of linear and non-deleting letter top-down (resp. bottom-up) transducers. **REC** is the class of recognizable forests.

3. EQUIVALENCE OF TORSION-FREE LETTER TO LETTER TOP-DOWN TRANSDUCERS

In this section, torsion-free letter to letter top-down transducers (called here relabelings) are considered. To establish the decidability of equivalence for the so defined class, we use a coding introduced by Doner [6] in the sixties and used in the Rabin's theorem (in the case of in finite trees) [14]. In the word case this construction was chosen by C. Frougny and J. Sakarovitch [12] to study rational relations with bounded delay, and in the tree case by M. Dauchet and S. Tison to prove the decidability of the theory of ground rewrite systems [5], [18].

For any relabeling T, for any couple of trees $(t, u) \in \hat{T}$, t and u have the same skeleton. So, to encode (t, u) in a tree, denoted by $[t, u]$, we just "superpose" the trees. For instance, $[t, u] = [b, \beta]([a, \alpha]([a, \alpha]), [c, \gamma])$ is the code of $(t, u) = (b(a(a), c), \beta(\alpha(\alpha), \gamma))$.

With every relabeling $T = \langle \Sigma, \Delta, Q, I, R \rangle$, we associate the automaton $A_T = \langle \Gamma, Q, I, R' \rangle$ where $\Gamma = \Sigma \times \Delta$ and R' is defined as follows:

$q([\sigma, \delta](x_1, \cdots, x_n)) \to [\sigma, \delta](q_1(x_1), \cdots, q_n(x_n))$ is a rule of R'

if and only if $q(\sigma(x_1, \cdots, x_n)) \to \delta(q_1(x_1), \cdots, q_n(x_n))$ is a rule of R.

It is easy to show that $[t, u] \in F(A_T) \Leftrightarrow (t, u) \in \hat{T}$ (where $F(A_T)$ denotes the forest recognized by the automaton A_T) and so it is possible to associate a recognizable forest with the tree transformation we consider and conversely. Thus, we inherit the good closure and decidability properties of recognizable forests.

PROPERTY: The class **REC** of recognizable forests is effectively closed under union, intersection and complementation and emptiness is decidable [13].

Closure of **T-LAB** under union.

Let T_1 and T_2 be two relabelings. With T_1 and T_2 we associate the automata A_1 and A_2 (as defined before). So

$$\begin{aligned}\hat{T}_1 \cup \hat{T}_2 &= \{(t, u)/[t, u] \in \hat{T}_1 \text{ or } (t, u) \in \hat{T}_2\}\\ &= \{(t, u)/[t, u] \in F(A_1) \text{ or } [t, u] \in F(A_2)\}\\ &= \{(t, u)/[t, u] \in F(A_1) \cup F(A_2)\}.\end{aligned}$$

As **REC** is closed under union, there exists an automaton A such that $F(A) = F(A_1) \cup F(A_2)$. So $\hat{T}_1 \cup \hat{T}_2 = \{(t, u)/[t, u] \in F(A)\}$. Now, let T be the relabeling associated with A then we obtain $\hat{T}_1 \cup \hat{T}_2 = \{(t, u) \in \hat{T}\} = \hat{T}$ and **T-LAB** is closed under union.

In the same way, we show that **T-LAB** is closed under intersection and difference. Emptiness is decidable. So we obtain,

THEOREM 3.1: *Equivalence in* **T-LAB** *is decidable.*

Proof: We use the fact two relabelings T_1 and T_2 are equivalent if and only if $(\hat{T}_1 - \hat{T}_2) \cup (\hat{T}_2 - \hat{T}_1) = \emptyset$. □

Remark: The following example illustrates the fact that we lose, in the case of transducers of **LCT-LL**, the closure under intersection.

Example:

$$\hat{T}_1 = \{(b(a^n, a^m), b(a_1^n, a_2^m)), n \in \mathbb{N}, m \in \mathbb{N}\}$$

and

$$\hat{T}_2 = \{(b(a^n, a^m), b(a_1^m, a_2^n), n \in \mathbb{N}, m \in \mathbb{N}\}.$$

We obtain $\hat{T}_1 \cap \hat{T}_2 = \{(b(a^n, a^n), b(a_1^n, a_2^m)), n \in \mathbb{N}\}$ which is not realizable by a top-down transducer because its domain is not recognizable [13].

4. EQUIVALENCE OF LINEAR AND NON-DELETING LETTER TO LETTER TRANSDUCERS

4.1. Preliminaries

In this part, we first establish the decidability of equivalence in **LCT-LL**. The main problem was illustrated in the example of the introduction.

