

# Restarting Tree Automata

**Heiko Stamer**

University of Kassel  
Department of Mathematics/Computer Science  
Heinrich-Plett-Straße 40, 34132 Kassel, Germany

stamer@theory.informatik.uni-kassel.de  
76F7 3011 329D 27DB 8D7C 3F97 4F58 4EB8 FB2B E14F

FORMAT-Workshop  
Universität Frankfurt, 3. März 2006

U N I K A S S E L  
V E R S I T Ä T

## 1 Introduction

## 2 Top-down Finite Tree Automata

## 3 Restarting Tree Automata

## 4 Open Questions

## Agenda:

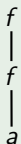
- Extend the computation model of restarting automata to more complex data structures, e.g. trees resp. terms of a free algebra
  - Intention: The already considered word-based language classes should fit into the new tree automata model in a natural way
  - Preserve the characteristic properties of the word-based automata (e.g. efficient membership problem, EPP, CPP)
- Establish new relationships and characterizations to the well-known classes of tree automata (e.g. FTAs, TPDAs, ATAs)
- Develop new applications for such restarting automata, e.g. verification of security protocols and tree transformations (XSLT)
- **Warning**: The ideas presented in this talk are work in progress

- $\mathcal{F}$  Finite set of **symbols** (ranked alphabet),  
associated with an arity function  $\text{Rnk} : \mathcal{F} \rightarrow \mathbb{N}_0$
- $\mathcal{F}_n$  Set of symbols with **arity  $n$** ,  $\mathcal{F}_n = \{f \in \mathcal{F} \mid \text{Rnk}(f) = n\}$
- $\mathcal{F}_0$  set of **constants**; simply written as  $a, b, c$
- $\mathcal{F}_n$   $n$ -ary symbols; simply written as  $f(\cdot, \cdot), g(\cdot), h(\cdot)$
- $\mathcal{X}$  Countable set of **variables** such that  $\mathcal{F}_0 \cap \mathcal{X} = \emptyset$
- $\mathcal{T}(\mathcal{F}, \mathcal{X})$  Set of **terms**; smallest set inductively defined by
- $\mathcal{F}_0 \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$  and  $\mathcal{X} \subseteq \mathcal{T}(\mathcal{F}, \mathcal{X})$
  - If  $n \geq 1$ ,  $f \in \mathcal{F}_n$  and  $t_1, \dots, t_n \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ , then  $f(t_1, \dots, t_n) \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ .
- $\text{Var}(t)$  Set of variables that occur in a term  $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$
- $\mathcal{T}(\mathcal{F})$  Set of **ground terms** (terms without variables),  
 $\mathcal{T}(\mathcal{F}) = \{t \in \mathcal{T}(\mathcal{F}, \mathcal{X}) \mid \text{Var}(t) = \emptyset\}$

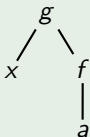
$t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$  is **linear**, if each variable from  $\mathcal{X}$  occurs at most once in  $t$ .

Example ( $\mathcal{F} = \{a, f(\cdot), g(\cdot, \cdot)\}$ ,  $\mathcal{X} = \{x, y\}$ )

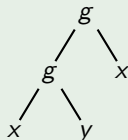
$f(f(a))$  [linear]



$g(x, f(a))$  [linear]



$g(g(x, y), x)$  [not linear]



A **position** is a word from  $\mathbb{N}^*$  (The  $\epsilon$  denotes the top-most position).

$\text{Pos}(t)$  is the set of **positions** of a term  $t$ . It is inductively defined by

- $\text{Pos}(t) = \{\epsilon\}$ , if  $t \in \mathcal{X}$  (variables) or  $t \in \mathcal{F}_0$  (constants)
- $\text{Pos}(f(t_1, \dots, t_n)) = \{\epsilon\} \cup \{ip \mid 1 \leq i \leq n \text{ and } p \in \text{Pos}(t_i)\}$

$t|_p$  denotes the **subterm** of  $t$  at the position  $p \in \text{Pos}(t)$ .

$t[s]_p$  is the term obtained by replacing in  $t$  the subterm  $t|_p$  by  $s$ .

$\text{Top}(t)$  denotes the symbol on the top-most position of  $t$ .

