

Beyond Regularity: Equational Tree Automata for Associative and Commutative Theories

Hitoshi Ohsaki

National Institute of Advanced Industrial Science and Technology
Nakoji 3-11-46, Amagasaki 661-0974, Japan

hitoshi.ohsaki@aist.go.jp
+81 6 6494 7823 (phone) +81 6 6491 5028 (fax)

Abstract. A new tree automata framework, called *equational tree automata*, is presented. In the newly introduced setting, congruence closures of recognizable tree languages are recognizable. Furthermore, we prove that in certain useful cases, recognizable tree languages are closed under union and intersection. To compare with early related work, e.g. [7], we discuss the relationship between linear bounded automata and equational tree automata. As a consequence, we obtain some (un)decidability results. We further present a hierarchy of 4 classes of tree languages.

Keywords. Tree automata, equational theory, decidability

1 Introduction

Over the past decade tree automata theory have been extensively studied and many applications were developed in various areas, e.g. for verification of cryptographic protocols [10, 12], subtyping in programming language [9] and reduction strategies in term rewriting [6]. The devised techniques are based on “regular” tree automata, which are the counterpart of classical finite automata. The tree automata framework is very useful in the sense that many decision problems are known to be decidable and recognizable tree languages are closed under boolean operations.

In contrast to the situations where regularity allows us to design terminating procedures easily, non-regular languages, such as term algebras modulo congruence, are considered to be troublesome in the framework. In fact, it is undecidable whether or not congruence closure of a regular tree language is regular [7]. Even, except for a few examples [11], AC-congruence closure of a regular tree language is not regular in general [5]. For instance, consider the signature $\mathcal{F} = \{f, a, b\}$, where f is a binary function symbol, and a and b are constants. Let L be the set of (ground) terms t such that the number of occurrences of a in t is the same as the number of occurrences of b in t . The tree language L is *not* regular, because of PUMPING LEMMA [2], although L' defined below is regular and L is the AC-closure of L' : $f(a, b) \in L'$ and $f(a, f(s, b)) \in L'$ for all $s \in L'$.

The aim of this paper is to introduce a new tree automata framework, called *equational tree automata* (ETA for short), in which congruence closures of recognizable languages are recognizable. Furthermore, we investigate the expressive

power of the new tree language theory by comparing with other well-known classes. In the sense, we are concerned in the paper with questions about equational tree automata like in the following:

- $\forall L : \text{TL}. P(L) \Rightarrow \exists \mathcal{A}/\mathcal{E} : \text{ETA}. \mathcal{L}(\mathcal{A}/\mathcal{E}) = L,$
- $\forall L : \text{TL}. Q(L) \Leftarrow \exists \mathcal{A}/\mathcal{E} : \text{ETA}. \mathcal{L}(\mathcal{A}/\mathcal{E}) = L.$

In the above formulae, P and Q are predicates for tree languages (TL for short). In particular, we spend the most of spaces to explain the relationship between the standard (finite bottom-up regular) tree automata and our equational extension. For instance, we discuss sufficient conditions for equational systems \mathcal{E} and tree languages L that satisfy

$$(Q1) \exists \mathcal{A} : \text{TA}. \mathcal{E}(\mathcal{L}(\mathcal{A})) = L \Leftrightarrow \exists \mathcal{B}/\mathcal{E} : \text{ETA}. \mathcal{L}(\mathcal{B}/\mathcal{E}) = L.$$

The question asks us: under which condition it holds or does not hold that a tree language is recognizable with a TA \mathcal{A} if and only if the \mathcal{E} -congruence closure is recognizable with an ETA \mathcal{B}/\mathcal{E} . Another instance to be considered is whether or not it holds that for any tree language L ,

$$(Q2) \exists \mathcal{A} : \text{TA}. \mathcal{L}(\mathcal{A}) = L \Leftrightarrow \exists \mathcal{B}/\mathcal{E} : \text{ETA}. \mathcal{L}(\mathcal{B}/\mathcal{E}) = \mathcal{E}(L).$$

The paper is organized as follows. The basics of tree automata and related theory are introduced in the next section. We show several positive answers to (Q1) and (Q2) in Section 3. We also present some decidability results by studying the relationship between linear bounded automata and equational tree automata. In Section 4, we show closure properties of union and intersection. We conclude in Section 5 by showing a hierarchy of 4 classes of tree languages. Open questions related to equational tree automata are also mentioned.

2 Preliminaries

A signature is a finite set \mathcal{F} of function symbols together with natural numbers n for every $f \in \mathcal{F}$. Here n is called the *arity* of f , denoted by $\text{arity}(f) = n$. Function symbols of arity 0 are called *constants*. We assume the existence of countably infinite sets of variables \mathcal{V} . The set $\mathcal{T}(\mathcal{F}, \mathcal{V})$ of terms is inductively defined as follows: $\mathcal{V} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{V})$ and $f(t_1, \dots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ if the arity of f is n and $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{V})$ for all $1 \leq i \leq n$. The set $\mathcal{T}(\mathcal{F}, \emptyset)$ of *ground* terms is denoted by $\mathcal{T}(\mathcal{F})$. Let \square be a fresh constant, named *hole*. The set $\mathcal{T}(\mathcal{F} \cup \{\square\}, \mathcal{V})$ of terms is denoted by $\mathcal{C}(\mathcal{F}, \mathcal{V})$. Elements of $\mathcal{C}(\mathcal{F}, \mathcal{V})$ are called *contexts*. The empty context is a hole. If C is a context with n holes and t_1, \dots, t_n are terms, $C[t_1, \dots, t_n]$ denotes the term obtained from C by replacing the holes from left to right by t_1, \dots, t_n . A *substitution* is a mapping σ from \mathcal{V} to $\mathcal{T}(\mathcal{F}, \mathcal{V})$. We write $t\sigma$ for the result of applying σ to a term t , where σ is extended as $f(t_1, \dots, t_n)\sigma = f(t_1\sigma, \dots, t_n\sigma)$. The set $\text{pos}(t)$ of *positions* in a term t is defined by

$$\text{pos}(t) = \begin{cases} \{\varepsilon\} & \text{if } t \text{ is a variable,} \\ \{\varepsilon\} \cup \{i \cdot p \mid 1 \leq i \leq n \text{ and } p \in \text{pos}(t_i)\} & \text{if } t = f(t_1, \dots, t_n). \end{cases}$$

Here ε is the empty sequence and $p \cdot q$ denotes concatenation of sequences p and q of positive integers. The position ε in $\text{pos}(t)$ is called the *root* of t and a symbol at ε is denoted by $\text{root}(t)$. A *subterm* of t at a position p is denoted by $t|_p$ and that is inductively defined as follows:

$$t|_p = \begin{cases} t & \text{if } p = \varepsilon, \\ t_{i|q} & \text{if } t = f(t_1, \dots, t_n) \text{ and } p = i \cdot q \text{ with } 1 \leq i \leq n. \end{cases}$$

The set $\text{pos}(t)$ is divided into two sets $\text{pos}_{\mathcal{V}}(t) = \{p \in \text{pos}(t) \mid t|_p \in \mathcal{V}\}$ and $\text{pos}_{\mathcal{F}}(t) = \text{pos}(t) \setminus \text{pos}_{\mathcal{V}}(t)$. Intuitively, $\text{pos}_{\mathcal{V}}(t)$ is the set of variable positions in t and $\text{pos}_{\mathcal{F}}(t)$ is the set of function symbols. The *length* of a term t , denoted by $|t|$, is the number of elements in $\text{pos}(t)$. The number of occurrences of a function symbol f in a term t is denoted by $\|t\|_f$. We write $\|t\|$ for the number of elements in $\text{pos}_{\mathcal{F}}(t)$. Note that $\|t\| = \sum_{f \in \mathcal{F}} \|t\|_f$. The set of variables appearing in t is denoted by $\text{var}(t)$ and the set of function symbols in t is denoted by $\text{fun}(t)$. Those multisets are denoted by $\text{var}_{\text{mul}}(t)$ and $\text{fun}_{\text{mul}}(t)$, respectively. The *height* of a term t , denoted by $\text{height}(t)$, is defined by $\text{height}(t) = 0$ if $t \in \mathcal{V}$; $\text{height}(t) = 1 + \max\{\text{height}(t_i) \mid 1 \leq i \leq n\}$ if $t = f(t_1, \dots, t_n)$.