First, we show that for two equivalent transformations the same torsions are used except for a finite number of trees (lemmas 4.1, 4.2). Next, for any integer Λ we built the Λ-normalized form T_{Θ}^{Λ} of a transducer T such that:

1. equivalence of Λ-normalized forms is easy to decide: these Λ-normalized forms are relabelings and so we use the result of section 3;

2. if T and T' are equivalent transducers then there is some integer Λ such that T_{Θ}^{Λ} and $T_{\Theta}'^{\Lambda}$ are equivalent (we use the fact that if T and T' are equivalent then the same torsions are used except for a finite number of trees).

As equivalence of T_{Θ}^{Λ} and $T_{\Theta}'^{\Lambda}$ is decidable (part 3), equivalence of T and T' is semi-decidable. Because non-equivalence is obviously semi-decidable, we conclude that equivalence is decidable (theorem 4.1). As a corollary, we obtain the same result for bottom-up transducers (theorem 4.2).

DEFINITIONS: Two sets of states $\{q_1, \cdots, q_n\}$ and $\{k_1, \cdots, k_m\}$ *are globally equivalent* if and only if $\bigcup_{i \in [n]} (\hat{T}_{q_i}) = \bigcup_{j \in [m]} (\hat{T}_{k_j})$.

Let $\mathcal{T}$ be a computation on $t = \sigma(t_1, \cdots, t_n)$ from state q.

$$\mathcal{T} : q(\sigma(t_1, \cdots, t_n)) \mapsto \delta(q_1(t_{\theta(1)}), \cdots, q_n(t_{\theta(n)})) \overset{*}{\mapsto} \delta(u_1, \cdots, u_n).$$

The *initial transformation* on t from state q is the triple (σ, δ, θ).

Notation: $\hat{T}_{q(\sigma, \delta, \theta)}$ denotes the transformation realized from state q by using the initial transformation (σ, δ, θ).

Remark: Results to be discussed below are described for letters of rank less than or equal to 2. They are easily transferred to the general situation.

Furthermore, for technical reasons, in sections 4.2 and 4.3, we will only consider infinitary transducers. **LCT-LL$_{\mathbf{i}}$** will denote the subclass of infinitary transducers of **LCT-LL**. These results are valid for **LCT-LL** (section 4.4).

4.2. Initial transformations realized from two globally equivalent sets of states of a transducers of LCT-LL$_{\mathbf{i}}$

4.2.1. *Case of trees of the form $\sigma(t_1, t_2)$ with $\pi(t_1) \neq \pi(t_2)$*

LEMMA 4.1: *From two globally equivalent sets of states the same initial transformations are realized on trees of the form $\sigma(t_1, t_2)$ for with $\pi(t_1) \neq \pi(t_2)$.*

Proof: Let E and F be two globally equivalent sets of states of a transducer T of **LCT-LL$_\mathbf{i}$**, q be a state of E and $(\sigma(t_1, t_2), \delta(u_1, u_2))$ be a couple of trees of $\hat{T}_{q_{(\sigma, \delta, \theta)}}$ with $\pi(t_1) \neq \pi(t_2)$. Let $\theta = id_{[2]}$ (the other case is similar) then, by property 2.1, $\pi(u_1) = \pi(t_1)$, $\pi(u_2) = \pi(t_2)$ and so $\pi(u_1) \neq \pi(u_2)$.

Suppose that there exist k in F and $\mu \neq \theta$ (here, $\mu = \langle 2; 2, 1\rangle$) such that $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{k_{(\sigma, \delta, \mu)}}$. Therefore, we would have $k(\sigma(t_1, t_2)) \mapsto \delta(k_1(t_2), k_2(t_1)) \overset{*}{\mapsto} \delta(u_1, u_2)$ and then $\pi(t_2)$ would be equal to $\pi(u_1)$ which contradicts the hypothesis. So, because E and F are globally equivalent sets of states, there exists at least one state $k \in F$ such that $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{k_{(\sigma, \delta, \theta)}}$. □

4.2.2. *Case of trees of the form $\sigma(t_1, t_2)$ with $\pi(t_1) = \pi(t_2)$*

The example of section 1 illustrates the fact that, from two equivalent states, initial transformations with different torsions can be realized for trees of the form $\sigma(t_1, t_2)$ with $\pi(t_1) = \pi(t_2)$. In the following lemma, we show that this phenomenon is of "bounded depth".

LEMMA 4.2: *From two globally equivalent sets of states the same initial transformations are realized on trees of the form $\sigma(t_1, t_2)$ with $\pi(t_1) = \pi(t_2)$, except for a finite number of trees.*

Proof: Let E and F be two globally equivalent sets of states of a transducer T of **LCT-LL$_\mathbf{i}$**. We consider the difference $\hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ (the problem is analogous if we consider $\hat{T}_{F_{(\sigma, \delta, \theta)}} - \hat{T}_{E_{(\sigma, \delta, \theta)}}$).

Let $\theta = id_{[2]}$ (the other case is similar).

