On

Positive TAGED

with a

Bounded Number of Constraints

P.-C. Héam, Vincent Hugot, O. Kouchnarenko

{pheam,vhugot,okouchna}@femto-st.fr

University of Franche-Comté

DGA & INRIA/CASSIS & FEMTO-ST (DISC)

July 2, 2013

Tree Automata

[Comon et al., 2008a]

Introduced in the fifties; **regular tree languages**:

- model-checking: programs, protocols,. . .
- automated theorem-proving
- XML schema and (esp. variants) query languages
- . . . and so much more

Doesn't deal with **comparisons** and **non-linearity**:

- $\{ f(u, u) \mid u \in \mathcal{T}(\Sigma) \}$ e.g. password verification
- $\{ f(u, v) \mid u, v \in \mathcal{T}(\Sigma), u \neq v \}$ e.g. primary keys
- $\mathcal{R}(\ell)$, ℓ regular, $\mathcal{R}$ a TRS e.g. $\{ g(x) \to f(x, x) \}(\mathcal{T}(\mathbb{A}))$

Tree Automata

Bottom-Up, Non-Deterministic, Finite

Tree Automaton $\mathcal{A} = \langle \mathbb{A}, Q, F, \Delta \rangle$:

$\mathbb{A}$	finite **ranked alphabet**
Q	finite set of **states**
F	**final** states, $F \subseteq Q$
Δ	finite set of **transitions**

Transition $r \in \Delta$:

$$\sigma(q_1, \dots, q_n) \to q \qquad \sigma \in \mathbb{A}_n \quad q_1, \dots, q_n, q \in Q$$

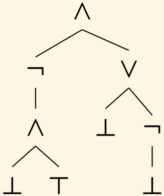
Tree Automata

Bottom-Up, Non-Deterministic, Finite

$\mathbb{A} = \{\, \wedge, \vee/2, \neg/1, \top, \bot/0 \,\}$, $Q = \{\, q_0, q_1 \,\}$, $F = \{q_1\}$, $\Delta =$

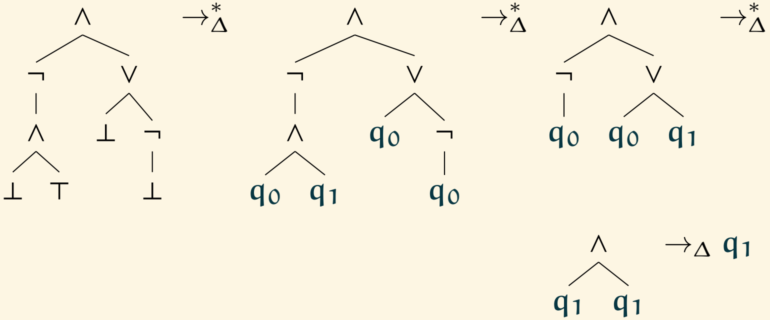
$$\left\{ \begin{array}{l} \top \to q_1, \quad \bot \to q_0, \quad \neg(q_b) \to q_{\neg b} \\ \wedge(q_b, q_{b'}) \to q_{b \wedge b'}, \quad \vee(q_b, q_{b'}) \to q_{b \vee b'} \end{array} \;\middle|\; b, b' \in \{\, 0, 1 \,\} \right\}$$

$t =$



Tree Automata

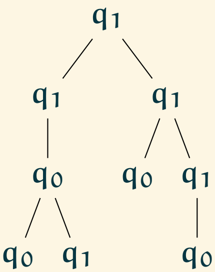
Bottom-Up, Non-Deterministic, Finite



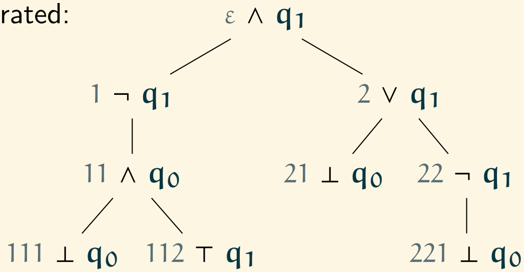
Tree Automata

Runs and Languages

The reduction $t \to_{\Delta}^{*} q_1$ is captured by the **run**:



decorated:



Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

TAGE, $TA^=$, Positive TAGED, $\mathcal{A} = \langle \mathbb{A}, Q, F, \Delta, \approxeq \rangle$:

$\langle \mathbb{A}, Q, F, \Delta \rangle$	vanilla **tree automaton** $ta(\mathcal{A})$
$\approxeq$	equality **constraints**, $\approxeq \subseteq Q^2$

Constraint $p \approxeq q$:

run ρ of $\mathcal{A}$ on t:

- **run** of $ta(\mathcal{A})$ on t
- **satisfying** $\approxeq$: $\forall \alpha, \beta \in \mathcal{P}(t);\ \rho(\alpha) \approxeq \rho(\beta) \Rightarrow t|_\alpha = t|_\beta$

accepting run: accepting for $ta(\mathcal{A})$

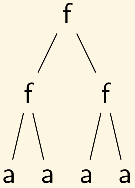
Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

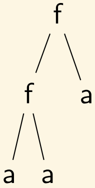
$\mathbb{A} = \{ a/_0, f/_2 \}$, $Q = \{ q, \hat{q}, q_f \}$, $F = \{ q_f \}$, $\hat{q} \approxeq \hat{q}$, and

$$\Delta = \{ f(\hat{q}, \hat{q}) \to q_f,\ f(q,q) \to q,\ f(q,q) \to \hat{q},\ a \to q,\ a \to \hat{q} \}$$