An *equation* over the signature \mathcal{F} is a pair (s, t) of terms $s, t \in \mathcal{T}(\mathcal{F}, \mathcal{V})$. The equation (s, t) is denoted by $s \approx t$. An equation $l \approx r$ is called *linear* if neither l nor r contains multiple positions of the same variable. We say $l \approx r$ is *variable-preserving* if $\text{var}_{\text{mul}}(l) = \text{var}_{\text{mul}}(r)$. A variable-preserving equation $l \approx r$ is called *length-preserving* if $\|l\| = \|r\|$. An equation $l \approx r$ is called *ground* if $l, r \in \mathcal{T}(\mathcal{F})$, i.e. $\text{var}(l) = \text{var}(r) = \emptyset$. An *equational system* (ES for short) \mathcal{E} is a set of equations. Given a set $\mathcal{F}' (\subseteq \mathcal{F})$ of some binary function symbols. The set of associativity axioms $f(f(x, y), z) \approx f(x, f(y, z))$ for all $f \in \mathcal{F}'$ is denoted by $A(\mathcal{F}')$, and the set of commutativity axioms $f(x, y) \approx f(y, x)$ for all $f \in \mathcal{F}'$ is $C(\mathcal{F}')$. We write $AC(\mathcal{F}')$ for the union of $A(\mathcal{F}')$ and $C(\mathcal{F}')$. If unnecessary to be explicit, we simply write A , C and AC . An ES \mathcal{E} is called *linear* (variable-preserving, length-preserving, ground) if it consists of linear (variable-preserving, length-preserving, ground) equations. The binary relation $s \rightarrow_{\mathcal{E}} t$ is defined by letting $s = C[l\sigma]$ and $t = C[r\sigma]$ for some equation $l \approx r \in \mathcal{E}$, context $C \in \mathcal{C}(\mathcal{F}, \mathcal{V})$ and substitution σ over $\mathcal{T}(\mathcal{F}, \mathcal{V})$. In the paper, it is not guaranteed that $r \approx l \in \mathcal{E}$ even if $l \approx r \in \mathcal{E}$. The symmetric closure of $\rightarrow_{\mathcal{E}}$ is denoted by $\vdash_{\mathcal{E}}$ and the equivalence relation of $\rightarrow_{\mathcal{E}}$ (i.e., the reflexive-transitive closure of $\vdash_{\mathcal{E}}$) is denoted by $\sim_{\mathcal{E}}$.

A *tree automaton* (TA for short) $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R})$ consists of a signature \mathcal{F} , a finite set \mathcal{Q} of states (special constants with $\mathcal{F} \cap \mathcal{Q} = \emptyset$), a set $\mathcal{Q}_f (\subseteq \mathcal{Q})$ of final states and a finite set \mathcal{R} of transition rules in one of the following forms:

$$f(p_1, \dots, p_n) \rightarrow q$$

or

$$f(p_1, \dots, p_n) \rightarrow f(q_1, \dots, q_n)$$

for some $f \in \mathcal{F}$ and $p_1, \dots, p_n, q_1, \dots, q_n, q \in \mathcal{Q}$. In the latter form the root symbols in the left- and right-hand sides must be the same. An *equational tree*

automaton \mathcal{A}/\mathcal{E} is the combination of a TA \mathcal{A} and an ES \mathcal{E} . We often denote \mathcal{A}/\mathcal{E} by the 5-tuple $(\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, \mathcal{E})$ for convenience. An ETA \mathcal{A}/\mathcal{E} is called *regular* if \mathcal{R} consists of transition rules in the shape of $f(p_1, \dots, p_n) \rightarrow q$. We say \mathcal{A}/\mathcal{E} is *quasi-regular* if for all $l \rightarrow r \in \mathcal{R}$ such that $\text{root}(l) \notin \text{fun}(\mathcal{E})$, $r \in \mathcal{Q}$. Here fun is extended to be $\text{fun}(\mathcal{E}) = \bigcup_{l \approx r \in \mathcal{E}} (\text{fun}(l) \cup \text{fun}(r))$. Every TA is transformed to a regular TA with the same expressive power. The details are described in the next section. We say \mathcal{A}/\mathcal{E} is a C-TA (commutative-tree automaton) if $\mathcal{E} = \text{C}$. An ETA \mathcal{R}/\mathcal{E} with $\mathcal{E} = \text{A}$ is called an A-TA (associative-tree automaton). Likewise, if $\mathcal{E} = \text{AC}$, it is called an AC-TA. We write $s \rightarrow_{\mathcal{A}/\mathcal{E}} t$ if there exist s', t' such that $s \sim_{\mathcal{E}} s'$, $s' = C[l]$, $t \sim_{\mathcal{E}} t'$ and $t' = C[r]$ for some transition rule $l \rightarrow r \in \mathcal{R}$ and context $C \in \mathcal{C}(\mathcal{F} \cup \mathcal{Q})$. The relation $\rightarrow_{\mathcal{A}/\mathcal{E}}$ on $\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$ is called *move relation* of \mathcal{A}/\mathcal{E} . The transitive closure and reflexive-transitive closure of $\rightarrow_{\mathcal{A}/\mathcal{E}}$ are denoted by $\rightarrow_{\mathcal{A}/\mathcal{E}}^+$ and $\rightarrow_{\mathcal{A}/\mathcal{E}}^*$. For a TA \mathcal{A} , we simply write $\rightarrow_{\mathcal{A}}$, $\rightarrow_{\mathcal{A}}^+$ and $\rightarrow_{\mathcal{A}}^*$, instead. A term $t \in \mathcal{T}(\mathcal{F})$ is *accepted* by \mathcal{A}/\mathcal{E} if $t \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$ for some $q \in \mathcal{Q}_f$. Elements of $\mathcal{L}(\mathcal{A}/\mathcal{E})$ are ground terms accepted by \mathcal{A}/\mathcal{E} . A tree language L over \mathcal{F} is some subset of $\mathcal{T}(\mathcal{F})$. We say a tree language L is *recognizable* with an ETA if there exists \mathcal{A}/\mathcal{E} such that $L = \mathcal{L}(\mathcal{A}/\mathcal{E})$. A tree language L is called *regular* if $L = \mathcal{L}(\mathcal{A})$ for some regular TA \mathcal{A} . We write $\mathcal{E}(L)$ for $\{t \in \mathcal{T}(\mathcal{F}) \mid t \sim_{\mathcal{E}} s \text{ for some } s \in L\}$ and we say $\mathcal{E}(L)$ is \mathcal{E} -congruence closure of L . Note that $\mathcal{E}(\mathcal{E}(L)) = \mathcal{E}(L)$ for any tree language L , however, $\mathcal{E}_1(\mathcal{E}_2(L)) \neq (\mathcal{E}_1 \cup \mathcal{E}_2)(L)$. By definition, if a tree language L is recognizable with an \mathcal{E} -TA, so is $\mathcal{E}(L)$. In the questions (Q1) and (Q2), one direction ‘ \Rightarrow ’ is trivial, because $\mathcal{E}(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}/\mathcal{E})$ in any case.

Finally we spend the remaining space for explaining some concepts on tree grammars [4]. A *tree grammar* \mathcal{G} is the 4-tuple $(\mathcal{F}, \mathcal{Q}, \mathbf{q}_0, \mathcal{R})$, whose components are the signature \mathcal{F} , a finite set \mathcal{Q} of state symbols with fixed arities, an initial state constant $\mathbf{q}_0 (\in \mathcal{Q})$ and a finite set \mathcal{R} of pairs (l, r) of terms $l, r \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}, \mathcal{V})$ such that $\text{var}(r) \subseteq \text{var}(l)$ and $\text{fun}(l) \cap \mathcal{Q} \neq \emptyset$. We write $l \rightarrow r$ for a pair $(l, r) \in \mathcal{R}$ and we write $\rightarrow_{\mathcal{G}}$ for the induced binary relation. A tree language L is *generatable* if $L = \{t \in \mathcal{T}(\mathcal{F}) \mid \mathbf{q}_0 \rightarrow_{\mathcal{G}}^* t\}$ for some tree grammar \mathcal{G} . For instance, we consider the tree language $L_1 = \{f^n(g^n(h^n(a))) \mid n \geq 0\}$. The tree language L_1 is generatable. Actually, it is represented by the tree grammar $\mathcal{G}_1 = (\mathcal{F}, \{\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}, \mathbf{q}_0, \mathcal{R}_1)$, where \mathcal{R}_1 :

$$\begin{array}{lll} \mathbf{q}_0 \rightarrow \mathbf{q}_1(a, a, a) & \mathbf{q}_2(x, g(y), z) \rightarrow \mathbf{q}_2(x, y, g(z)) & \mathbf{q}_3(a, z) \rightarrow z \\ \mathbf{q}_1(x, y, z) \rightarrow \mathbf{q}_1(f(x), g(y), h(z)) & \mathbf{q}_2(x, a, z) \rightarrow \mathbf{q}_3(x, z) & \\ \mathbf{q}_1(x, y, z) \rightarrow \mathbf{q}_2(x, y, z) & \mathbf{q}_3(f(x), z) \rightarrow \mathbf{q}_3(x, f(z)) & \end{array}$$

3 Recognizability and Some Decidability Results

We start this section by showing the previous tree language L_1 is *not* recognizable with an ETA. First we state the following property.

Lemma 1. *For every TA \mathcal{A} there exists a regular TA \mathcal{B} such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.*

Proof. Suppose $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R})$. We take $\mathcal{B} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}')$ by letting $\mathcal{R}' = \{f(p_1, \dots, p_n) \rightarrow q \mid f \in \mathcal{F} \text{ and } p_1, \dots, p_n, q \in \mathcal{Q} \text{ such that } f(p_1, \dots, p_n) \rightarrow_{\mathcal{A}}^* q\}$.