With every couple $(\sigma(t_1, t_2), \delta(u_1, u_2)) \in \hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ we associate the set $C_1 = \{(k_1, k_2)$ such that $\exists\, k \in F$, $k(\sigma(x_1, x_2)) \to \delta(k_1(x_1), k_2(x_2))$ is a rule of T and $(t_1, u_1) \notin \hat{T}_{k_1}\}$ and the set $C_2 = \{(k_1, k_2)$ such that $\exists\, k \in F$, $k(\sigma(x_1, x_2)) \to \delta(k_1(x_1), k_2(x_2))$ is a rule of T and $(t_2, u_2) \notin \hat{T}_{k_2}\}$.

If $\hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ was infinite then, because T is a finite state transducer, there would exist $(\sigma(t_1, t_2), \delta(u_1, u_2))$ and $(\sigma(t_1', t_2'), \delta(u_1', u_2'))$ in $\hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ associated with the same sets C_1 and C_2 and such that $\pi(t_1) \neq \pi(t_2')$. Then $(\sigma(t_1, t_2'), \delta(u_1, u_2'))$ would be in $\hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ which contradicts lemma 4.1. Thus the difference $\hat{T}_{E_{(\sigma, \delta, \theta)}} - \hat{T}_{F_{(\sigma, \delta, \theta)}}$ is finite. □

4.3. Λ-Normalized form of a transducer of $\mathbf{LCT\text{-}LL_i}$

For any integer Λ, we associate with any transducer T of $\mathbf{LCT\text{-}LL_i}$ its Λ-normalized form built in two steps.

First, for every state q, for every couple of trees $(t, u) \in \hat{T}_q$, such that $\pi(t) \leqq \Lambda$, we add a rule of the form $q^{<\Lambda}(t) \to u$ if $\pi(t) < \Lambda$ or of the form $q^{\Lambda}(t) \to u$ if $\pi(t) = \Lambda$, where t and u are identified with new letters. We also adapt the "non-ground" rules of T so that the computation (t, u) is not possible otherwise. We obtain T^{Λ} with is called the Λ-*semi-normalized* form of T. We show that (lemma 4.4) if Λ is large enough then, from sets of globally equivalent states, transformations with the same torsions can be realized for all trees.

Then, we remove the torsions in the right-hand side of the rules of the Λ-semi-normalized form T^{Λ}, an indication of the torsion being encoded in each letter, and we obtain the Λ-normalized form denoted by T^{Λ}_{Θ}. For instance, with the rule $q(\sigma(x_1, x_2)) \to \delta(q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ we obtain the rule $q(\sigma(x_1, x_2)) \to \langle\delta, \theta\rangle (q_{\theta^{-1}(1)}(x_1), q_{\theta^{-1}(2)}(x_2))$ ($\langle\delta, \theta\rangle$ is a new letter).

4.3.1. Λ-*semi-normalized form: definition and construction*

Let $T = \langle\Sigma, \Delta, Q, I, R\rangle$ be a transducer of $\mathbf{LCT\text{-}LL_i}$. For any integer Λ, we associate with T the transducer $T^{\Lambda} = \langle\Sigma \cup \Sigma^{\Lambda}, \Delta \cup \Delta^{\Lambda}, Q^{\Lambda}, I^{\Lambda}, R^{\Lambda}\rangle$ where Σ^{Λ} and Δ^{Λ} are new alphabets, the letters of which can be interpreted as trees of *dom* ($\hat{T}$) and *im* ($\hat{T}$) of depth less than or equal to Λ, and Q^{Λ}, I^{Λ} and R^{Λ} are defined by

– $q^{<\Lambda}$ and q^{Λ} are states of Q^{Λ} if and only if q is a state of Q and they are in I^{Λ} if and only if q is in I.

– $q^{<\Lambda}(t) \to u$ (resp. $q^{\Lambda}(t) \to u$) is a rule of R^{Λ}, t is a letter of Σ^{Λ} and u is a letter of Δ^{Λ} if and only if $(t, u) \in \hat{T}_q$ with $t \in T_{\Sigma}$ and $\pi(t) < \Lambda$ (resp. $\pi(t) = \Lambda$).

– $q^{\Lambda}(\sigma(x)) \to \delta(q_i^{\Lambda}(x))$ is a rule of R^{Λ} if and only if $q(\sigma(x)) \to \delta(q_i(x))$ is a rule of R.

– $q^{\Lambda}(\sigma(x_1, x_2)) \to \delta(q_i^{\Lambda}(x_{\theta(1)}), q_j^{\Lambda}(x_{\theta(2)}))$,

$q^{\Lambda}(\sigma(x_1, x_2)) \to \delta(q_i^{<\Lambda}(x_{\theta(1)}), \delta(q_j^{\Lambda}(x_{\theta(2)}))$ and

$q^{\Lambda}(\sigma(x_1, x_2)) \to \delta(q_i^{\Lambda}(x_{\theta(1)}), q_j^{<\Lambda}(x_{\theta(2)}))$ are rules of R^{Λ}

if and only if $q(\sigma(x_1, x_2)) \to \delta(q_i(x_{\theta(1)}), q_j(x_{\theta(2)}))$ is a rule of R.