$u =$



and $v =$

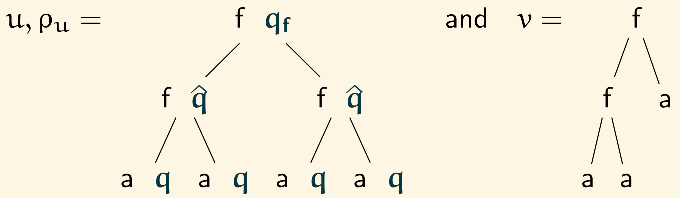


Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

$\mathbb{A} = \{ a/_0, f/_2 \}$, $Q = \{ q, \hat{q}, q_f \}$, $F = \{ q_f \}$, $\hat{q} \cong \hat{q}$, and

$$\Delta = \{ f(\hat{q}, \hat{q}) \to q_f,\ f(q,q) \to q,\ f(q,q) \to \hat{q},\ a \to q,\ a \to \hat{q} \}$$

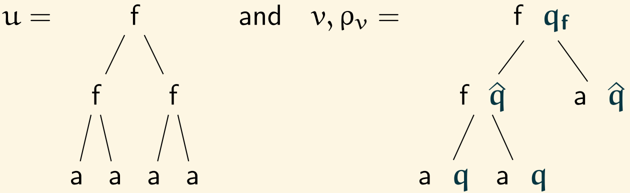


Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

$\mathbb{A} = \{ a/_0, f/_2 \}$, $Q = \{ q, \hat{q}, q_f \}$, $F = \{ q_f \}$, $\hat{q} \approxeq \hat{q}$, and

$$\Delta = \{ f(\hat{q}, \hat{q}) \to q_f,\ f(q,q) \to q,\ f(q,q) \to \hat{q},\ a \to q,\ a \to \hat{q} \}$$

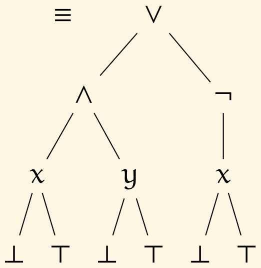


Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

$\mathbb{A} = \{\wedge, \vee/_2, \neg/_1, \top, \bot/_0\} \uplus \mathbb{X}$, $Q = \{q_0, q_1\} \uplus \{v_x \mid x \in \mathbb{X}\}$ and $F = \{q_1\}$, new rules $\top \to v_x$, $\bot \to v_x$, $x(q_0, v_x) \to q_1$, $x(v_x, q_1) \to q_0$ for each $x \in \mathbb{X}$, $v_x \approxeq v_x$.

$(x \wedge y) \vee \neg x \quad \equiv$

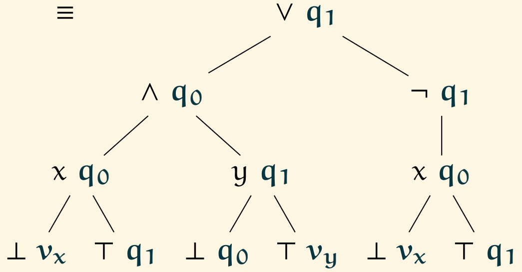


Tree Automata

With Global Equality Constraints [Filiot et al., 2008]

$\mathbb{A} = \{ \wedge, \vee/2, \neg/1, \top, \bot/0 \} \uplus \mathbb{X}$, $Q = \{ q_0, q_1 \} \uplus \{ v_x \mid x \in \mathbb{X} \}$ and $F = \{q_1\}$, new rules $\top \to v_x$, $\bot \to v_x$, $x(q_0, v_x) \to q_1$, $x(v_x, q_1) \to q_0$ for each $x \in \mathbb{X}$, $v_x \approxeq v_x$.

$(x \wedge y) \vee \neg x \quad \equiv$



TA versus RTA versus $TA^=$

Closure, Complexity and Decidability

	TA	RTA ($p \approx p$)	$TA^=$
$\cup$	PTime	PTime	PTime
$\cap$	PTime	ExpTime	ExpTime
$\neg$	ExpTime	$\varnothing$	$\varnothing$
$t \in \mathcal{L}(\mathcal{A})$?	PTime	NP-c	NP-c [a]
$\mathcal{L}(\mathcal{A}) = \varnothing$?	linear-time	linear-time	ExpTime-c
$\lvert\mathcal{L}(\mathcal{A})\rvert \in \mathbb{N}$?	PTime	PTime	ExpTime-c
$\mathcal{L}(\mathcal{A}) = \mathcal{T}(\Sigma)$?	ExpTime-c	*undecidable*	*undecidable*
$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?	ExpTime-c	*undecidable*	*undecidable*
$\mathcal{L}(\bigcap_i \mathcal{A}_i) = \varnothing$?	ExpTime-c	ExpTime-c	ExpTime-c

[a] SAT solver approach: [Héam et al., 2010].

TA versus RTA versus $TA^=$

Closure, Complexity and Decidability

	TA	**RTA** $(p \approx p)$	$\mathbf{TA^=}$
$\cup$	PTIME	PTIME	PTIME
$\cap$	PTIME	EXPTIME	EXPTIME
$\neg$	EXPTIME	$\varnothing$	$\varnothing$
$t \in \mathcal{L}(\mathcal{A})$?	PTIME	NP-c	NP-c [(a)]
$\mathcal{L}(\mathcal{A}) = \varnothing$?	linear-time	**linear-time**	**ExpTime-c**
$\lvert\mathcal{L}(\mathcal{A})\rvert \in \mathbb{N}$?	PTIME	**PTime**	**ExpTime-c**
$\mathcal{L}(\mathcal{A}) = \mathcal{T}(\Sigma)$?	EXPTIME-c	*undecidable*	*undecidable*
$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?	EXPTIME-c	*undecidable*	*undecidable*
$\mathcal{L}(\bigcap_i \mathcal{A}_i) = \varnothing$?	EXPTIME-c	EXPTIME-c	EXPTIME-c

(a) SAT solver approach: [Héam et al., 2010].