$q\}$. Then it is easy to prove that for all $t \in \mathcal{T}(\mathcal{F})$, $t \rightarrow_{\mathcal{A}}^* q \in \mathcal{Q}$ if and only if $t \rightarrow_{\mathcal{B}}^* q \in \mathcal{Q}$. \square

We suppose to the contradiction that there is an ETA $\mathcal{A}/\mathcal{E} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, \mathcal{E})$ such that $\mathcal{L}(\mathcal{A}/\mathcal{E}) = L_1$. In this case \mathcal{E} is non-empty; otherwise, L_1 is recognized by a regular TA (Lemma 1). We take a term $t = f^n(g^n(h^n(a)))$ such that $n > |\mathcal{Q}| + |l|$ and $n > |\mathcal{Q}| + |r|$ for every $l \approx r \in \mathcal{E}$. Since t is accepted by \mathcal{A}/\mathcal{E} , there exists a derivation $t \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$ for some $q \in \mathcal{Q}_f$. Suppose $t \rightarrow_{\mathcal{A}}^* f^n(g^n(h^{n-m}(p))) \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$ and $p \in \mathcal{Q}$. In case $0 \leq m \leq |\mathcal{Q}|$, $f^n(g^n(h^{n-m}(p))) \sim_{\mathcal{E}} s'$ if and only if $f^n(g^n(h^{n-m}(p))) = s'$ (as there is no term $t' \notin L_1$ such that $t \sim_{\mathcal{E}} t'$). This admits a derivation $t \rightarrow_{\mathcal{A}}^* f^n(g^n(h^{n-m_1}(p'))) \rightarrow_{\mathcal{A}}^+ f^n(g^n(h^{n-m_2}(p''))) \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$ for some $p' \in \mathcal{Q}$ and $m_1 < m_2 \leq |\mathcal{Q}|$. This implies $f^n(g^n(h^{n-m_1}(h^{(m_1-m_2) \times i}(h^{m_2}(a)))))$ is accepted for any $i \geq 0$, but it contradicts to the assumption. Therefore L_1 is not recognizable, and thus, every generatable tree language is not recognizable. One should notice that $\{f^n(g^n(a)) \mid n \geq 0\}$ is recognized by an ETA (but not by a TA) having an equation $f(g(x)) \approx f(f(g(g(x))))$.

On the other hand, every recognizable tree language $\mathcal{L}(\mathcal{A}/\mathcal{E})$ is generatable whenever \mathcal{E} is linear. Let $\mathcal{A}/\mathcal{E} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, \mathcal{E})$. For all $f \in \mathcal{F}$, we take fresh state symbols q_f (for tree grammar) such that $\text{arity}(q_f) = \text{arity}(f)$. Now we define the tree grammar $\mathcal{G} = (\mathcal{F}, \mathcal{Q}', q_0, \mathcal{R}' \cup \mathcal{R}'')$ as follows: Let ϕ be the mapping defined by $\phi(t) = t$ if $t \in \mathcal{V}$ or $t \in \mathcal{Q}$; $\phi(t) = q_f(\phi(t_1), \dots, \phi(t_n))$ if $t = f(t_1, \dots, t_n)$. Then $\mathcal{Q}' = \mathcal{Q} \cup \{q_0, q_*\} \cup \{q_f \mid f \in \mathcal{F}\}$, $\mathcal{R}' = \{\phi(r) \rightarrow \phi(l) \mid l \rightarrow r \in \mathcal{R}\} \cup \{q_f(x_1, \dots, x_n) \rightarrow f(x_1, \dots, x_n) \mid f \in \mathcal{F}\} \cup \{q_0 \rightarrow q \mid q \in \mathcal{Q}_f\} \cup \{q_* \rightarrow q_f(q_*, \dots, q_*) \mid f \in \mathcal{F}\} \cup \{q_* \rightarrow q \mid q \in \mathcal{Q}\}$ and \mathcal{R}'' consists of rules $\phi(l)\sigma_1 \rightarrow \phi(r)\sigma_1\sigma_2$ for all $l \approx r \in \mathcal{E} \cup \mathcal{E}^{-1}$ and substitutions σ_1, σ_2 . Here $\mathcal{E}^{-1} = \{s \approx t \mid t \approx s \in \mathcal{E}\}$, $\sigma_1 = \{x \mapsto q(x_1, \dots, x_n)\}$ for some $q \in \mathcal{Q}' \setminus \{q_0, q_*\}$ and fresh variables x_1, \dots, x_n if $l = x (\in \mathcal{V})$; otherwise, $\sigma_1 = \emptyset$. Moreover, $\sigma_2 = \{y_1 \mapsto q_*, \dots, y_k \mapsto q_* \mid y_i \in \text{var}(r) \setminus \text{var}(l) \text{ for all } 1 \leq i \leq k\}$. The tree grammar \mathcal{G} satisfies $q_0 \rightarrow_{\mathcal{G}}^* t \in \mathcal{T}(\mathcal{F})$ if and only if $t \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$ for some $q \in \mathcal{Q}_f$.

We now consider the initial questions (Q1) and (Q2). First we observe that $\mathcal{E}(\mathcal{L}(\mathcal{A})) \neq \mathcal{L}(\mathcal{A}/\mathcal{E})$. For instance, let $\mathcal{E}_1 = \{f(x, x) \approx g(x, x)\}$ and $\mathcal{A}_1 = (\{f, g, a, b\}, \{q_1, q_2\}, \{q_2\}, \{a \rightarrow q_1, b \rightarrow q_1, f(q_1, q_1) \rightarrow q_2\})$. It is trivial that $\mathcal{L}(\mathcal{A}_1) = \{f(a, a), f(a, b), f(b, a), f(b, b)\}$, and then, $\mathcal{E}_1(\mathcal{L}(\mathcal{A}_1)) = \mathcal{L}(\mathcal{A}_1) \cup \{g(a, a), g(b, b)\}$. On the other hand, $\mathcal{L}(\mathcal{A}_1/\mathcal{E}_1) = \{f(s, t), g(s, t) \mid s, t \in \{a, b\}\}$. Note that $g(a, b) \rightarrow_{\mathcal{A}_1}^+ g(q_1, q_1) \sim_{\mathcal{E}_1} f(q_1, q_1) \rightarrow_{\mathcal{A}_1} q_2$, although $g(a, b) \notin \mathcal{E}_1(\mathcal{L}(\mathcal{A}_1))$.

Unfortunately, linearity of \mathcal{E} is insufficient to guarantee $\mathcal{E}(\mathcal{L}(\mathcal{A})) = \mathcal{L}(\mathcal{A}/\mathcal{E})$. Consider $\mathcal{A}_2 = (\{f, a, b\}, \{q_1, q_2, q_3\}, \{q_2\}, \{a \rightarrow q_1, b \rightarrow q_2, f(q_1, q_2) \rightarrow f(q_3, q_3), f(q_3, q_2) \rightarrow q_2\})$ and $\mathcal{E}_2 = \text{A}(\{f\})$. Since only b is reduced to q_2 by \mathcal{A}_2 , $\mathcal{L}(\mathcal{A}_2) = \{b\}$, and then $\mathcal{E}_2(\mathcal{L}(\mathcal{A}_2)) = \{b\}$. Let $t = f(f(a, b), b)$. The subterm $f(a, b)$ can be reduced to $f(q_1, q_2)$, then $t \rightarrow_{\mathcal{A}_2}^* f(f(q_1, q_2), q_2) \rightarrow_{\mathcal{A}_2} f(f(q_3, q_3), q_2)$. Due to associativity of f , $f(f(q_3, q_3), q_2) \sim_{\mathcal{E}_2} f(q_3, f(q_3, q_2))$, and thus, $f(q_3, f(q_3, q_2)) \rightarrow_{\mathcal{A}_2}^* q_2$. Hence t is accepted by $\mathcal{A}_2/\mathcal{E}_2$.

As a consequence, we obtain a partial solution to the initial questions.

Lemma 2. *Every regular ETA \mathcal{A}/\mathcal{E} with \mathcal{E} linear satisfies $\mathcal{E}(\mathcal{L}(\mathcal{A})) = \mathcal{L}(\mathcal{A}/\mathcal{E})$.*

Proof. Since $\mathcal{E}(\mathcal{L}(\mathcal{A})) \subseteq \mathcal{L}(\mathcal{A}/\mathcal{E})$ is trivial, we show the reverse. It suffices to prove $\rightarrow_{\mathcal{A}} \cdot \vdash_{\mathcal{E}} \subseteq \vdash_{\mathcal{E}} \cdot \rightarrow_{\bar{\mathcal{A}}}$. Here $\rightarrow_{\bar{\mathcal{A}}}$ denotes the reflexive closure of $\rightarrow_{\mathcal{A}}$. Let $\mathcal{A}/\mathcal{E} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, \mathcal{E})$ and suppose $s \rightarrow_{\mathcal{A}} t \vdash_{\mathcal{E}} u$ such that $s = C[l]$ and $t = C[r]$ for some $l \rightarrow r \in \mathcal{R}$, and moreover, $t = C'[l'\sigma]$ and $u = C'[r'\sigma]$ for some $l' \approx r' \in \mathcal{E} \cup \mathcal{E}^{-1}$. Since r is a state in \mathcal{Q} , there are the two cases as follows: If $l' \approx r'$ is applied above r , then r occurs below or at a variable position of l' . Suppose $\sigma = \{x_1 \mapsto t_1, \dots, x_i \mapsto D[r], \dots, x_n \mapsto t_n \mid x_i \in \text{var}(l') \cup \text{var}(r') \text{ for all } 1 \leq i \leq n\}$. Then we take $\sigma' = (\sigma \setminus \{x_i \mapsto D[r]\}) \cup \{x_i \mapsto D[l]\}$, and we obtain $s = C'[l'\sigma']$. Since $l' \approx r'$ is linear, we also obtain $C'[l'\sigma'] \vdash_{\mathcal{E}} C'[r'\sigma']$ and $C'[r'\sigma'] \rightarrow_{\bar{\mathcal{A}}} u$. Otherwise (i.e., if $l' \approx r'$ is applied at a parallel position of r), $s \rightarrow_{\mathcal{A}} t \vdash_{\mathcal{E}} u$ obviously implies $s \vdash_{\mathcal{E}} t' \rightarrow_{\mathcal{A}} u$ for some t' . \square

In this case the emptiness problem (i.e., a question if $\mathcal{L}(\mathcal{A}/\mathcal{E}) = \emptyset$) is decidable, because $\mathcal{L}(\mathcal{A}) = \emptyset$ if and only if $\mathcal{E}(\mathcal{L}(\mathcal{A})) = \emptyset$. In case \mathcal{E} is also length-preserving, membership and finiteness problems are decidable.