Example: Let T and T' be the transducers defined in section 1. For $\Lambda = 1$, for instance, their Λ-semi-normalized forms T^{Λ} and T'^{Λ} are defined by:

Ground rules of T^{Λ}.

$$q_1^{<\Lambda}(a) \to a$$
$$q_2^{<\Lambda}(\alpha) \to \alpha$$
$$q_1^{\Lambda}(a(a)) \to a(a)$$
$$q_2^{\Lambda}(\alpha(\alpha)) \to \alpha(\alpha)$$
$$q^{\Lambda}(\sigma(a, \alpha)) \to \delta(a, \alpha)$$

Non-ground rules of T^{Λ}.

$$q^{\Lambda}(\sigma(x, y)) \to \delta(q_1^{\Lambda}(x), q_2^{\Lambda}(y))$$
$$q^{\Lambda}(\sigma(x, y)) \to \delta(q_1^{<\Lambda}(x), q_2^{\Lambda}(y))$$
$$q^{\Lambda}(\sigma(x, y)) \to \delta(q_1^{\Lambda}(x), q_2^{<\Lambda}(y))$$
$$q_1^{\Lambda}(a(x)) \to \alpha(q_1^{\Lambda}(x))$$
$$q_2^{\Lambda}(\alpha(x)) \to \alpha(q_2^{\Lambda}(x))$$

Ground rules of T'^{Λ}.

$$k_1^{<\Lambda}(a) \to a \qquad k_2^{<\Lambda}(\alpha) \to \alpha$$
$$k_{11}^{<\Lambda}(a) \to a \qquad k_{22}^{<\Lambda}(\alpha) \to \alpha$$
$$k_3^{<\Lambda}(\alpha) \to a \qquad k_4^{<\Lambda}(a) \to \alpha$$
$$k_1^{\Lambda}(a(a)) \to a(a) \qquad k_2^{\Lambda}(\alpha(\alpha)) \to \alpha(\alpha)$$
$$k_{11}^{\Lambda}(a(a)) \to a(a) \qquad k_{22}^{\Lambda}(\alpha(\alpha)) \to \alpha(\alpha)$$
$$k_1'^{\Lambda}(a(a)) \to a(a) \qquad k_2'^{\Lambda}(\alpha(\alpha)) \to \alpha(\alpha)$$
$$k^{\Lambda}(\sigma(a, \alpha) \to \delta(a, \alpha)$$

Non-ground rules of T'^{Λ}.

$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1'^{\Lambda}(x), k_2^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1'^{<\Lambda}(x), k_2^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1'^{\Lambda}(x), k_2^{<\Lambda}(y))$$

$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1^{\Lambda}(x), k_2'^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1^{<\Lambda}(x), k_2'^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_1^{\Lambda}(x), k_2'^{<\Lambda}(y))$$

$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_3^{\Lambda}(y), k_4^{\Lambda}(x))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_3^{<\Lambda}(y), k_4^{\Lambda}(x))$$
$$k^{\Lambda}(\sigma(x, y)) \to \delta(k_3^{\Lambda}(y), k_4^{<\Lambda}(x))$$

$$k_1^{\Lambda}(a(x)) \to a(k_1^{\Lambda}(x)) \qquad k_2^{\Lambda}(\alpha(x)) \to \alpha(k_2^{\Lambda}(x))$$
$$k_1'^{\Lambda}(a(x)) \to a(k_{11}^{\Lambda}(x)) \qquad k_2'^{\Lambda}(\alpha(x)) \to \alpha(k_{22}^{\Lambda}(x))$$
$$k_{11}^{\Lambda}(a(x)) \to a(k_{11}^{\Lambda}(x)) \qquad k_{22}^{\Lambda}(\alpha(x)) \to \alpha(k_{22}^{\Lambda}(x))$$

Remark: Identification of T and T^{Λ}.

For every computation $q(t) \overset{*}{\mapsto} u$ in T, with $\pi(t) \leqq \Lambda$, we have in T^{Λ} one rule of the form $q^{<\Lambda}(t) \to u$ if $\pi(t) < \Lambda$, or of the form $q^{<\Lambda}(t) \to u$ if $\pi(t) \leqq \Lambda$. Here, in fact, we identify ground trees of depth less than or equal to Λ with new letters and thus it is unique computation for (t, u) in T^{Λ}.