$\|t\|$ ,  $\text{Hgt}(t)$  The **size** and the **height** of  $t$  are inductively defined by

$$\|t\| = 0 \qquad \text{Hgt}(t) = 0 \qquad \text{if } t \in \mathcal{X},$$

$$\|t\| = 1 \qquad \text{Hgt}(t) = 1 \qquad \text{if } t \in \mathcal{F}_0,$$

$$\|t\| = 1 + \sum_{i=1}^n \|(t|_i)\| \qquad \text{if } \text{Top}(t) \in \mathcal{F}_n,$$

$$\text{Hgt}(t) = 1 + \max_{i=1}^n \text{Hgt}(t|_i) \qquad \text{if } \text{Top}(t) \in \mathcal{F}_n.$$

## Definition

Let  $\mathcal{X}_m \subseteq \mathcal{X}$  be a set of  $m \geq 1$  variables. A linear term  $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}_m)$  where  $\text{Var}(t) = \mathcal{X}_m$  is called **context** and the expression  $t[t_1, \dots, t_m]$  denotes the term which is obtained from  $t$  by replacing each variable  $x_i \in \mathcal{X}_m$  by  $t_i \in \mathcal{T}(\mathcal{F}, \mathcal{X}_m)$  for all  $1 \leq i \leq m$ .

## Definition

A **substitution** is a function  $\sigma : \mathcal{X} \rightarrow \mathcal{T}(\mathcal{F}, \mathcal{X})$ , which can be uniquely extended to an endomorphism  $\sigma : \mathcal{T}(\mathcal{F}, \mathcal{X}) \rightarrow \mathcal{T}(\mathcal{F}, \mathcal{X})$ . A **ground substitution** is a corresponding mapping  $\sigma : \mathcal{T}(\mathcal{F}, \mathcal{X}) \rightarrow \mathcal{T}(\mathcal{F})$ .

## Definition

A **term rewriting system (TRS)** is a set  $\mathcal{R}$  of rewrite rules  $l \rightarrow r$ , where  $l, r \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ ,  $l \notin \mathcal{X}$ , and  $\text{Var}(l) \supseteq \text{Var}(r)$ .

A rewrite rule is **left-linear (resp. right-linear)**, if each variable of  $l$  (resp.  $r$ ) occurs only once in  $l$  (resp.  $r$ ). A rule is **linear**, if it is left and right-linear. A TRS  $\mathcal{R}$  is **linear (resp. left-linear, right-linear)**, if every rewrite rule of  $\mathcal{R}$  is linear (resp. left-linear, right-linear).

The TRS  $\mathcal{R}$  induces a **rewriting relation**  $\rightarrow_{\mathcal{R}}$  over  $\mathcal{T}(\mathcal{F}, \mathcal{X})$ :

$$t \rightarrow_{\mathcal{R}} t' \Leftrightarrow \exists l \rightarrow r \in \mathcal{R}, \exists p \in \text{Pos}(t), \exists \sigma \text{ such that } t|_p = \sigma(l) \text{ and } t' = t[\sigma(r)]_p$$

The reflexive transitive closure of  $\rightarrow_{\mathcal{R}}$  is denoted by  $\rightarrow_{\mathcal{R}}^*$ .

A TRS  $\mathcal{R}$  is **terminating**, if there is no infinite reduction chain

$$t_0 \rightarrow_{\mathcal{R}} t_1 \rightarrow_{\mathcal{R}} t_2 \rightarrow_{\mathcal{R}} \cdots \quad \text{for any } t_0, t_1, t_2, \dots \in \mathcal{T}(\mathcal{F}, \mathcal{X}).$$

$\mathcal{Q}$  is a finite set **states** (unary symbols) such that  $\mathcal{Q} \cap \mathcal{F} = \emptyset$ .

$\mathcal{T}(\mathcal{F} \cup \mathcal{Q})$  is the set of **configurations** of a tree automata.

### Definition

A **transition** is a linear rewrite rule  $t \rightarrow t'$  where  $t, t' \in \mathcal{T}(\mathcal{F} \cup \mathcal{Q}, \mathcal{X})$ .

A **normalized top-down transition** is a transition

$$q(f(x_1, \dots, x_n)) \rightarrow f(q_1(x_1), \dots, q_n(x_n)),$$

where  $n \geq 1$ ,  $f \in \mathcal{F}_n$ ,  $x_1, \dots, x_n \in \mathcal{X}$ ,  $q, q_1, \dots, q_n \in \mathcal{Q}$ .

A **leaf transition** is a transition  $q(a) \rightarrow a$  where  $a \in \mathcal{F}_0$  and  $q \in \mathcal{Q}$ .