Along the same lines of the proof of Lemma 1, we obtain another statement.

Lemma 3. *Every ETA \mathcal{A}/C has a TA \mathcal{B} that satisfies $\mathcal{L}(\mathcal{A}/C) = \mathcal{L}(\mathcal{B})$.*

Proof. We use the similar construction of the proof of Lemma 1. Let $\mathcal{A}' = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}')$, where $\mathcal{R}' = \{f(p_1, \dots, p_n) \rightarrow q \mid f \in \mathcal{F} \text{ and } p_1, \dots, p_n, q \in \mathcal{Q} \text{ such that } f(p_1, \dots, p_n) \rightarrow_{\mathcal{A}/C}^* q\}$. It is easy to show $\mathcal{L}(\mathcal{A}/C) = \mathcal{L}(\mathcal{A}'/C)$. By Lemma 2 we have $\mathcal{L}(\mathcal{A}'/C) = C(\mathcal{L}(\mathcal{A}'))$. Moreover, every C-congruence closure of a regular tree language is recognizable with a regular TA (e.g. Exercise 12(3) in [2]). Hence there exists a TA \mathcal{B} such that $C(\mathcal{L}(\mathcal{A}')) = \mathcal{L}(\mathcal{B})$. \square

Accordingly, we have the positive partial solutions: if \mathcal{E} is a linear ES,

$$\forall L : \text{TL}, \exists \mathcal{A} : \text{regular TA. } \mathcal{E}(\mathcal{L}(\mathcal{A})) = L \Leftrightarrow \exists \mathcal{B}/\mathcal{E} : \text{regular ETA. } \mathcal{L}(\mathcal{B}/\mathcal{E}) = L.$$

As a special case, if $\mathcal{E} = C$, the regularity condition for \mathcal{A} and \mathcal{B} is unnecessary (Lemma 3). Moreover, we showed C-TA's have the same expressive power as regular TA's have. Another particular known case is \mathcal{E} ground. Dauchet and Tison [3] showed $\mathcal{E}(\mathcal{L}(\mathcal{A}))$ is regular whenever \mathcal{E} is ground, so there exists a regular ETA \mathcal{B}/\mathcal{E} such that $\mathcal{L}(\mathcal{B}/\mathcal{E}) = \mathcal{E}(\mathcal{L}(\mathcal{A}))$ (as there exists a regular TA \mathcal{B} such that $\mathcal{L}(\mathcal{B}) = \mathcal{E}(\mathcal{L}(\mathcal{B})) = \mathcal{L}(\mathcal{B}/\mathcal{E})$). We can also prove $\mathcal{L}(\mathcal{B}/\mathcal{E})$ is regular for any ETA \mathcal{B}/\mathcal{E} with \mathcal{E} ground, because $\rightarrow_{\mathcal{B}} \cdot \vdash_{\mathcal{E}} \subseteq \vdash_{\mathcal{E}} \cdot \rightarrow_{\mathcal{B}}$ in this case.

In contrast to the situation in Lemma 2, \mathcal{A} and \mathcal{B} in the above formula are not necessarily the same. This leads us to a new question; that is, whether or not the regularity of \mathcal{A} and \mathcal{B} is really essential in the formula. We are concerned with this question in the following part.

Ground term rewriting and regular tree automata are closely related each other. In fact, the ‘‘word problem’’ for ground theory is solvable, by reducing to TA's intersection-emptiness problem which is decidable. The same result holds for ground C-theory. In contrast, it is known that word problem for ground A-theory is undecidable, which was proved by Post [13]. In term rewriting, Deruyver and Gilleron [5] showed reachability of ground A-term rewriting is undecidable.

In equational tree automata framework a similar phenomenon can be found. The problem is regularizability of transition rules of A-symbols in A-TA. One should notice that every A-TA can be transformed to a *quasi-regular* A-TA.

Lemma 4. *Every \mathcal{A}/A has a quasi-regular \mathcal{B}/A that satisfies $\mathcal{L}(\mathcal{A}/A) = \mathcal{L}(\mathcal{B}/A)$.*

Proof. Let $\mathcal{A}/A = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, A)$. We take $\mathcal{A}'/A = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}', A)$, where $\mathcal{R}' = \{f(p_1, \dots, p_n) \rightarrow q \mid f \in \mathcal{F} \setminus \text{fun}(A) \text{ and } p_1, \dots, p_n, q \in \mathcal{Q} \text{ such that } f(p_1, \dots, p_n) \rightarrow_A^* q\}$. We take $\mathcal{B}/A = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{S}, A)$ by letting \mathcal{S} be the union of \mathcal{R}' and $\{l \rightarrow r \in \mathcal{R} \mid \text{root}(l) \in \text{fun}(A)\}$. Suppose $t \in \mathcal{T}(\mathcal{Q} \cup \mathcal{F})$ such that $t \rightarrow_{\mathcal{A}/A}^* q$ for some $q \in \mathcal{Q}$. Using the induction on the size of terms we show $t \rightarrow_{\mathcal{B}/A}^* q$ below. If t is a constant $c (\notin \mathcal{Q})$, there exists a rule $c \rightarrow q \in \mathcal{R}$, and thus $c \rightarrow q \in \mathcal{R}'$. If t is a state then $t = q$. Otherwise, there exist a term $t' \in \mathcal{T}(\mathcal{Q} \cup \mathcal{F})$ and a derivation $t \rightarrow_{\mathcal{A}/A}^* t' \rightarrow_{\mathcal{A}/A}^* q$ such that $t = C[t_1, \dots, t_n]$, $t' = C'[q_1, \dots, q_n]$, $C \sim_A C'$, $t_i \rightarrow_{\mathcal{A}/A}^* q_i (\in \mathcal{Q})$ and $t_i \in \mathcal{T}(\mathcal{Q} \cup (\mathcal{F} \setminus \text{fun}(A)))$ for all $1 \leq i \leq n$. Here we assume t_1, \dots, t_n are maximal subterms. Let $\mathcal{R}_1 = \{l \rightarrow r \in \mathcal{R} \mid \text{root}(l) \in \text{fun}(A)\}$ and $\mathcal{R}_2 = \{l \rightarrow r \in \mathcal{R} \mid \text{root}(l) \notin \text{fun}(A)\}$. Since $t_i \rightarrow_{\mathcal{A}/A}^* q_i$ is performed by \mathcal{R}_2 , we obtain $t_i \rightarrow_{\mathcal{A}'}^* q_i$. This implies $t \rightarrow_{\mathcal{A}'/A}^* t' \rightarrow_{\mathcal{A}'/A}^* q$. Again we observe that there exist a term t'' and a derivation $t' \rightarrow_{\mathcal{A}/A}^* t'' \rightarrow_{\mathcal{A}/A}^* q$ such that $t' = D[t'_1, \dots, t'_m]$, $t'' = D'[p_1, \dots, p_m]$, $D \sim_A D'$, $t'_i \rightarrow_{\mathcal{A}/A}^* p_i (\in \mathcal{Q})$ and $t'_i \in \mathcal{T}(\text{fun}(A))$ for all $1 \leq i \leq m$. Here we assume t'_1, \dots, t'_m are maximal. In this case $t'_i \rightarrow_{\mathcal{B}/A}^* p_i$ (actually $t'_i \rightarrow_{\mathcal{R}_1/A}^* p_i$), and thus, $t' \rightarrow_{\mathcal{B}/A}^* t''$. Since $|t''| < |t|$, $t'' \rightarrow_{\mathcal{B}/A}^* q$ by induction hypothesis. Moreover, from the fact that $\rightarrow_{\mathcal{A}'} \subseteq \rightarrow_{\mathcal{B}}$, we obtain $t \rightarrow_{\mathcal{B}/A}^* q$.