For every couple of trees (t, u) in $\hat{T}$ with $\pi(t) > \Lambda$, there exists a unique decomposition of t and u in $t = t_0(t_1, \cdots, t_n)$ and $u = u_0(u_1, \cdots, u_n)$ where:

– for any i in $[n]$, $\pi(t_i) \leqq \Lambda$ and there exists no subtree of t, of depth less than or equal to Λ, for which t_i is a proper subtree

– and such that the computations

$$q(t_0(t_1, \cdots, t_n)) \overset{*}{\mapsto} u_0(q_1(t_{\theta(1)}), \cdots, q_n(t_{\theta(n)})) \quad \text{in } T$$

and

$$q^{\Lambda}(t_0(t_1, \cdots, t_n)) \overset{*}{\mapsto} u_0(q_1'(t_{\theta(1)}), \cdots, q_n'(t_{\theta(n)})) \quad \text{in } T^{\Lambda}$$

(for any i in $[n]$, q_i' is either $q_i^{<\Lambda}$ or q_i^{Λ}) are analogous, that is to say they only differ from one another in the label of the states ($q_i^{<\Lambda}$, or q_i^{Λ}, is used in T^{Λ} if q_i is used in T).

So, for any Λ, we identify T and T^{Λ} and, for any (t, u) in $T_{\Sigma \cup \Sigma^{\Lambda}}$, $\pi(t)$ will denote the depth of the "corresponding tree" of T_{Σ}.

In the next lemmas we show that, from two equivalent sets of states, if Λ is large enough then transformations with the same torsions can be realized for all trees in the Λ-normalized form.

LEMMA 4.3: *When Λ is large enough, from two globally equivalent sets of states of T^{Λ} the same initial transformations can be realized for all trees.*

Proof: Let E and F be two globally equivalent sets of states of T^{Λ} and let (t, u) be a couple of trees of $\hat{T}_E^{\Lambda} = \hat{T}_F^{\Lambda}$.

From lemma 4.2 we deduce that, if Λ is great enough, $\pi(t) > \Lambda$ implies that the same initial torsions can be used in the computation of (t, u) from E and F. In the case $\pi(t) \leqq \Lambda$ we get obviously the same result. □

LEMMA 4.4: *When Λ is large enough, from two globally equivalent sets of states of T^{Λ} transformations with the same torsions can be realized for all trees.*

Proof: The proof is by induction on the depth of the computations.

A computation such as $q(t_0(t_1, \cdots, t_n)) \overset{*}{\mapsto} u_0(q_1(t_{\theta(1)}), \cdots, q_n(t_{\theta(n)}))$ is said to be *of depth* p if and only if each state q_i (for $i \in [n]$) is obtained after exactly $p-1$ steps of rewriting.

We consider, here two equivalent states q and k. The result we obtain can be generalized without difficulties to globally equivalent sets of states.

Let (t, u) be a couple of trees of $\hat{T}_q^{\Lambda} = \hat{T}_k^{\Lambda}$, with $\pi(t) > \Lambda$ (in the case $\pi(t) \leqq \Lambda$ t is in fact a letter).

Suppose property true up to depth p. We show it is true again at depth $p+1$.

- First case: $p < \pi(t) - \Lambda$.

We consider the transformations realized from states q and k with the same torsions up to the depth p:

$$q(t) = q(t_0, \cdots, t_n)) \overset{*}{\mapsto} u_0(q_1(t_{\theta(1)}), \cdots, q_n(t_{\theta(n)}))$$
$$\overset{*}{\mapsto} u_0(u_1, \cdots, u_n) = u$$

and
$$k(t) = k(t_0(t_1, \cdots, t_n)) \overset{*}{\mapsto} u_0(k_1(t_{\theta(1)}), \cdots, k_n(t_{\theta(n)}))$$
$$\overset{*}{\mapsto} u_0(u_1, \cdots, u_n) = u$$

with $\pi(t_0) = p$.

Let $i \in [n]$ and let us consider the sets

$$C_i = \{q_i / q(t_0(x_1, \cdots, x_n)) \overset{*}{\mapsto} u_0(q_{j_1}(x_{\theta(1)}), \cdots, q_i(x_{\theta(i)}), \cdots, q_{j_n}(x_{\theta(n)}))\}$$

and in the same way

$$D_i = \{k_i / k(t_0(x_1, \cdots, x_n)) \overset{*}{\mapsto} u_0(k_{j_1}(x_{\theta(1)}), \cdots, k_i(x_{\theta(i)}), \cdots, k_{j_n}(x_{\theta(n)}))\}.$$

Suppose that from the sets of states C_i and D_i the transformation $(t_{\theta(i)}, u_i)$ cannot be realized with the same initial torsion. Then C_i and D_i would not be globally equivalent (lemma 4.3) and there would exist at least one couple of trees $(\bar{t}, \bar{u})$ and $\hat{T}_{C_i} - \hat{T}_{D_i}$.