$TA^=$ versus $TA_k^=$

Restriction on the **kind** of constraints ⇒ **lower complexity** (RTA)

Restriction on the **number** of constraints ⇒ **?**

$TA_k^=$ $\mathcal{A} = \langle \Sigma, Q, F, \Delta, \approxeq \rangle$:

$\langle \Sigma, Q, F, \Delta, \approxeq \rangle$	$TA^=$ $\mathcal{A}$
$\approxeq$	such that $\mathrm{Card}(\approxeq) \leqslant k$

$\mathsf{TA}^{=}$ versus $\mathsf{TA}^{=}_{k}$

Restriction on the **kind** of constraints $\Rightarrow$ **lower complexity** (RTA)

Restriction on the **number** of constraints $\Rightarrow$ **?**

$\mathsf{TA}^{=}_{k}$ $\mathcal{A} = \langle \Sigma, Q, F, \Delta, \approxeq \rangle$:

$\langle \Sigma, Q, F, \Delta, \approxeq \rangle$	$\mathsf{TA}^{=}$ $\mathcal{A}$
$\approxeq$	such that $\operatorname{Card}(\approxeq) \leqslant k$

Expressive power?

$$\mathsf{TA} = \mathsf{TA}^{=}_{0} \subset \mathsf{TA}^{=}_{1} \subset \cdots \subset \mathsf{TA}^{=}_{k} \subset \mathsf{TA}^{=}_{k+1} \subset \cdots \subset \mathsf{TA}^{=} = \bigcup_{k \in \mathbb{N}} \mathsf{TA}^{=}_{k}$$

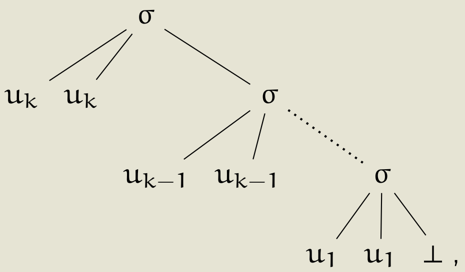
so $\forall k \geqslant 0,\ \mathcal{L}(\mathsf{TA}^{=}_{k}) \subseteq \mathcal{L}\left(\mathsf{TA}^{=}_{k+1}\right)$. Are the inclusions **strict**? Up to some rank? Is there a $k \in \mathbb{N}$ such that $\mathcal{L}(\mathsf{TA}^{=}_{k}) = \mathcal{L}(\mathsf{TA}^{=})$?

Expressive Power

The Separation Languages $L = (\ell_k)_{k \in \mathbb{N}}$ [Hugot, 2013]

$$\biguplus_{i=1}^{k} \mathbb{A}_i \uplus \{\, \sigma/3, \bot/0 \,\} \qquad \mathbb{A}_i = \{\, a_i, b_i/0, f_i, g_i/2 \,\}$$

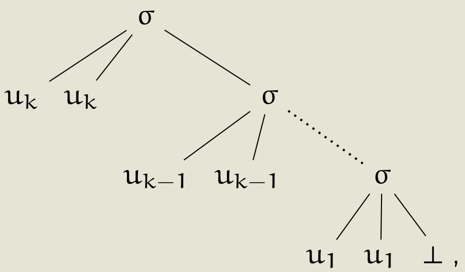
$$\ell_0 = \{\bot\} \quad \forall k \geqslant 1, \ell_k = \{\, \sigma(u, u, t_{k-1}) \mid u \in \mathcal{T}(\mathbb{A}_k), t_{k-1} \in \ell_{k-1} \,\}$$



$u_i \in \mathcal{T}(\mathbb{A}_i)\ \forall i$

Expressive Power

The Separation Languages $L = (\ell_k)_{k \in \mathbb{N}}$ [Hugot, 2013]



$u_i \in \mathcal{T}(\mathbb{A}_i)\ \forall i$

$\ell_1 \in \mathcal{L}(\mathsf{TA}_1^{=}) \setminus \mathcal{L}(\mathsf{TA})$ $\approx$ ground instances of $f(x,x)$.

$\ell_k \in \mathcal{L}(\mathsf{TA}_k^{=}) \setminus \mathcal{L}(\mathsf{TA}_{k-1}^{=}), \quad \forall k \geqslant 1.$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathsf{TA}_k^{\equiv}) \setminus \mathcal{L}(\mathsf{TA}_{k-1}^{\equiv})$ [Hugot, 2013]

Show $\ell_k \in \mathcal{L}(\mathbf{TA}_k^{\equiv})$: $\mathcal{A}_k \in \mathsf{TA}_k^{\equiv}$ such that $\mathcal{L}(\mathcal{A}_k) = \ell_k$

$\mathcal{U}_i \in \mathsf{TA}$ universal, $\mathcal{U}_i{:}F = \{q_i^u\}$, for all i. $\mathcal{A}_k$ is

$$Q = \{q_0^v\} \uplus \biguplus_{i=1}^{k} \mathcal{U}_i{:}Q \uplus \{q_i^v\} \qquad F = \{q_1^v\} \qquad q_i^u \approxeq q_i^u,\ \forall i \in [\![1,k]\!]$$

$$\Delta = \left\{ \sigma(q_i^u, q_i^u, q_{i-1}^v) \to q_i^v \;\middle|\; i \in [\![1,k]\!] \right\} \cup \{ \bot \to q_0^v \}.$$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathsf{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathsf{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

active constrained states:

$$\operatorname{acs}\rho = \{\, \rho(\alpha) \mid \alpha \in \mathcal{P}(\rho), \exists \beta \in \mathcal{P}(\rho) \setminus \{\alpha\} : \rho(\alpha) \approxeq \rho(\beta) \,\}$$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathsf{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathsf{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathsf{TA}^{\equiv}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathsf{TA}^{\equiv}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{=}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{=}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{\equiv}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left|t|_\alpha\right| > |Q|$, for all $\alpha \in 3^*(1+2)$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{\equiv}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left| t|_\alpha \right| > |Q|$, for all $\alpha \in 3^*(1+2)$
- Suppose $\exists \alpha \in 3^*(1+2)$ such that $\operatorname{ran} \rho|_\alpha \cap \operatorname{acs} \rho = \varnothing$. $\mathcal{A}$ acts as BUTA wrt. $t|_\alpha$; pump $\rho|_\alpha$, get $t' \notin \ell_k$, but $t' \in \mathcal{L}(\mathcal{A})$.