The inverse is an easy consequence of the property that $\rightarrow_{\mathcal{B}/A} \subseteq \rightarrow_{\mathcal{A}/A}^+$. \square

However, every quasi-regular A-TA is not always simplified to a regular A-TA with the same expressive power. In the remaining (but major) part of this section we explain the reason, by introducing *linear bounded automata* found in Hopcroft and Ullman [8]. A linear bounded automaton (LBA for short) M is the 7-tuple $(\Sigma, \mathcal{Q}, \mathcal{Q}_f, q_0, \#, \$, \mathcal{S})$. Each of the components denotes:

- Σ : a finite set of tape symbols,
- \mathcal{Q} : a finite set of state symbols such that $\Sigma \cap \mathcal{Q} = \emptyset$,
- $\mathcal{Q}_f (\subseteq \mathcal{Q})$: a set of final states,
- $q_0 (\in \mathcal{Q})$: an initial state symbol,
- $\# (\notin \Sigma)$: left-endmark,
- $\$ (\notin \Sigma)$: right-endmark,
- \mathcal{S} : a finite set of string rewrite rules in the form of either $aqb \rightarrow q'ab'$ or $qb \rightarrow b'q'$ for some $a, a', b, b' \in \Sigma$ and $q, q' \in \mathcal{Q}$. (Basic notions of string rewriting are explained, e.g. in [1].) As a special case of the former rule, it is allowed to be $b = \$$ whenever $b' = \$$. Similarly, in the latter case, $b = \#$ is allowed whenever $b' = \#$. If there exists a rule $aqb \rightarrow q'ab'$ for some $a \in \Sigma$, we assume \mathcal{S} contains $cqb \rightarrow q'cb'$ for all $c \in \Sigma$.

LBA is a Turing machine whose tape length is finitely bounded. As we showed in the example in Section 2, Turing machine (or tree grammar) is too general to discuss the expressive power of ETA. In the following part we show equivalence of LBA and (a special case of) ETA, and so we use such a resource bounded Turing machine.

A word w is a finite (possibly empty) sequence of alphabets over Σ . The empty word is denoted by ϵ and the set of all words over Σ is Σ^* . A language is a subset of Σ^* . A move relation on $(\Sigma \cup Q \cup \{\#, \$\})^*$ with respect to M , denoted by \rightarrow_M , is defined as follows: $u \rightarrow_M v$ if there exists a rule $l \rightarrow r \in \mathcal{S}$ such that $u = u_1 l u_2$ and $v = u_1 r u_2$. A word w is called accepted by M if $q_0 \# w \$ \rightarrow_M^* u q v$ for some $q \in Q_f$ and $u, v \in (\Sigma \cup \{\#, \$\})^*$. The set $\mathcal{L}(M)$ consists of words accepted by M . By definition, it is allowed to be $\epsilon \in \mathcal{L}(M)$. We say a language L is recognizable with an LBA if there exists an LBA M such that $L = \mathcal{L}(M)$.

It is known that emptiness problem is undecidable for LBA; that means, there is no algorithm deciding whether a language recognized by an arbitrary LBA is empty. This implies that for an arbitrary LBA M , if there exists an ETA \mathcal{A}/A simulating M , we may not find a regular ETA \mathcal{B}/A such that $\mathcal{L}(\mathcal{A}/A) = \mathcal{L}(\mathcal{B}/A)$. Otherwise, we can determine whether $\mathcal{L}(M) = \emptyset$ by examining $\mathcal{L}(\mathcal{B}) = \emptyset$, because $\mathcal{L}(\mathcal{B}/A) = \emptyset$ if and only if $\mathcal{L}(\mathcal{B}) = \emptyset$ due to Lemma 2.

Thus, all we have to do in the remaining part is to show that for an arbitrary LBA M , there exists an associated ETA \mathcal{A}_M/A such that $\mathcal{L}(M) = \emptyset$ if and only if $\mathcal{L}(\mathcal{A}_M/A) = \emptyset$.

Given an LBA $M = (\Sigma, Q, Q_f, q_0, \#, \$, \mathcal{S})$. Let us take $\mathcal{F}_M = \Sigma \cup \{q_0, \#, \$\} \cup \{f\}$ such that f is a fresh binary function symbol assumed to be associative. The set Q_M of state symbols is the union of $Q_1 = \{\alpha_q, \bar{\alpha}_q \mid q \in Q\}$ and $Q_2 = \{\beta_a \mid a \in \Sigma \cup \{\#, \$\}\}$ together with fresh state symbols, $*q_0, *#, \circ_1, \circ_2, \diamond$ such that $Q_{Mf} = \{\diamond\}$. The set \mathcal{R}_M consists of the following transition rules:

1. $q_0 \rightarrow *q_0$,
2. $\# \rightarrow *#$,
3. $f(*q_0, *#) \rightarrow f(\alpha_{q_0}, \beta_{\#})$,
4. $a \rightarrow \beta_a$ for all $a \in \Sigma \cup \{\$, \}$,
5. $f(\alpha_p, \beta_a) \rightarrow f(\beta_b, \alpha_q)$ if $p a \rightarrow b q \in \mathcal{S}$ for some $p, q \in Q$ and $a, b \in \Sigma$,
6. $f(\alpha_p, \beta_a) \rightarrow f(\bar{\alpha}_q, \beta_b)$ if $c p a \rightarrow q c b \in \mathcal{S}$ for some $p, q \in Q$ and $a, b, c \in \Sigma$,
7. $f(\beta_a, \bar{\alpha}_q) \rightarrow f(\alpha_q, \beta_a)$ for all $a \in \Sigma \cup \{\#, \$\}$ and $q \in Q$,
8. $f(\alpha_q, \beta_a) \rightarrow f(\circ_1, \beta_a)$ and $f(\alpha_q, \beta_{\#}) \rightarrow \circ_2$ for all $q \in Q_f$ and $a \in \Sigma \cup \{\$, \}$,
9. $f(\beta_a, \circ_1) \rightarrow \circ_1$ and $f(\circ_2, \beta_a) \rightarrow \circ_2$ for all $a \in \Sigma$,
10. $f(\beta_{\#}, \circ_1) \rightarrow \circ_2$
11. $f(\circ_2, \beta_{\$}) \rightarrow \diamond$.

Henceforth, we write $t = C_f[[t_1, \dots, t_n]]$ if $t = C[t_1, \dots, t_n]$ such that C is a non-empty and maximal context consisting of a function symbol f . If unnecessary f to be explicit, we simply write $t = C[[t_1, \dots, t_n]]$.

The idea of the previous construction is described below. We take an ETA $\mathcal{A}_M/A = (\mathcal{F}_M, \mathcal{Q}_M, \mathcal{Q}_{M_f}, \mathcal{R}_M, A)$ with $A = \{f(f(x, y), z) \approx f(x, f(y, z))\}$. In the setting, a word $w = a_1 a_2 a_3 \dots a_n$ is represented by a term $C[[a_1, a_2, a_3, \dots, a_n]]$ and an *initial instantaneous description* for w , i.e. $q_0 \# a_1 a_2 \dots a_n \$$, is represented as $C[[q_0, \#, a_1, a_2, \dots, a_n, \$]]$.

The first three rules 1–3 examine whether q_0 and $\#$ are located in the order of $q_0 \#$ at the initial stage. Using the transition rules 4, each tape symbol together with right-endmark of a term is replaced by a corresponding state symbol in \mathcal{Q}_M . This step is not necessarily performed at once.

In case M admits the move relation $\# p a_1 a_2 \dots a_n \$ \rightarrow_M \# b_1 q a_2 \dots a_n \$$, there exists the corresponding derivation

$$\begin{aligned} C[[\beta_{\#}, \alpha_p, \beta_{a_1}, \beta_{a_2}, \beta_{a_n}, \beta_{\$}]] &\sim_A f(f(\dots f(f(\beta_{\#}, f(\alpha_p, \beta_{a_1})), \beta_{a_2}) \dots, \beta_{a_n}), \beta_{\$}) \\ &\rightarrow_{\mathcal{A}_M} f(f(\dots f(f(\beta_{\#}, f(\beta_{b_1}, \alpha_q)), \beta_{a_2}) \dots, \beta_{a_n}), \beta_{\$}). \end{aligned}$$

If there is a rule $b_1 q a_2 \rightarrow r b_1 b_2 \in \mathcal{S}$ and it is applied at the next step, then

$$\begin{aligned} C[[\beta_{\#}, \beta_{b_1}, \alpha_q, \beta_{a_2}, \dots, \beta_{a_n}, \beta_{\$}]] &\sim_A f(f(\dots f(\beta_{\#}, f(\beta_{b_1}, f(\alpha_q, \beta_{a_2}))) \dots, \beta_{a_n}), \beta_{\$}) \\ &\rightarrow_{\mathcal{A}_M} f(f(\dots f(\beta_{\#}, f(\beta_{b_1}, f(\bar{\alpha}_r, \beta_{b_2}))) \dots, \beta_{a_n}), \beta_{\$}) \\ &\sim_A f(f(\dots f(\beta_{\#}, f(f(\beta_{b_1}, \bar{\alpha}_r), \beta_{b_2})) \dots, \beta_{a_n}), \beta_{\$}) \\ &\rightarrow_{\mathcal{A}_M} f(f(\dots f(\beta_{\#}, f(f(\alpha_r, \beta_{b_1}), \beta_{b_2})) \dots, \beta_{a_n}), \beta_{\$}). \end{aligned}$$

Lemma 5. *If an LBA M admits a move relation*

$$q_0 \# a_1 \dots a_n \$ \xrightarrow*_M b_0 b_1 b_2 \dots b_{i-1} p b_i \dots b_n b_{n+1},$$

the associated ETA \mathcal{A}_M/A simulates the computation sequence by resulting in the derivation

$$C[[q_0, \#, a_1, \dots, a_n, \$]] \xrightarrow*_{\mathcal{A}_M/A} C[[\beta_{b_0}, \beta_{b_1}, \dots, \beta_{b_{i-1}}, \alpha_p, \beta_{b_i}, \dots, \beta_{b_n}, \beta_{b_{n+1}}]].$$

Proof. Use induction on the length of M -move relation. □

Hence \mathcal{A}_M/A results in a term $C[[\beta_{b_0}, \beta_{b_1}, \dots, \beta_{b_{i-1}}, \alpha_p, \beta_{b_i}, \dots, \beta_{b_n}, \beta_{b_{n+1}}]]$, provided $q_0 \# a_1 \dots a_n \$ \xrightarrow*_M b_0 b_1 \dots b_{i-1} p b_i \dots b_n b_{n+1}$. In case $p \in \mathcal{Q}_f$, a subterm $f(\alpha_p, \beta_{b_i})$ is replaced by $f(\alpha_1, \beta_{b_i})$ using the rules 8. Moreover, the whole term is simplified to $f(\alpha_2, \beta_{\$})$ by the rules 9–10. Finally \diamond is obtained by applying the rule 11.