## Definition (top-down NFTA)

A **top-down nondeterministic finite tree automaton (NFTA)** is a four-tuple  $\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_0, \Delta)$ , where

- $\mathcal{F}$  is a finite ranked alphabet,  $\mathcal{Q}$  is a finite set of states,
- $\mathcal{Q}_0 \subseteq \mathcal{Q}$  is a set of initial states, and
- $\Delta$  is a set of normalized top-down and leaf transitions.

The **move relation** of  $\mathcal{A}$  induced by the TRS  $\Delta$  is denoted by  $\rightarrow_{\mathcal{A}}$ .

The reflexive and transitive closure of  $\rightarrow_{\mathcal{A}}$  is denoted by  $\rightarrow_{\mathcal{A}}^*$ .

The **tree language recognized by  $\mathcal{A}$**  is

$$L(\mathcal{A}) = \{t \in \mathcal{T}(\mathcal{F}) \mid \exists q \in \mathcal{Q}_0 : q(t) \rightarrow_{\mathcal{A}}^* t\}.$$

A finite tree automaton is **deterministic (DFTA)**, if  $\mathcal{Q}_0$  is a singleton and there are no two transitions in  $\Delta$  with the same left-hand side.

A tree language is called **regular**, if it can be recognized by a NFTA.

Example  $(\mathcal{A} = (\mathcal{F}, \mathcal{Q}, \mathcal{Q}_0, \Delta), \mathcal{F} = \{f(\cdot), g(\cdot), a\}, \mathcal{Q} = \{q_0, q_1\}, \mathcal{Q}_0 = \{q_0\}, \Delta = \{q_0(f(x_1)) \rightarrow f(q_0(x_1)), q_0(g(x_1)) \rightarrow g(q_1(x_1)), q_1(a) \rightarrow a\})$

- Obviously,  $\mathcal{A}$  is a DFTA and all transitions in  $\Delta$  are normalized.
- The input  $f(f(g(a))) \in \mathcal{T}(\mathcal{F})$  can be recognized as follows:

$$\begin{aligned}
 q_0(f(f(g(a)))) &\xrightarrow{\mathcal{A}}^{(1)} f(q_0(f(g(a)))) \xrightarrow{\mathcal{A}}^{(1)} f(f(q_0(g(a)))) \\
 f(f(q_0(g(a)))) &\xrightarrow{\mathcal{A}}^{(2)} f(f(g(q_1(a)))) \xrightarrow{\mathcal{A}}^{(3)} f(f(g(a)))
 \end{aligned}$$

- The regular tree language recognized by  $\mathcal{A}$  is

$$L(\mathcal{A}) = \{g(a), f(g(a)), f(f(g(a))), \dots\} = \{f^*(g(a))\}.$$

## Well-known facts about FTAs (cf. Comon et al. TATA book)

- Only in top-down case: DFTAs are less powerful than NFTAs.

$$\{f(a, b), f(b, a)\} \in \mathcal{L}(\text{NFTA}) \setminus \mathcal{L}(\text{DFTA})$$

- The class of regular tree languages, i.e.  $\mathcal{L}(\text{NFTA})$ , is closed under
  - union, intersection, and complement,
  - linear tree homomorphisms and inverse tree homomorphisms.
- The uniform membership problem is decidable in polynomial time.
- The emptiness (linear time), finiteness (polynomial time), and equivalence (EXPTIME-complete for NFTA) are decidable.

## Definition

A **top-down restarting tree automaton (RRWWTA)** is formally described by a six-tuple  $\mathcal{A} = (\mathcal{F}, \mathcal{G}, Q, q_0, k, \Delta)$ , where

- $\mathcal{F}$  is a ranked input alphabet,  $\mathcal{G} \supseteq \mathcal{F}$  is a finite ranked working alphabet,
- $Q = Q_1 \cup Q_2$  is a finite set of states such that  $Q_1 \cap Q_2 = \emptyset$ ,
- $q_0 \in Q_1$  is the initial and restart state,
- $k \geq 1$  is the height of the read/write window, and
- $\Delta = \Delta_1 \cup \Delta_2$  is a “restricted” term rewriting system on  $\mathcal{G} \cup Q$ .