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{=}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{=}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left| t|_\alpha \right| > |Q|$, for all $\alpha \in 3^*(1+2)$
- $\forall \alpha \in 3^*(1+2),\ \operatorname{ran} \rho|_\alpha \cap \operatorname{acs} \rho \neq \varnothing$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathsf{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathsf{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathsf{TA}^{\equiv}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathsf{TA}^{\equiv}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left| t|_\alpha \right| > |Q|$, for all $\alpha \in 3^*(1+2)$
- $\forall \alpha \in 3^*(1+2),\ \operatorname{ran} \rho|_\alpha \cap \operatorname{acs} \rho \neq \varnothing$
- $i \neq j$, p_i acs for u_i, p_j for u_j. $\exists$ acs $q_i, q_j : p_i \approxeq q_i, p_j \approxeq q_j$. Suppose q_i in subrun of u_j. Then $\exists s_i \trianglelefteq u_i, s_j \trianglelefteq u_j$, $s_i = s_j$. But $u_i \in \mathcal{T}(\mathbb{A}_i)$ and $u_j \in \mathcal{T}(\mathbb{A}_j)$, thus $s_i \in \mathcal{T}(\mathbb{A}_i)$ and $s_j \in \mathcal{T}(\mathbb{A}_j)$. $\mathcal{T}(\mathbb{A}_i) \cap \mathcal{T}(\mathbb{A}_j) = \varnothing$, thus $s_i = s_j \in \varnothing$.

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{\equiv}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{\equiv}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{\equiv}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho,\ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left| t|_\alpha \right| > |Q|$, for all $\alpha \in 3^*(1+2)$
- $\forall \alpha \in 3^*(1+2),\ \operatorname{ran} \rho|_\alpha \cap \operatorname{acs} \rho \neq \varnothing$
- Each pair of u_i needs its own fresh state(s) $p_i \approxeq q_i$

Expressive Power

Show $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k}) \setminus \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ [Hugot, 2013]

Show $\ell_k \notin \mathcal{L}(\mathbf{TA}^{=}_{k-1})$:

- Assume $\ell_k \in \mathcal{L}(\mathrm{TA}^{=}_{k-1})$ i.e. $\exists \mathcal{A} \in \mathrm{TA}^{=}_{k-1} : \mathcal{L}(\mathcal{A}) = \ell_k$
- $\forall \rho, \ \nexists \alpha, \beta : \alpha \neq \beta, \alpha \in 3^*, \rho(\alpha) \approxeq \rho(\beta)$
- Pick $t \in \ell_k$ such that $\left| t|_\alpha \right| > |Q|$, for all $\alpha \in 3^*(1+2)$
- $\forall \alpha \in 3^*(1+2), \ \operatorname{ran} \rho|_\alpha \cap \operatorname{acs} \rho \neq \varnothing$
- Each pair of u_i needs its own fresh state(s) $p_i \approxeq q_i$
- $\mathcal{A}$ does not exist, contradiction.

The Membership Problem

General Idea & Strategy

Membership complexity : $t \in \mathcal{L}(\mathcal{A})$?

NP-complete	for	$TA^{=}$
PTime	for	$TA^{=}_{k}, \forall k \in \mathbb{N}$

Proof **Strategy** :

- Choose each $P \subseteq \mathrm{dom} \approxeq = \{ p \mid \exists q : p \approxeq q \text{ or } q \approxeq p \}$
- Given P, turn $\approxeq$ into an equivalence relation $\asymp_P$
- Try all possible "housings" of the $\approxeq$-classes into t
- For each housing, try to build an accepting run

$\approxeq$ is Not an Equivalence

(but we can pretend it is)

Example: Given $p \approxeq r$ and $r \approxeq q$, what of $p \approxeq q$?

Does r actually appear in the run ?

yes : $p \approxeq q$ implied

no : $p \approxeq r$ and $r \approxeq q$ are moot.

Fix $P \subseteq \mathrm{dom} \approxeq$. Any run ρ such that $(\mathbf{ran}\,\rho) \cap (\mathbf{dom} \approxeq) = \mathbf{P}$ is accepting for $\mathcal{A}$ iff it is so for

$$\mathcal{A}_P = \wr \mathcal{A} \mid \approxeq := \left(\approxeq \cap P^2\right)^{\equiv} \int ,$$

symmetric, transitive, reflexive closure under $\mathrm{dom}(\approxeq \cap P^2)$.

Groups & Similarity Classes

Groups $\mathbb{G}_P$: set of $\approxeq$-equivalence classes (given P)

$$\mathbb{G}_P = \frac{\mathrm{dom}(\approxeq \cap P^2)}{(\approxeq \cap P^2)^{\equiv}} = \frac{\mathrm{dom}(\approxeq \cap P^2)}{\asymp_P}$$

Similarity **Classes** $\mathbb{S}_t$ of t :

$$\forall \alpha, \beta \in \mathcal{P}(t);\ \alpha \sim \beta \iff t|_\alpha = t|_\beta$$

$$\textbf{classes } \mathbb{S}_t = \mathcal{P}(t)/_\sim$$

Housings

And Their Compatibility with the Constraints

Characterisation of **Satisfaction** of $\approxeq$:

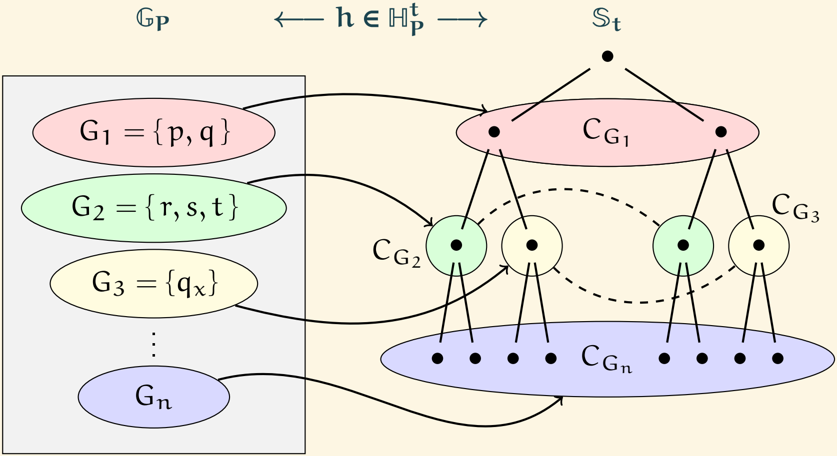
$$\forall G \in \mathbb{G}_P;\ \exists C_G \in \mathbb{S}_t : \rho^{-1}(G) \subseteq C_G$$

Housings $\mathbb{H}_P^t$ of P within t :

The map $G \mapsto C_G$ is a **P-housing of ρ in t**, **compatible** with ρ

$$\mathbb{H}_P^t = \mathbb{G}_P \to \mathbb{S}_t$$

is the set of all possible P-housings on t.



$\mathbb{G}_{\mathbf{P}}$
$\longleftarrow \mathbf{h} \in \mathbb{H}_{\mathbf{P}}^{\mathbf{t}} \longrightarrow$
$\mathbb{S}_{\mathbf{t}}$
$G_1 = \{p, q\}$
$G_2 = \{r, s, t\}$
$G_3 = \{q_x\}$
$\vdots$
G_n
C_{G_1}
C_{G_2}
C_{G_3}
C_{G_n}

Proof Outline

For $TA_k^=$

Operations Needed :

- **Choose P:** 2^{2k} possible $P \subseteq \mathrm{dom} \approxeq$
- **Choose housing:** $\left|\mathbb{S}_t{}^{\mathbb{G}_P}\right| = |\mathbb{S}_t|^{|\mathbb{G}_P|} \leqslant \|t\|^{2k}$ P-housings on t
- $\Rightarrow 4^k \cdot \|t\|^{2k}$ tests in total

↬ **polynomial** compatibility test = variant of **reachability**

Is a final state reachable if states $q \in P$ can only go in $h([q]_{\asymp_P})$?

Compatibility Test

In Polynomial Time

Simple variant of **reachability** algorithm:

Given P and $h \in \mathbb{H}_P^t$, there exists a compatible run iff

$$\Phi_t^{P,h}(\varepsilon) \cap F \neq \varnothing \ ,$$

where

$$\Phi_t^{P,h}(\alpha) = \left\{ q \in Q \;\middle|\; \begin{array}{l} t(\alpha)(p_1, \dots, p_n) \to q \in \Delta \\ \forall i \in [\![1, n]\!],\ p_i \in \Phi_t^{P,h}(\alpha.i) \\ q \in \bigcup \mathbb{G}_P \implies \alpha \in h([q]_{\asymp_P}) \\ q \notin \mathrm{dom}(\approxeq) \setminus P \end{array} \right\} .$$

Compatibility Test

In Polynomial Time

Simple variant of **reachability** algorithm:

Given P and $h \in \mathbb{H}_P^t$, there exists a compatible run iff

$$\Phi_t^{P,h}(\varepsilon) \cap F \neq \varnothing ,$$

where

$$\Phi_t^{P,h}(\alpha) = \left\{ q \in Q \;\middle|\; \begin{array}{l} t(\alpha)(p_1, \dots, p_n) \to q \in \Delta \\ \forall i \in [\![1, n]\!],\ p_i \in \Phi_t^{P,h}(\alpha.i) \\ q \in \bigcup \mathbb{G}_P \implies \alpha \in h([q]_{\asymp_P}) \\ q \notin \mathrm{dom}(\approxeq) \setminus P \end{array} \right\} .$$

Compatibility Test

In Polynomial Time

Simple variant of **reachability** algorithm:

Given P and $h \in \mathbb{H}_P^t$, there exists a compatible run iff

$$\Phi_t^{P,h}(\varepsilon) \cap F \neq \varnothing \,,$$

where

$$\Phi_t^{P,h}(\alpha) = \left\{ q \in Q \;\middle|\; \begin{array}{l} t(\alpha)(p_1, \dots, p_n) \to q \in \Delta \\ \forall i \in [\![1, n]\!],\ p_i \in \Phi_t^{P,h}(\alpha.i) \\ q \in \bigcup \mathbb{G}_P \implies \alpha \in h([q]_{\asymp_P}) \\ q \notin \mathrm{dom}(\approxeq) \setminus P \end{array} \right\} .$$

Compatibility Test

In Polynomial Time

Simple variant of **reachability** algorithm:

Given P and $h \in \mathbb{H}_P^t$, there exists a compatible run iff

$$\Phi_t^{P,h}(\varepsilon) \cap F \neq \varnothing ,$$

where

$$\Phi_t^{P,h}(\alpha) = \left\{ q \in Q \;\middle|\; \begin{array}{l} t(\alpha)(p_1, \dots, p_n) \to q \in \Delta \\ \forall i \in [\![1, n]\!],\ p_i \in \Phi_t^{P,h}(\alpha.i) \\ q \in \bigcup \mathbb{G}_P \implies \alpha \in h([q]_{\asymp_P}) \\ q \notin \operatorname{dom}(\approxeq) \setminus P \end{array} \right\} .$$

Compatibility Test

In Polynomial Time

Simple variant of **reachability** algorithm:

Given P and $h \in \mathbb{H}_P^t$, there exists a compatible run iff

$$\Phi_t^{P,h}(\varepsilon) \cap F \neq \varnothing \ ,$$

where

$$\Phi_t^{P,h}(\alpha) = \left\{ q \in Q \;\middle|\; \begin{array}{l} t(\alpha)(p_1, \dots, p_n) \to q \in \Delta \\ \forall i \in [\![1, n]\!],\ p_i \in \Phi_t^{P,h}(\alpha.i) \\ q \in \bigcup \mathbb{G}_P \implies \alpha \in h([q]_{\asymp_P}) \\ q \notin \mathrm{dom}(\approxeq) \setminus P \end{array} \right\} .$$

Rigidification

Problem : Given TA$^{=}$ $\mathcal{A}$, build equivalent RTA $\mathcal{B}$.