So for $(t_0(t_1, \cdots, \bar{t}, \cdots, t_n), u_0(u_1, \cdots, \bar{u}, \cdots, \bar{u}_n))$ with $t_{\theta(i)} = \bar{t}$ we would have

$$q(t_0)(t_1, \cdots, \bar{t}, \cdots, t_n)) \overset{*}{\mapsto} u_0(q_1(t_{\theta(1)}), \cdots, q_i(\bar{t}), \cdots, q_n(t_{\theta(n)}))$$
$$\overset{*}{\mapsto} u_0(u_1, \cdots, \bar{u}, \cdots, u_n)$$

when for any computation

$$k(t_0(t_1, \cdots, \bar{t}, \cdots, t_n)) \overset{*}{\mapsto} u_0(k_{j_1}(t_{\theta(1)}), \cdots, k_i(\bar{t}), \cdots, k_{j_n}(t_{\theta(n)})),$$

where $k_i \in D_i$, $(\bar{t}, \bar{u})$ is not transformed from k_i.

Now, q and k are equivalent states and so

$$(t_0(t_1, \cdots, \bar{t}, \cdots, t_n), u_0(u_1, \cdots, \bar{u}, \cdots, \bar{u}_n)) \in \hat{T}_k.$$

Thus, we would have a computation from state k whose torsions are different from the torsions applied in the computation from state q before depth p. That contradicts the hypothesis. Consequently, the same torsions can be used at depth $p + 1$.

• Second case: $p = \pi(t) - \Lambda$.

We have the same transformations at depth $p + 1$ because, in this case, trees which are transformed are in fact letters of Σ^Λ. □

4.3.2. *Λ-normalized form of a transducer of* **LCT-LL$_i$**

Let $T = \langle \Sigma, \Delta, Q, I, R\rangle$ be a transducer of **LCT-LL$_i$** and $T^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta \cup \Delta^\Lambda, Q^\Lambda, I^\Lambda, R^\Lambda\rangle$ be its Λ-semi-normalized form.

We associate with T the transducer $T_\Theta^\Lambda = \langle \Sigma \cup \Sigma^\Lambda, \Delta_\Theta, Q^\Lambda, I^\Lambda, R_\Theta\rangle$ where Δ_Θ and R_Θ are defined by

– $q(\sigma) \to \langle \delta, \text{id}\rangle$ is a rule of R_Θ and $\langle \delta, \text{id}\rangle$ is a letter of Δ_Θ if and only if $q(\sigma) \to \delta$ is a rule of R^Λ.

– $q(\sigma(x)) \to \langle \delta, \text{id}\rangle (q_i(x))$ is a rule of R_Θ and $\langle \delta, \text{id}\rangle$ is a letter of Δ_Θ if and only if $q(\sigma(x)) \to \delta(q_i(x))$ is a rule of R^Λ.

– $q(\sigma(x_1, x_2)) \to \langle \delta, \theta\rangle (q_{\theta^{-1}(1)}(x_1), (q_{\theta^{-1}(2)}(x_2))$ is a rule of R_θ and $\langle \delta, \theta\rangle$ is a letter of Δ_Θ if and only if $q(\sigma(x_1, x_2))) \to \delta(q_1(x_{\theta(1)}), q_2(x_{\theta(2)}))$ is a rule of R^Λ.

Remark: The Λ-normalized form of any transducer of **LCT-LL$_i$** is a transducer of **T-LAB**.

Example: We consider the transducers T and T' defined in section 1 and whose Λ-semi-normalized forms were constructed in 4.3.1 for $\Lambda = 1$. To obtain their 1-normalized forms, we just remove the torsions which appear in the right-hand side of the rules; an indication of the torsion being encoded in each letter. We denote by *id* the identity and by μ the torsion defined by $\mu(1) = 2$ and $\mu(2) = 1$.

Ground rules of T_Θ^Λ.

$$\begin{array}{ll} q_1^{<\Lambda}(a) \to \langle a, \mathrm{id}\rangle & q_2^{<\Lambda}(\alpha) \to \langle \alpha, \mathrm{id}\rangle \\ q_1^{\Lambda}(a(a)) \to \langle a(a), \mathrm{id}\rangle & q_2^{\Lambda}(\alpha(\alpha)) \to \langle \alpha(\alpha), \mathrm{id}\rangle \\ q^{\Lambda}(\sigma(a, \alpha)) \to \langle \delta(a, \alpha)), \mathrm{id}\rangle & \end{array}$$

Non-ground rules of T_Θ^Λ.

$$\begin{array}{l} q^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (q_1^{\Lambda}(x), q_2^{\Lambda}(y)) \\ q^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (q_1^{<\Lambda}(x), q_2^{\Lambda}(y)) \\ q^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (q_1^{\Lambda}(x), q_2^{<\Lambda}(y)) \\ q_1^{\Lambda}(a(x)) \to \langle a, \mathrm{id}\rangle (q_1^{\Lambda}(x)) \\ q_2^{\Lambda}(\alpha(x)) \to \langle \alpha, \mathrm{id}\rangle (q_2^{\Lambda}(x)) \end{array}$$