A **configuration** is a term from  $\mathcal{T}(\mathcal{G} \cup Q)$ . The **general move relation**  $\rightarrow_{\Delta}$  and its reflexive transitive closure  $\rightarrow_{\Delta}^*$  are induced by the TRS  $\Delta$ . The **accepting move relation**  $\rightarrow_{\Delta_1}$  and its reflexive transitive closure  $\rightarrow_{\Delta_1}^*$  are induced by the TRS  $\Delta_1$ . The **tree language** recognized by  $\mathcal{A}$  is

$$L(\mathcal{A}) = \left\{ t_0 \in \mathcal{T}(\mathcal{F}) \mid \begin{array}{l} \exists \ell \geq 1, \exists t_1, \dots, t_{\ell} \in \mathcal{T}(\mathcal{G}) \text{ such that} \\ q_0(t_0) \rightarrow_{\Delta}^* t_1, \dots, q_0(t_{\ell-1}) \rightarrow_{\Delta_1}^* t_{\ell} \end{array} \right\}.$$

An automaton  $\mathcal{A}$  is said to be **deterministic (det-RRWWTA)**, if there are no critical pairs between rewrite rules from  $\Delta$ , i.e.  $\text{CP}(\Delta) = \emptyset$ .

The first rule set  $\Delta_1$  contains only:

- 1 **Normalized top-down transitions** of the form

$$q(f(x_1, \dots, x_n)) \rightarrow f(q_1(x_1), \dots, q_n(x_n)),$$

where  $n \geq 1$ ,  $f \in \mathcal{G}_n$ ,  $x_1, \dots, x_n \in \mathcal{X}$ , and  $q, q_1, \dots, q_n \in \mathcal{Q}_1$ .

- 2 **Leaf transitions** of the form  $q(a) \rightarrow a$ , where  $a \in \mathcal{G}_0$  and  $q \in \mathcal{Q}_1$ .

The second rule set  $\Delta_2$  contains only following transitions:

- 1 **Normalized top-down transitions** of the form

$$q(f(x_1, \dots, x_n)) \rightarrow f(q_1(x_1), \dots, q_n(x_n)),$$

where  $n \geq 1$ ,  $f \in \mathcal{G}_n$ ,  $x_1, \dots, x_n \in \mathcal{X}$ , and  $q, q_1, \dots, q_n \in \mathcal{Q}_2$ .

- 2 **Leaf transitions** of the form  $q(a) \rightarrow a$ , where  $a \in \mathcal{G}_0$  and  $q \in \mathcal{Q}_2$ .

### 3 Height-reducing rewrite transitions, i.e. (linear) rewrite rules

$$q(t) \rightarrow t'[q_1(x_1), \dots, q_m(x_m)],$$

where  $m \geq 0$ ,  $t \in \mathcal{T}(\mathcal{F}, \mathcal{X}_m)$  is a term,  $t' \in \mathcal{T}(\mathcal{F}, \mathcal{X}_m)$  is a context,  $q \in \mathcal{Q}_1$  is a state of the first partition of  $\mathcal{Q}$ , and  $q_1, \dots, q_m \in \mathcal{Q}_2$  are states of the second partition of  $\mathcal{Q}$ .

- It is required that such rules are height-reducing and **upper-bounded by the height of the read/write window**, i.e.  $\text{Hgt}(t') < \text{Hgt}(t) \leq k$ .
- Such transitions trigger a restart of the automaton, if the resulting configuration  $c$  does not contain any states (synchronization), i.e. each computation branch was successfully finished with a leaf transition. Restarting the automaton means to continue with  $q_0(c)$ .

Note that all top-down transitions shift states always from outer to inner positions. Further, the rewrite and leaf transitions are height-reducing. Thus the corresponding move relations yield no infinite reduction chains.

### Fact

*For any RRWWTA-automaton  $\mathcal{A} = (\mathcal{F}, \mathcal{G}, \mathcal{Q}, q_0, k, \Delta)$  the corresponding term rewriting system  $\Delta$  is terminating.*

Another property follows immediately by the finiteness of  $\mathcal{F}$ ,  $\mathcal{G}$ , and  $\mathcal{Q}$ .

### Fact

*The number of cycles performed by any RRWWTA during a computation on a given input tree  $t \in \mathcal{T}(\mathcal{F})$  is linearly bounded by  $\text{Hgt}(t)$ .*

The automata model does not permit propagation of state information between distinct cycles of a computation, except by using auxiliary symbols from the set  $\mathcal{G} \setminus \mathcal{F}$ .

Hence we have the well-known properties of restarting automata.