General Result [Filiot, 2008, Lem. 5.3.5]

Exponential construction: $\|\mathcal{B}\| \leqslant \mathbf{O}(2^{\|\mathcal{A}\|^2})$

In the case of $\mathbf{TA}_1^{=}$:

Polynomial construction: $\|\mathcal{B}\| \leqslant \mathbf{O}(\|\mathcal{A}\|^2)$

Idea : Simulate a constraint $p \approxeq q,\ p \neq q$ by a TA intersection

Rigidification: Construction

$$\mathcal{B} = \mathcal{B}_p^\neg \uplus \mathcal{B}_q^\neg \uplus \wr \mathcal{A} \mid Q', \Delta', q_f \approxeq q_f \int$$

$$\mathcal{B}_p^\neg = \wr \mathcal{A} \mid Q \setminus \{p\} \int \qquad \mathcal{B}_q^\neg = \wr \mathcal{A} \mid Q \setminus \{q\} \int$$

$$Q' = (Q \setminus \{p, q\}) \uplus (\mathcal{B}_{pq} : Q) \qquad \Delta' = \Delta_{pq}^{q_f} \uplus (\mathcal{B}_{pq} : \Delta)$$

$$\mathcal{B}_{pq} = \mathcal{B}_p \otimes \mathcal{B}_q \qquad q_f = (p, q)$$

$$\mathcal{B}_p = \wr \mathcal{B}_q^\neg \mid F := \{p\}, \Delta := \Delta_p \int \qquad \mathcal{B}_q = \wr \mathcal{B}_p^\neg \mid F := \{q\}, \Delta := \Delta_q \int$$

$$\Delta_p = \mathcal{B}_q^\neg : \Delta \setminus \{\ldots p \ldots \to \ldots\} \qquad \Delta_q = \mathcal{B}_p^\neg : \Delta \setminus \{\ldots q \ldots \to \ldots\}$$

$\Delta_{pq}^{q_f}$ is $\mathcal{A} : \Delta$ from which all left-hand side occurrences of p or q have been replaced by q_f.

Emptiness

Outline of the Result and Proof

Complexity of **Emptiness** : $\mathcal{L}(\mathcal{A}) = \varnothing$?

PTime (quadratic)	for	$\mathrm{TA}_1^{\equiv}$
ExpTime-complete	for	$\mathrm{TA}_k^{\equiv}$, $k \geqslant 2$

$\mathrm{TA}_1^{\equiv}$: immediate by **rigidification**. Emptiness for RTA: linear time

$\mathrm{TA}_2^{\equiv}$: Reduction of **intersection-emptiness** of n TA $\mathcal{A}_1, \dots, \mathcal{A}_n$.
Generalisation of the usual argument [Filiot et al., 2008, Thm. 1] from "unlimited constraints" to **"two constraints"**

$$L = \varnothing \iff \bigcap_{i=1}^{n} \mathcal{L}(\mathcal{A}_i) = \varnothing$$

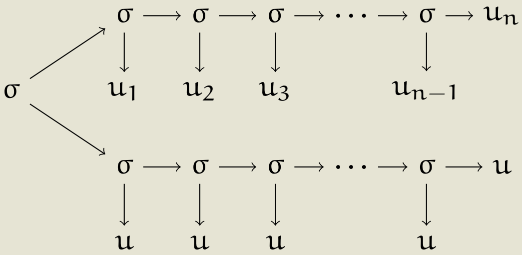


Figure: Reduction of intersection-emptiness: the language.

where $\forall i,\ x_i \in \mathcal{L}(\mathcal{A}_i)$ and $x = x_i$

Finiteness

Outline of the Result and Proof

Complexity of **Finiteness** : $|\mathcal{L}(\mathcal{A})| \in \mathbb{N}$?

PTime	for	$\mathsf{TA}_1^{\equiv}$
ExpTime-complete	for	$\mathsf{TA}_k^{\equiv}$, $k \geqslant 2$

$\mathsf{TA}_1^{\equiv}$: immediate by **rigidification**. Finiteness for RTA is PTime

$\mathsf{TA}_2^{\equiv}$: Reduction of **Emptiness for $\mathsf{TA}_2^{\equiv}$**.

Finiteness

Outline of the Result and Proof

$$\mathcal{A}' = \wr\, \mathcal{A} \mid Q \uplus \{p\}, F := \{p\}, \Sigma \uplus \{\sigma/1\}, \Delta' \int$$
$$\text{where } \Delta' = \Delta \cup \{\, \sigma(q_f) \to p \mid q_f \in F \,\} \cup \{\, \sigma(p) \to p \,\}$$

$$\text{if } \mathcal{L}(\mathcal{A}) = \varnothing \quad \textbf{then} \quad \mathcal{L}(\mathcal{A}') = \varnothing$$
$$\text{if } t \in \mathcal{L}(\mathcal{A}) \quad \textbf{then} \quad \sigma^+(t) \subseteq \mathcal{L}(\mathcal{A}')$$

$\mathcal{L}(\mathcal{A}')$ is **finite** $\iff$ $\mathcal{L}(\mathcal{A})$ is **empty**

Summary
and Perspectives

Refined **complexity** and **expressiveness** results:

- **Expressiveness:** $\mathsf{TA}_k^{\equiv}$ form a strict hierarchy
- **Membership:** NP-c for $\mathsf{TA}^{\equiv}$, but PTIME for $\mathsf{TA}_k^{\equiv}$, $\forall k$
- **Emptiness:** quadratic for $\mathsf{TA}_1^{\equiv}$, EXPTIME-complete for $\mathsf{TA}_2^{\equiv}$
- **Finiteness:** PTIME for $\mathsf{TA}_1^{\equiv}$, EXPTIME-complete for $\mathsf{TA}_2^{\equiv}$

Left **to do**:

Effects of $\not\approx$, flat constraints, efficient heuristics, etcetera.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Positional constraints:

- **TALEDC** [Mongy, 1981]: local equality and disequality constraints. $f(q_1, q_2, q_3)[13 \approx 2, 12 \not\approx 13] \to q$.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Positional constraints:

- **TALEDC** [Mongy, 1981]: local equality and disequality constraints. $f(q_1, q_2, q_3)[13 \approx 2, 12 \not\approx 13] \to q$.
- **TAPLEDC** [Comon et al., 2008a, Chap. 4, as AWEDC]: propositional extension. $C := \beta \approx \gamma \mid \beta \not\approx \gamma \mid C \wedge C \mid \neg C$.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Positional constraints:

- **TALEDC** [Mongy, 1981]: local equality and disequality constraints. $f(q_1, q_2, q_3)[13 \approx 2, 12 \not\approx 13] \to q$.
- **TAPLEDC** [Comon et al., 2008a, Chap. 4, as AWEDC]: propositional extension. $C := \beta \approx \gamma \mid \beta \not\approx \gamma \mid C \wedge C \mid \neg C$.
- **TABB** [Bogaert and Tison, 1992]: restriction to constraints between brothers: $\beta\{\approx, \not\approx\}\gamma$ such that $|\beta| = |\gamma| = 1$.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Positional constraints:

- **TALEDC** [Mongy, 1981]: local equality and disequality constraints. $f(q_1, q_2, q_3)[13 \approx 2, 12 \not\approx 13] \to q$.
- **TAPLEDC** [Comon et al., 2008a, Chap. 4, as AWEDC]: propositional extension. $C := \beta \approx \gamma \mid \beta \not\approx \gamma \mid C \wedge C \mid \neg C$.
- **TABB** [Bogaert and Tison, 1992]: restriction to constraints between brothers: $\beta\{\approx, \not\approx\}\gamma$ such that $|\beta| = |\gamma| = 1$.
- **RA** [Dauchet et al., 1995]: bounds equality constraints along any leaf-to-root path of execution.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Positional constraints:

- **TALEDC** [Mongy, 1981]: local equality and disequality constraints. $f(q_1, q_2, q_3)[13 \approx 2, 12 \not\approx 13] \to q$.
- **TAPLEDC** [Comon et al., 2008a, Chap. 4, as AWEDC]: propositional extension. $C := \beta \approx \gamma \mid \beta \not\approx \gamma \mid C \wedge C \mid \neg C$.
- **TABB** [Bogaert and Tison, 1992]: restriction to constraints between brothers: $\beta\{\approx, \not\approx\}\gamma$ such that $|\beta| = |\gamma| = 1$.
- **RA** [Dauchet et al., 1995]: bounds equality constraints along any leaf-to-root path of execution.
- **GRA** [Caron et al., 1994]: relaxes RA, but only where constraints between brothers are involved.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Global constraints:

- **TAGED** [Filiot et al., 2008, Filiot et al., 2010, Filiot, 2008].

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Global constraints:

- **TAGED** [Filiot et al., 2008, Filiot et al., 2010, Filiot, 2008].
- **RTA** [Jacquemard et al., 2009, Jacquemard et al., 2011, Vacher, 2010]: rigid states $q \approx q$.

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Global constraints:

- **TAGED** [Filiot et al., 2008, Filiot et al., 2010, Filiot, 2008].
- **RTA** [Jacquemard et al., 2009, Jacquemard et al., 2011, Vacher, 2010]: rigid states $q \approxeq q$.
- **Parick+E**, **NParick+EDB** [Barguñó et al., 2010]:

$$\sum_{q \in Q} a_q |q| \geqslant b \quad \text{or} \quad \sum_{q \in Q} a_q \|q\| \geqslant b \quad \text{with } a_q, b \in \mathbb{Z}, \forall q \in Q\ .$$

Tree Automata With Constraints

Brief survey: [Hugot, 2013]

Global constraints:

- **TAGED** [Filiot et al., 2008, Filiot et al., 2010, Filiot, 2008].
- **RTA** [Jacquemard et al., 2009, Jacquemard et al., 2011, Vacher, 2010]: rigid states $q \approxeq q$.
- **Parick+E**, **NParick+EDB** [Barguñó et al., 2010]:

$$\sum_{q \in Q} a_q |q| \geqslant b \quad \text{or} \quad \sum_{q \in Q} a_q \|q\| \geqslant b \quad \text{with } a_q, b \in \mathbb{Z}, \forall q \in Q \,.$$

- **DAGA** [Charatonik, 1999]: equivalent to TAGD.