This obviously implies soundness of the construction, with respect to acceptability. To be formalized, it is represented as follows. Let $\Gamma = \Sigma \cup \mathcal{Q} \cup \{\#, \$\}$ and define the mapping

$$\langle w \rangle = \begin{cases} f(a, \langle u \rangle) & \text{if } w = a u \text{ for some } a \in \Gamma \text{ and } u \in \Gamma^+, \\ w & \text{if } w \in \Gamma. \end{cases}$$

Due to Lemma 5 together with the preceding observation, the soundness property is established.

Lemma 6. *Every \mathcal{A}_M/A associated with an LBA $M = (\Sigma, \mathcal{Q}, \mathcal{Q}_f, q_0, \#, \$, S)$ satisfies $\langle q_0 \# w \$ \rangle \rightarrow_{\mathcal{A}_M/A}^* \diamond$ for all $w \in \mathcal{L}(M)$. \square*

Next we show the reverse also holds. Looking at the transition rules of M , we can observe that: if $t \rightarrow_{\mathcal{A}_M/A}^* \diamond$, there exists a derivation represented as follows. Let $p \in \mathcal{Q}_f$.

$$\begin{aligned}
t &\sim_A C[[q_0, \#, a_1, \dots, a_n, \$]] \\
&\rightarrow_{\mathcal{A}_M/A}^+ C[[\alpha_{q_0}, \beta_{\#}, \beta_{a_1}, \dots, \beta_{a_n}, \beta_{\$}]] \quad \dots (1) \\
&\rightarrow_{\mathcal{A}_M/A}^* C[[\beta_{b_0}, \beta_{b_1}, \dots, \beta_{b_{i-1}}, \alpha_p, \beta_{b_i}, \dots, \beta_{b_n}, \beta_{b_{n+1}}]] \quad \dots (2) \\
&\rightarrow_{\mathcal{A}_M/A}^+ f(o_2, \beta_{\$}) \\
&\rightarrow_{\mathcal{A}_M/A} \diamond.
\end{aligned}$$

More precisely, we have a derivation $t \rightarrow_{\mathcal{A}_M/A}^* \diamond$ only in the case $t \rightarrow_{\mathcal{A}_M/A}^* f(o_2, \beta_{\$})$. On the other hand, t has to contain the initial state symbol q_0 and the endmarks $\#$ and $\$$ such that q_0 is located left-next to $\#$. And, for any $s, s' \in \mathcal{T}(\mathcal{F}_M \cup \mathcal{Q}_M)$, it holds that if $s \rightarrow_{\mathcal{A}_M/A}^* s'$ then $\|s\|_{\$} + \|s\|_{\beta_{\$}} + \|s\|_{\diamond} = \|s'\|_{\$} + \|s'\|_{\beta_{\$}} + \|s'\|_{\diamond}$ and

$$\begin{aligned}
&\sum_{q \in \mathcal{Q}} (\|s\|_q + \|s\|_{\alpha_q} + \|s\|_{\overline{\alpha}_q}) + \|s\|_{*q_0} + \|s\|_{o_1} + \|s\|_{o_2} + \|s\|_{\diamond} \\
&= \sum_{q \in \mathcal{Q}} (\|s'\|_q + \|s'\|_{\alpha_q} + \|s'\|_{\overline{\alpha}_q}) + \|s'\|_{*q_0} + \|s'\|_{o_1} + \|s'\|_{o_2} + \|s'\|_{\diamond}.
\end{aligned}$$

Moreover, $\|t\|_{\#} = \|t\|_{q_0}$ by the transition rule 3. Then,

$$t = C[[c_1, \dots, c_{j-1}, q_0, \#, c_j, \dots, c_{j+k}, \$, c_{j+k+1}, \dots, c_m]]$$

or

$$t = C[[c_1, \dots, c_{j-1}, \$, c_j, \dots, c_{j+k}, q_0, \#, c_{j+k+1}, \dots, c_m]]$$

for some context C and $c_1, \dots, c_m \in \Sigma$. Since $t \rightarrow_{\mathcal{A}_M/A}^* \diamond$, we can assume without loss of generality that

$$t \rightarrow_{\mathcal{A}_M/A}^* C[[\beta_{c_1}, \dots, \beta_{c_{j-1}}, \alpha_{q_0}, \beta_{\#}, \beta_{c_j}, \dots, \beta_{c_{j+k}}, \beta_{\$}, \beta_{c_{j+k+1}}, \dots, \beta_{c_m}]] \rightarrow_{\mathcal{A}_M/A}^* \diamond$$

or

$$t \rightarrow_{\mathcal{A}_M/A}^* C[[\beta_{c_1}, \dots, \beta_{c_{j-1}}, \beta_{\$}, \beta_{c_j}, \dots, \beta_{c_{j+k}}, \alpha_{q_0}, \beta_{\#}, \beta_{c_{j+k+1}}, \dots, \beta_{c_m}]] \rightarrow_{\mathcal{A}_M/A}^* \diamond.$$

Since $t \rightarrow_{\mathcal{A}_M/A}^* f(o_2, \beta_{\$})$, the former derivation is the case, which can be proved by induction on the length of a derivation $s \rightarrow_{\mathcal{A}_M/A}^* f(o_2, \beta_{\$})$. Moreover, we obtain $j = 1$ and $k = n - 1$ such that $c_i = a_i$ for all $1 \leq i \leq n$. Before we apply a transition rule 8, there is no applicable rules other than rules 5–7.

We let u be a term appearing in between (1) and (2), and we define the mapping \mathbf{str} as follows.

$$\mathbf{str}(s) = \begin{cases} \mathbf{str}(s_1) \mathbf{str}(s_2) & \text{if } s = f(s_1, s_2), \\ q & \text{if } s = \alpha_a \text{ for some } q \in \mathcal{Q}, \\ a & \text{if } s = \beta_a \text{ for some } a \in \Sigma \cup \{\#, \$\}. \end{cases}$$

Then we obtain the following property.

Lemma 7. *Let v be a term in (2). If $u \rightarrow_{\mathcal{A}_M/A}^* v$ and $\mathbf{str}(u) \in (\Sigma \cup \{\#, \$\} \cup \mathcal{Q})^*$, then $\mathbf{str}(u) \rightarrow_M^* \mathbf{str}(v)$.*

Proof. We use induction on the length of $u \rightarrow_{\mathcal{A}_M/A}^* v$. The base case is trivial, because $u = v$. For the induction step we suppose $u \rightarrow_{\mathcal{A}_M/A}^+ v$. By assumption, u does not contain $\bar{\alpha}_q$ for any $q \in \mathcal{Q}$. Let $u \rightarrow_{\mathcal{A}_M/A} u' \rightarrow_{\mathcal{A}_M/A}^* v$. If there exists a transition rule $f(\alpha_q, \beta_a) \rightarrow f(\beta_b, \alpha_r) \in \mathcal{R}_M$ and it is applied to u , then $\mathbf{str}(u') = \# d_1 \dots d_{i-2} b r d_i \dots d_n \$$ for some $d_1, \dots, d_{i-2}, b, d_i, \dots, d_n \in \Sigma$. So, $\mathbf{str}(u) = \# d_1 \dots d_{i-2} q a d_i \dots d_n \$$, and thus, $\mathbf{str}(u) \rightarrow_M \mathbf{str}(u')$. Otherwise, there is a rule $f(\alpha_q, \beta_a) \rightarrow f(\bar{\alpha}_r, \beta_b) \in \mathcal{R}_M$ and it is applied to u . In this case there is also a (and only) transition rule $f(\beta_c, \bar{\alpha}_r) \rightarrow f(\alpha_r, \beta_c)$ which is applicable to u' . So, $u' \rightarrow_{\mathcal{A}_M/A} u''$ such that $\mathbf{str}(u'') = \# d_1 \dots d_{i-2} c r b d_{i+1} \dots d_n \$$ for some $d_1, \dots, d_{i-2}, c, b, d_{i+1}, \dots, d_n \in \Sigma$. This implies $\mathbf{str}(u) = \# d_1 \dots d_{i-2} c a q d_{i+1} \dots d_n \$$ and $\mathbf{str}(u) \rightarrow_M \mathbf{str}(u'')$. \square

As a consequence, completeness (with respect to acceptability) is established.