Ground rules of $T_\Theta'^\Lambda$.

$$\begin{array}{ll} k_1^{<\Lambda}(a) \to \langle a, \mathrm{id}\rangle & k_2^{<\Lambda}(\alpha) \to \langle \alpha, \mathrm{id}\rangle \\ k_{11}^{<\Lambda}(a) \to \langle a, \mathrm{id}\rangle & k_{22}^{<\Lambda}(\alpha) \to \langle \alpha, \mathrm{id}\rangle \\ k_3^{<\Lambda}(\alpha) \to \langle a, \mathrm{id}\rangle & k_4^{<\Lambda}(a) \to \langle \alpha, \mathrm{id}\rangle \\ k_1^{\Lambda}(a(a)) \to \langle a(a), \mathrm{id}\rangle & k_2^{\Lambda}(\alpha(\alpha)) \to \langle \alpha(\alpha), \mathrm{id}\rangle \\ k_{11}^{\Lambda}(a(a)) \to \langle a(a), \mathrm{id}\rangle & k_{22}^{\Lambda}(\alpha(\alpha)) \to \langle \alpha(\alpha), \mathrm{id}\rangle \\ k_1'^{\Lambda}(a(a)) \to \langle a(a), \mathrm{id}\rangle & k_2'^{\Lambda}(\alpha(\alpha)) \to \langle \alpha(\alpha), \mathrm{id}\rangle \\ k^{\Lambda}(\sigma(a, \alpha) \to \langle \delta(a, \alpha), \mathrm{id}\rangle & \end{array}$$

Non-ground rules of $T_\Theta'^\Lambda$.

$$\begin{array}{l} k^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (k_1'^{\Lambda}(x), k_2^{\Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (k_1'^{<\Lambda}(x), k_2^{\Lambda}(y)) \\ k^{\Lambda}(\sigma(x, y)) \to \langle \delta, \mathrm{id}\rangle (k_1'^{\Lambda}(x), k_2^{<\Lambda}(y)) \end{array}$$

$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mathrm{id}\rangle (k_1^{\Lambda}(x), k_2'^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mathrm{id}\rangle (k_1^{<\Lambda}(x), k_2'^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mathrm{id}\rangle (k_1^{\Lambda}(x), k_2'^{<\Lambda}(y))$$

$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mu\rangle (k_4^{\Lambda}(x), k_3^{\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mu\rangle (k_4^{\Lambda}(x), k_3^{<\Lambda}(y))$$
$$k^{\Lambda}(\sigma(x, y)) \to \langle\delta, \mu\rangle (k_4^{<\Lambda}(x), k_3^{\Lambda}(y))$$

$$k_1^{\Lambda}(a(x)) \to \langle a, \mathrm{id}\rangle (k_1^{\Lambda}(x)) \qquad k_2^{\Lambda}(\alpha(x)) \to \langle \alpha, \mathrm{id}\rangle (k_2^{\Lambda}(x))$$
$$k_1'^{\Lambda}(a(x)) \to \langle a, \mathrm{id}\rangle (k_{11}^{\Lambda}(x)) \qquad k_2'^{\Lambda}(\alpha(x)) \to \langle \alpha, \mathrm{id}\rangle (k_{22}^{\Lambda}(x))$$
$$k_{11}^{\Lambda}(a(x)) \to \langle a, \mathrm{id}\rangle (k_{11}^{\Lambda}(x)) \qquad k_{22}^{\Lambda}(\alpha(x)) \to \langle \alpha, \mathrm{id}\rangle (k_{22}^{\Lambda}(x))$$

4.4. Decidability of equivalence in LCT-LL

The results obtained in the previous sections are easily transfered to the general situation. Obviously, for any Λ, the Λ-normalized form of any transducer of **LCT-LL** can be computed in the same way. Moreover, if N_F is the number of finitary states of a transducer T then the depth of any tree of *dom* $(\hat{T})$ is at most N_F and then as soon as Λ is greater than N_F, we will only have infinitary states. So lemmas 4.3 and 4.4 are valid in the general case.

LEMMA 4.5: *Let T and T' be two transducers of* **LCT-LL**. *T and T' are equivalent if and only if for some Λ the relabelings T_{Θ}^{Λ} and $T_{\Theta}'^{\Lambda}$ are equivalent.*

Proof: With lemma 4.4, it is obvious that if E and F are globally equivalent sets of T^{Λ} then, when Λ is large enough, $\hat{T}_{\Theta_E}^{\Lambda}$ and $\hat{T}_{\Theta_F}^{\Lambda}$ are equal. To conclude, we use the sets of initial states of T and T'. □

Lemma 4.5 states that equivalence is semi-decidable (because equivalence of relabelings is decidable). As non-equivalence is semi-decidable we get:

THEOREM 4.1: *Equivalence of linear and non-deleting letter to letter top-down transducers is decidable.*

4.5. Decidability of equivalence for bottom-up transducers

In this section, we show that the results obtained in the previous section are valid for the class of linear and non-deleting letter to letter bottom-up transducers (denoted by **LCB-LL**).