Tree Automata With Constraints

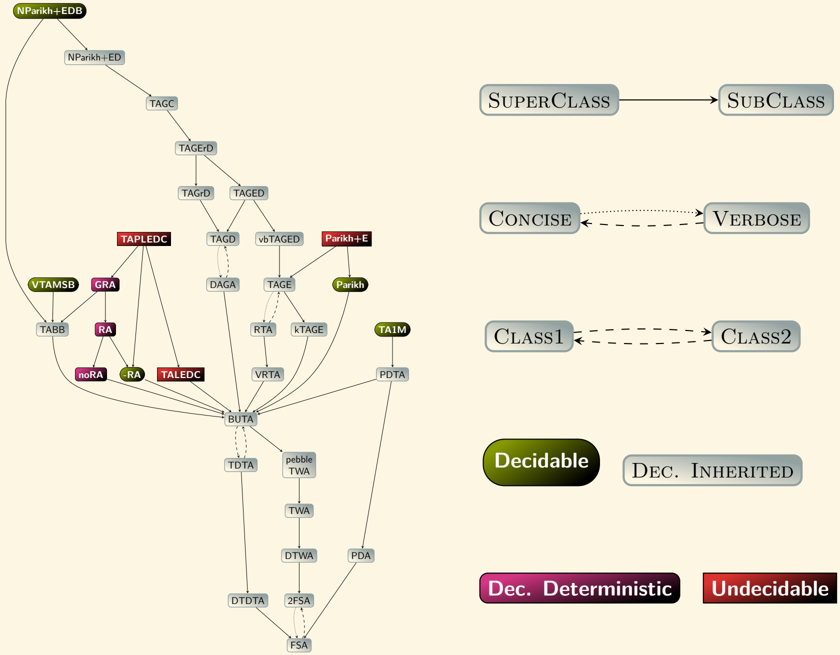
Brief survey: [Hugot, 2013]

Global constraints:

- **TAGED** [Filiot et al., 2008, Filiot et al., 2010, Filiot, 2008].
- **RTA** [Jacquemard et al., 2009, Jacquemard et al., 2011, Vacher, 2010]: rigid states $q \approxeq q$.
- **Parick+E**, **NParick+EDB** [Barguñó et al., 2010]:

$$\sum_{q \in Q} a_q |q| \geqslant b \quad \text{or} \quad \sum_{q \in Q} a_q \|q\| \geqslant b \quad \text{with } a_q, b \in \mathbb{Z}, \forall q \in Q \,.$$

- **DAGA** [Charatonik, 1999]: equivalent to TAGD.
- **TA1M** [Comon and Cortier, 2005, Comon et al., 2008b]: one tree memory generalises pushdown automata.


NParikh+EDB
NParikh+ED
TAGC
TAGErD
TAGrD
TAGED
TAPLEDC
TAGD
vbTAGED
Parikh+E
VTAMSB
GRA
DAGA
TAGE
Parikh
TABB
RA
RTA
kTAGE
TA1M
noRA
-RA
TALEDC
VRTA
PDTA
BUTA
TDTA
pebble TWA
TWA
DTWA
PDA
DTDTA
2FSA
FSA
SuperClass
SubClass
Concise
Verbose
Class1
Class2
Decidable
Dec. Inherited
Dec. Deterministic
Undecidable

References I

Barguñó, L., Creus, C., Godoy, G., Jacquemard, F., and Vacher, C. (2010).
The emptiness problem for tree automata with global constraints.
In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS'10)*, pages 263–272, Edinburgh, Scotland, UK. IEEE Computer Society Press.

Bogaert, B. and Tison, S. (1992).
Equality and disequality constraints on direct subterms in tree automata.
In Finkel, A. and Jantzen, M., editors, *STACS*, volume 577 of *Lecture Notes in Computer Science*, pages 161–171. Springer.

References II

Caron, A.-C., Comon, H., Coquidé, J.-L., Dauchet, M., and Jacquemard, F. (1994).
Pumping, cleaning and symbolic constraints solving.
In Abiteboul, S. and Shamir, E., editors, *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 436–449. Springer.

Charatonik, W. (1999).
Automata on DAG representations of finite trees.
Tech. Report, (-).

Comon, H. and Cortier, V. (2005).
Tree automata with one memory set constraints and cryptographic protocols.
Theoretical Computer Science, 331(1):143–214.

References III

Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., and Tommasi, M. (2008a).
Tree Automata Techniques and Applications.
release November 18, 2008.

Comon, H., Jacquemard, F., and Perrin, N. (2008b).
Visibly tree automata with memory and constraints.
Logical Methods in Computer Science, 4(2).

Dauchet, M., Caron, A.-C., and Coquidé, J.-L. (1995).
Automata for reduction properties solving.
J. Symb. Comput., 20(2):215–233.

References IV

Filiot, E. (2008).
Logics for n-ary queries in trees.
PhD thesis, Université des Sciences et Technologie de Lille - Lille I.

Filiot, E., Talbot, J.-M., and Tison, S. (2008).
Tree automata with global constraints.
In *Developments in Language Theory*, volume 5257 of *Lecture Notes in Computer Science*, pages 314–326. Springer.

Filiot, E., Talbot, J.-M., and Tison, S. (2010).
Tree automata with global constraints.
Int. J. Found. Comput. Sci., 21(4):571–596.

References V

Héam, P., Hugot, V., and Kouchnarenko, O. (2010).
SAT Solvers for Queries over Tree Automata with Constraints.
In *Third International Conference on Software Testing, Verification, and Validation Workshops*, pages 343–348. IEEE.

Hugot, V. (2013).
Tree Automata, Approximations, and Constraints for Verification – Tree (Not Quite) Regular Model-Checking.
Ph.D. thesis (to be defended), Université de Franche-Comté.

Jacquemard, F., Klay, F., and Vacher, C. (2009).
Rigid tree automata.
In Horia Dediu, A., Mihai Ionescu, A., and Martín-Vide, C., editors, *Proceedings of the 3rd International Conference on Language and Automata Theory and Applications (LATA'09)*,

References VI

volume 5457 of *Lecture Notes in Computer Science*, pages 446–457, Tarragona, Spain. Springer.

Jacquemard, F., Klay, F., and Vacher, C. (2011).
Rigid tree automata and applications.
Inf. Comput., 209(3):486–512.

Mongy, J. (1981).
Transformation de noyaux reconnaissables d'arbres. Forêts RATEG.
Ph.d. thesis, LIF de Lille, Université des Sciences et Technologies de Lille, Lille, France.

Vacher, C. (2010).
Tree automata with global constraints for the verification of security properties.
Ph.D. thesis, ENS Cachan.