Lemma 8. *Every \mathcal{A}_M/A associated with an LBA $M = (\Sigma, \mathcal{Q}, \mathcal{Q}_f, q_0, \#, \$, \mathcal{S})$ satisfies: for all $t \in \mathcal{T}(\mathcal{F}_M)$, if $t \rightarrow_{\mathcal{A}_M/A}^* \diamond$ then $t \sim_A \langle q_0 \# w \$ \rangle$ and $w \in \mathcal{L}(M)$. \square*

We know $\mathcal{L}(M)$ is empty if and only if $\mathcal{L}(\mathcal{A}_M/A)$ is empty. Moreover, the former property ($\mathcal{L}(M) = \emptyset$) is known to be undecidable, and so is the latter.

Corollary 1. *For an arbitrary A-TA it is undecidable whether a tree language recognized by the A-TA is empty. \square*

Hence A-TA is not always regularized, although it can be quasi-regularized.

Theorem 1. *There exists an ETA \mathcal{A}/A such that $\mathcal{L}(\mathcal{A}/A) \neq A(\mathcal{L}(B))$ for any TA B . \square*

In fact, the language $P = \{w \in \{a\}^* \mid |w| = 2^n \text{ and } n \geq 0\}$ is recognizable with LBA, and thus, the tree language $T = \{t \mid t \sim_A \langle q_0 \# w \$ \rangle \text{ and } w \in W\}$ is recognizable with an A-TA. However, T is not recognizable with a regular A-TA, and then $\{\langle q_0 \# w \$ \rangle \mid w \in P\}$ is not recognizable with a regular TA. In other words, even if there exists a tree language L such that $A(L) = \mathcal{L}(\mathcal{A}/A)$, L is not recognizable with a TA in general.

Undecidability of finiteness is also obtained, because finiteness problem of LBA is undecidable. Note that $\{\langle q_0 \# w \$ \rangle \mid w \in \mathcal{L}(M)\}$ is finite if and only if $\mathcal{L}(\mathcal{A}_M/A)$ is finite.

Corollary 2. *For an arbitrary A-TA it is undecidable whether a tree language recognized by the A-TA is finite. \square*

Furthermore, a question if $\mathcal{L}(M) = \Sigma^*$ is known to be undecidable for an arbitrary LBA M . This yields the following undecidability results.

Corollary 3. *Let \mathcal{A}/\mathcal{A} and \mathcal{B}/\mathcal{A} be ETA. It is undecidable to test the subset relation $\mathcal{L}(\mathcal{A}/\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B}/\mathcal{A})$. Equivalence test $\mathcal{L}(\mathcal{A}/\mathcal{A}) = \mathcal{L}(\mathcal{B}/\mathcal{A})$ is also undecidable.*

Proof. Let M_1 be an arbitrary LBA and M_2 be an LBA such that $\mathcal{L}(M_2) = \Sigma^*$, e.g. $M_2 = (\Sigma, \{q_0\}, \{q_0\}, q_0, \#, \$, \emptyset)$. Then we take $\mathcal{A} = \mathcal{A}_{M_2}$ and $\mathcal{B} = \mathcal{A}_{M_1}$ together with $\mathcal{A} = \{f(f(x, y), z) \approx f(x, f(y, z))\}$. As we can see, $\mathcal{L}(\mathcal{A}/\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B}/\mathcal{A})$ (and $\mathcal{L}(\mathcal{A}/\mathcal{A}) = \mathcal{L}(\mathcal{B}/\mathcal{A})$) if and only if $\Sigma^* = \mathcal{L}(M_1)$. \square

4 Closure Properties

As we discussed in the previous section, equational tree automata are sometimes too powerful. Nevertheless, the recognizable tree languages are still useful in a certain situation, as they are closed under two operations: union and intersection. In this section we discuss the closure properties of \mathcal{E} -tree languages.

Theorem 2. *If \mathcal{E} is a variable-preserving ES, the union of tree languages L_1, L_2 recognized by ETA's \mathcal{A}/\mathcal{E} and \mathcal{B}/\mathcal{E} is recognizable with an ETA \mathcal{C}/\mathcal{E} .*

Proof. Let $\mathcal{A}/\mathcal{E} = (\mathcal{Q}_A, \mathcal{F}, \mathcal{Q}_{A_f}, \mathcal{R}_A, \mathcal{E})$ and $\mathcal{B}/\mathcal{E} = (\mathcal{Q}_B, \mathcal{F}, \mathcal{Q}_{B_f}, \mathcal{R}_B, \mathcal{E})$. We assume without loss of generality that $\mathcal{Q}_A \cap \mathcal{Q}_B = \emptyset$. We take the TA $\mathcal{C} = (\mathcal{Q}, \mathcal{F}, \mathcal{Q}_f, \mathcal{R}, \mathcal{E})$ as follows. $\mathcal{Q} = \mathcal{Q}_A \cup \mathcal{Q}_B$, $\mathcal{Q}_f = \mathcal{Q}_{A_f} \cup \mathcal{Q}_{B_f}$ and $\mathcal{R} = \mathcal{R}_A \cup \mathcal{R}_B$. Below we show the two properties: (1) $s \rightarrow_{\mathcal{C}/\mathcal{E}}^* p \in \mathcal{Q}_A$ if and only if $s \rightarrow_{\mathcal{A}/\mathcal{E}}^* p$ and (2) $s \rightarrow_{\mathcal{C}/\mathcal{E}}^* q \in \mathcal{Q}_B$ if and only if $s \rightarrow_{\mathcal{B}/\mathcal{E}}^* q$. Since the “if” parts of both properties are trivial, it suffices to show the “only if”. We observe that if $t \rightarrow_{\mathcal{C}/\mathcal{E}} t'$ and $\text{fun}(t) \cap \mathcal{Q}_A \neq \emptyset$ then $\text{fun}(t') \cap \mathcal{Q}_A \neq \emptyset$, because of \mathcal{E} variable-preserving. Moreover, if $t \rightarrow_{\mathcal{A}/\mathcal{E}} t'$ then $\text{fun}(t') \cap \mathcal{Q}_A \neq \emptyset$. The same property holds also for \mathcal{Q}_B and $\rightarrow_{\mathcal{B}/\mathcal{E}}$. This implies that if $t \rightarrow_{\mathcal{C}/\mathcal{E}}^* q \notin \mathcal{Q}_B$ then $t \rightarrow_{\mathcal{A}/\mathcal{E}}^* q$. Hence the property (1) holds. Similarly, the property (2) can be proved. \square

Let P be a set of ill-formed tape statuses. For instance, P contains $\# w \$$ (missing a state symbol in a tape) and $\# p \# w \$$ (extra right-endmark). The set $\{\langle p \rangle \mid p \in P\}$ is regular, and then $L = \{t \mid t \sim_A \langle p \rangle \text{ and } p \in P\}$ is recognizable with a (regular) A-TA. We take the union of the tree languages L and $\mathcal{L}(\mathcal{A}_M/\mathcal{A})$, which is recognizable with an A-TA due to the above theorem. Since universality problem of LBA, i.e. a question if $\{q_0 \# w \$ \mid w \in \Sigma^*\} = \mathcal{L}(M)$, is undecidable, so is to test $L \cup \mathcal{L}(\mathcal{A}_M/\mathcal{A}) = \mathcal{T}(\mathcal{F}_M)$.

Corollary 4. *For an arbitrary ETA \mathcal{A}/\mathcal{A} over the signature \mathcal{F} it is undecidable whether $\mathcal{L}(\mathcal{A}/\mathcal{A}) = \mathcal{T}(\mathcal{F})$.* \square

Next we discuss the intersection. Regular tree languages are closed under intersection [2]. Tree languages recognizable with C-TA are also closed under intersection (Corollary 3). The remaining questions to be considered as useful cases are closedness of A- and AC-TA's.

Theorem 3. *If $\mathcal{E} = \text{A}$ or $\mathcal{E} = \text{AC}$, the intersection of tree languages L_1, L_2 recognized by ETA's \mathcal{A}/\mathcal{E} and \mathcal{B}/\mathcal{E} is recognizable with an ETA \mathcal{C}/\mathcal{E} .*

Proof. We show the proof sketch for the A-case. Note that the same proof construction can be applied to the AC-case. Let $\mathcal{G} (\subseteq \mathcal{F})$ be the set of binary symbols of A-axioms and let $\mathcal{A}/\mathcal{A} = (\mathcal{F}, \mathcal{Q}_{\mathcal{A}}, \mathcal{Q}_{\mathcal{A}f}, \mathcal{R}_{\mathcal{A}}, \mathcal{A})$ and $\mathcal{B}/\mathcal{A} = (\mathcal{F}, \mathcal{Q}_{\mathcal{B}}, \mathcal{Q}_{\mathcal{B}f}, \mathcal{R}_{\mathcal{B}}, \mathcal{A})$ such that $\mathcal{Q}_{\mathcal{A}} \cap \mathcal{Q}_{\mathcal{B}} = \emptyset$. Due to Corollary 4, we assume without loss of generality that \mathcal{A}/\mathcal{A} and \mathcal{B}/\mathcal{A} are quasi-regular. Define the ETA $\mathcal{C}/\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_f, \mathcal{R}, \mathcal{A})$ as follows. $\mathcal{Q} = (\mathcal{Q}_{\mathcal{A}} \times \mathcal{Q}_{\mathcal{B}}) \cup \mathcal{Q}_{\mathcal{A}} \cup \mathcal{Q}_{\mathcal{B}}$ and $\mathcal{Q}_f = \mathcal{Q}_{\mathcal{A}f} \times \mathcal{Q}_{\mathcal{B}f}$. The set \mathcal{R} of transition rules are the union of the 4 sets $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3$ and \mathcal{R}_4 defined below.