In [7], J. Engelfriet showed (theorem 2.9) that the class of linear and non-deleting bottom-up transducers is equal to the class of linear and non-deleting top-down transducers.

Let B_1 and B_2 be two linear and non-deleting letter to letter bottom-up transducers and T_1 and T_2 be the linear and non-deleting letter to letter top-down transducers which realize the same transformations. Because T_1 and T_2 are deduced from B_1 and B_2 by reversing the rules (*see* proof of theorem 2.9 in [7]), T_1 and T_2 are letter to letter transducers. It is obvious that B_1 and B_2 are equivalent if and only if T_1 and T_2 are equivalent. Now equivalence is decidable in **LCT-LL**, therefore it is in **LCB-LL**.

THEOREM 4.2: *Equivalence of linear and non-deleting letter to letter bottom-up transducers is decidable.*

5. CONCLUSION

In this paper we investigated the problem of the decidability of equivalence for a particular class of non deterministic tree transducers. We showed that equivalence is decidable for linear and non-deleting letter to letter transducers, in the top-down case and in the bottom-up one.

We conjecture that equivalence is decidable in the non-linear case as in the deleting one.

ACKNOWLEDGEMENTS

We wish to thank André Arnold, the referees and our colleague Francis Bossut for making very helpful suggestions which contributed to significant improvements of the results of this paper.

REFERENCES

1. L. BOASSON and J.-M. AUTEBERT, *Transductions rationnelles. Applications aux langages algébriques,* Masson, Collection ERI, 1988.
2. A. ARNOLD and M. DAUCHET, Morphismes et bimorphismes d'arbres, *Theoretical Computer Sciences,* 1982, *20*, pp. 33-93.
3. J. BERSTEL, *Transductions and context-free languages*, Teubner Verlag Stuttgart, 1979.
4. S. BOZAPALIDIS, Alphabetic tree relations, *TCS 99*. 1992, pp. 177-211.
5. M. DAUCHET and S. TISON, The theory of ground rewrite systems is decidable, *Proceedings of 5th IEEE Symposium on Logic in Computer Sciences*, Philadelphia, June 4-7, 1990, pp. 242-248.
6. J. DONER, Tree acceptors and some of their applications, *Journal of Computer and System Sciences,* 1970, *4*, pp. 406-451.
7. J. ENGELFRIET, Bottom-up and top-down tree transformations: a comparison, *Mathematical system theory*, 1975, *9*, pp. 198-231.
8. J. ENGELFRIET, Top-down tree transducers with regular look-ahead, *Mathematical System Theory*, 1977, *10*, pp. 289-303.

9. J. Engelfriet, A hierarchy of tree transucers, *Proc. 3^e colloque sur les Arbres en Algèbre et en Programmation,* Université de Lille I, 1978.
10. J. Engelfriet, *Some open questions and recent results on tree transducers and tree languages*, Formal language theory, Academic press, 1980, pp. 241-286.
11. Z. Esik, Decidability results concerning tree transducers I, *Acta Cybernetica*, 1980, Tome 5, Fasc. 1, pp. 1-20, Szeged.
12. C. Frougny and J. Sakarovitch, Relations rationnelles à délai borné. Actes des Journées Montoises, 49-52, Université de Mons-Hainaut, Belgique, 1990, *Proceedings of International Colloquium on Words, Languages and Combinatorics*, Kyoto, 1990.
13. F. Gecseg and M. Steinby, *Tree automata*, Akademiai Kiado, Budapest, 1984.
14. M. O. Rabin, Decidability of second-order theories and automata on infinite trees, 1969, *141, Trans. Amer. Math. Soc.,* pp. 1-35.
15. W. C. Rounds, *Trees, transducers and transformations*, Ph. D. Dissertation, Stanford University, 1968.
16. H. Seidl, Equivalence of finite-valued bottom-up finite state tree transducers is decidable, Universität des Saarlandes, Saarbrücken, Germany, *Proceeding of CAAP 90*, pp. 269-284.
17. J. W. Thatcher, *Generalized*^{2} sequential machines, *Journal of Computer and System Sciences*, 1970, *4,* pp. 339-367.
18. S. Tison, *Automates comme outil de décision dans les arbres,* Dossier d'habilitation à diriger des recherches, Lille, 1990.
19. Z. Zachar, The solvability of the equivalence problem for deterministic frontier-to-root tree transducers, *Acta Cybernetica*, 1978, *v.4.*, pp. 167-177.