### Fact (Error Preserving Property)

*Let  $\mathcal{A} = (\mathcal{F}, \mathcal{G}, \mathcal{Q}, q_0, k, \Delta)$  be a RRWTA, and let  $u, v \in \mathcal{T}(\mathcal{F})$ . If  $q_0(u) \rightarrow_{\Delta}^* v$  holds and  $u \notin L(\mathcal{A})$ , then  $v \notin L(\mathcal{A})$ .*

### Fact (Correctness Preserving Property)

*Let  $\mathcal{A} = (\mathcal{F}, \mathcal{G}, \mathcal{Q}, q_0, k, \Delta)$  be a RRWTA, and let  $u, v \in \mathcal{T}(\mathcal{F})$ . If  $q_0(u) \rightarrow_{\Delta}^* v$  is part of an accepting computation of  $\mathcal{A}$ , then  $v \in L(\mathcal{A})$ .*

## Example

Let  $\mathcal{F} = \mathcal{G} = \{f(\cdot, \cdot), g(\cdot), a\}$  be the ranked input resp. working alphabet,  $\mathcal{Q} = \mathcal{Q}_1 \cup \mathcal{Q}_2$  the finite set of states with a partition into  $\mathcal{Q}_1 = \{q_0, q_1\}$  and  $\mathcal{Q}_2 = \{q_2\}$ , and  $k = 2$  the height of the read/write window of  $\mathcal{A}_1$ .

The general TRS  $\Delta$  is given by the following linear rewrite rules:

$$\Delta_1 : \quad q_0(f(x_1, x_2)) \rightarrow f(q_1(x_1), q_1(x_2)) \quad q_1(a) \rightarrow a$$

$$\Delta_2 : \quad q_0(f(g(x_1), g(x_2))) \rightarrow f(q_2(x_1), q_2(x_2)) \\ q_2(g(x_1)) \rightarrow g(q_2(x_1)) \quad q_2(a) \rightarrow a$$

The nondeterministic RRWWTA-automaton  $\mathcal{A}_1 = (\mathcal{F}, \mathcal{G}, \mathcal{Q}, q_0, k, \Delta)$  recognizes the tree language  $L_1 = \{f(g^n(a), g^n(a)) \mid n \geq 0\} \notin \mathcal{L}(\text{NFTA})$ .

## Example

The finite tree language  $L_2 = \{f(a, b), f(b, a)\} \notin \mathcal{L}(\text{DFTA})$  can be recognized by the det-RRWTA-automaton  $\mathcal{A}_2 = (\mathcal{F}, \mathcal{G}, \mathcal{Q}, q_0, k, \Delta)$ , where  $\mathcal{F} = \{f(\cdot, \cdot), a, b\}$ ,  $\mathcal{G} = \{\hat{a}, \hat{b}\}$ ,  $\mathcal{Q} = \{q_0\}$ ,  $k = 2$ , and the TRS  $\Delta$  is given by the following rewrite rules:

$$\begin{array}{ll} \Delta_1 : & q_0(\hat{a}) \rightarrow \hat{a} \qquad q_0(\hat{b}) \rightarrow \hat{b} \\ \Delta_2 : & q_0(f(a, b)) \rightarrow \hat{a} \qquad q_0(f(b, a)) \rightarrow \hat{b} \end{array}$$

Hence we have the following inclusions:

$$\begin{array}{ccc} \mathcal{L}(\text{DFTA}) & \subsetneq & \mathcal{L}(\text{det-RRWTA}) \\ \uparrow \cap & & \uparrow \cap \\ \mathcal{L}(\text{NFTA}) & \subsetneq & \mathcal{L}(\text{RRWTA}) \end{array}$$

Currently, there are many **open questions** to solve:

- 1 Pumping Lemma
- 2 Expressive Power, e.g. (top-)context-free tree languages
- 3 Closure Properties

- Intersection: RRWWTAs can recognize the languages

$$L_1 = \{f(g^n(h^m(a)), g^n(h^p(a))) \mid n, m, p \geq 1\},$$

$$L_2 = \{f(g^m(h^n(a)), g^p(h^n(a))) \mid n, m, p \geq 1\},$$




$$L_3 = \{f(g^n(h^m(a)), g^p(h^n(a))) \mid n, m, p \geq 1\}.$$

But the intersection

$$\bigcap_{i=1,2,3} L_i = \{f(g^n(h^n(a)), g^n(h^n(a))) \mid n \geq 1\}$$

is not a context-free tree language.

- 4 Decision Problems
- 5 Yield Languages and Path Languages

-  Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Sophie Tison, Marc Tommasi.  
Tree Automata Techniques and Applications.  
<http://www.grappa.univ-lille3.fr/tata/>
-  Friedrich Otto.  
Restarting Automata and their Relations to the Chomsky Hierarchy.  
Proceedings DLT 2003, LNCS 2710, pp. 55–74, 2003.
-  Friedrich Otto.  
Restarting Automata.  
Notes for a Course at the 3rd International PhD School in Formal Languages and Applications, 2004.