$$\begin{array}{ll}
\mathcal{R}_1 & f((p_1, q_1), \dots, (p_n, q_n)) \rightarrow (p, q) \quad \begin{array}{l} \forall f \in \mathcal{F} \setminus \mathcal{G} \\ \forall f(p_1, \dots, p_n) \rightarrow p \in \mathcal{R}_{\mathcal{A}} \\ \forall f(q_1, \dots, q_n) \rightarrow q \in \mathcal{R}_{\mathcal{B}} \end{array} \\
\mathcal{R}_2 & \begin{array}{l} g((p_1, q_1), (p_2, q_2)) \rightarrow g((p, q_1), q_2) \\ g((p_1, q_1), (p_2, q_2)) \rightarrow g((r_1, q_1), (r_2, q_2)) \end{array} \quad \begin{array}{l} \forall g \in \mathcal{G}, \forall q_1, q_2 \in \mathcal{Q}_{\mathcal{B}} \\ \forall g(p_1, p_2) \rightarrow p \in \mathcal{R}_{\mathcal{A}} \\ \forall g \in \mathcal{G}, \forall q_1, q_2 \in \mathcal{Q}_{\mathcal{B}} \\ \forall g(p_1, p_2) \rightarrow g(r_1, r_2) \in \mathcal{R}_{\mathcal{A}} \end{array} \\
\mathcal{R}_3 & \begin{array}{l} g(q_1, (p_2, q_2)) \rightarrow (p_2, q) \\ g(q_1, (p_2, q_2)) \rightarrow g(r_1, (p_2, r_2)) \end{array} \quad \begin{array}{l} \forall g \in \mathcal{G}, \forall p_1, p_2 \in \mathcal{Q}_{\mathcal{A}} \\ \forall g(q_1, q_2) \rightarrow q \in \mathcal{R}_{\mathcal{B}} \\ \forall g \in \mathcal{G}, \forall p_1, p_2 \in \mathcal{Q}_{\mathcal{A}} \\ \forall g(q_1, q_2) \rightarrow g(r_1, r_2) \in \mathcal{R}_{\mathcal{B}} \end{array} \\
\mathcal{R}_4 & \begin{array}{l} g((p, q_1), q_2) \rightarrow g(q_1, (p, q_2)) \\ g(q_1, (p, q_2)) \rightarrow g((p, q_1), q_2) \end{array} \quad \forall g \in \mathcal{G}, \forall q_1, q_2, q \in \mathcal{Q}_{\mathcal{B}}
\end{array}$$

The ETA \mathcal{C}/\mathcal{A} satisfies that for any term $t \in \mathcal{T}(\mathcal{F})$, $t \rightarrow_{\mathcal{C}/\mathcal{A}}^* (p, q) \in \mathcal{Q}_f$ if and only if $t \rightarrow_{\mathcal{A}/\mathcal{A}}^* p \in \mathcal{Q}_{\mathcal{A}f}$ and $t \rightarrow_{\mathcal{B}/\mathcal{A}}^* q \in \mathcal{Q}_{\mathcal{B}f}$. \square

This theorem holds also for \mathcal{E} -TA whose ES \mathcal{E} consists of equations in the shape of $f(x, f(y, z)) \approx f(y, f(x, z))$. Kaji *et al.* [10] pointed out that in order to express some key-exchange protocols using term rewriting, those axioms are required.

5 Concluding Remarks

In the paper we introduced equational tree automata together with the undecidability results. We also showed the closure properties of union and intersection for equational tree automata. The newly introduced tree automata framework is almost optimal from the beneficial reason and it obtains our goal: to propose a class of tree languages in which congruence closures of recognized languages are recognizable. Furthermore, we presented the relationship between the standard TA and our equational extension. Fig.1 illustrates the result on a hierarchy of 4 classes of tree languages (in case \mathcal{E} is linear). In the above figure the smallest area TA denotes the class of tree languages recognizable with a regular TA.

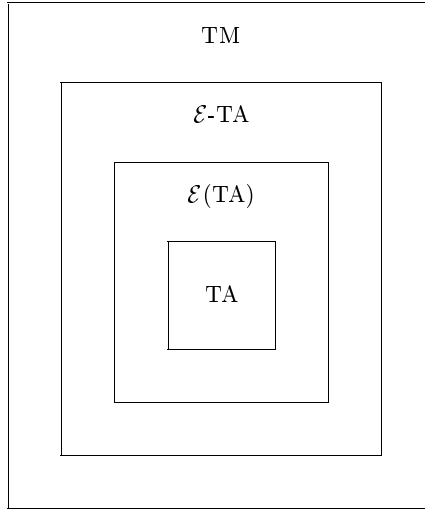


Fig. 1. Hierarchy of tree languages

The second smallest area $\mathcal{E}(\text{TA})$ is \mathcal{E} -congruence closure of TA, and $\mathcal{E}\text{-TA}$ is the class of ETA. The largest area TM denotes the set of generatable tree languages. These inclusion relations are strict each other. It is unclear so far whether the strict inclusion holds between TM and $\mathcal{E}\text{-TA}$ also for \mathcal{E} non-linear.

In order to discuss (minimal) equational extension of tree automata it would be important to consider whether the following question is positive:

- If $\mathcal{E} = \text{A}$ (or $\mathcal{E} = \text{AC}$),

$$\forall \mathcal{A}, \mathcal{B} : \text{regular TA}, \exists \mathcal{C} : \text{regular TA. } \mathcal{E}(\mathcal{L}(\mathcal{A})) \cap \mathcal{E}(\mathcal{L}(\mathcal{B})) = \mathcal{E}(\mathcal{L}(\mathcal{C}))?$$

There are two more interesting questions about equational tree automata.

- Regularizability of AC-TA;

$$\forall \mathcal{A}/\text{AC} : \text{ETA}, \exists \mathcal{B}/\text{AC} : \text{regular ETA. } \mathcal{L}(\mathcal{A}/\text{AC}) = \mathcal{L}(\mathcal{B}/\text{AC})?$$

- Closure under complement of A- and AC-TA; if $\mathcal{E} = \text{A}$ (or $\mathcal{E} = \text{AC}$),

$$\forall \mathcal{A}/\mathcal{E} : \text{ETA}, \exists \mathcal{B}/\mathcal{E} : \text{ETA. } \overline{\mathcal{L}(\mathcal{A}/\mathcal{E})} = \mathcal{L}(\mathcal{B}/\mathcal{E})?$$

We observe that equational tree automata are closely related to *context-sensitive grammar* (Section 9.3, [8]), so it is conjectured that the second question is positively solved.

Acknowledgements. I would like to thank Aart Middeldorp for his continuous help and encouragement.

References

1. R.V. Book and F. Otto: *String-Rewriting Systems*, Texts and Monographs in Computer Science, Springer-Verlag, 1993.
2. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison and M. Tommasi: *Tree Automata Techniques and Applications*, draft, 1997. Available on <http://www.grappa.univ-lille3.fr/tata/>.
3. M. Dauchet and S. Tison: *The Theory of Ground Rewrite Systems is Decidable*, Proc. 5th LICS, Philadelphia (Pennsylvania), pp. 242–248, 1990.
4. M. Dauchet and S. Tison: Structural Complexity of Classes of Tree Languages, In *Tree Automata and Languages*, Studies in Computer Science and Artificial Intelligence 10, pp. 327–353, Elsevier Science Publishers B.V., 1992.
5. A. Deruyver and R. Gilleron: *The Reachability Problem for Ground TRS and Some Extensions*, Proc. CAAP '89, Barcelona (Spain), LNCS 351, pp. 227–243, 1989.
6. I. Durand and A. Middeldorp: *Decidable Call by Need Computations in Term Rewriting (Extended Abstract)*, Proc. 14th CADE, Townsville (Australia), LNAI 1249, pp. 4–18, 1997.
7. R. Gilleron and S. Tison: *Regular Tree Languages and Rewriting Systems*, Fundamenta Informaticae 24, pp. 157–175, 1995.
8. J.E. Hopcroft and J.D. Ullman: *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Company, 1979.
9. H. Hosoya, J. Vouillon and B.C. Pierce: *Regular Expression Types for XML*, Proc. 5th ICFP, Montreal (Canada), SIGPLAN Notices 35(9), pp. 11–22, 2000.
10. Y. Kaji, T. Fujiwara and T. Kasami: *Solving a Unification Problem under Constrained Substitutions Using Tree Automata*, Journal of Symbolic Computation 23, pp. 79–117, 1997.
11. D. Lugiez and J.L. Moysset: *Tree Automata Help One to Solve Equational Formulae in AC-Theories*, Journal of Symbolic Computation 18(4), pp. 297–318, 1994.
12. D. Monniaux: *Abstracting Cryptographic Protocols with Tree Automata*, Proc. 6th SAS, Venice (Italy), LNCS 1694, pp. 149–163, 1999.
13. E.L. Post: *Recursive Unsolvability of a Problem of Thue*, Journal of Symbolic Logic 13, pp. 1–11, 1947.