# Advanced Topics in Term Rewriting

## LVA 703610

`http://cl-informatik.uibk.ac.at/teaching/ws06/attr/`

Georg Moser

office hours: Tuesday, 16:00–18:00 (3M09)





## Tree Automata

- tree automaton is quadruple $\mathcal{A} = (\mathcal{F}, Q, Q_f, \Delta)$ with
  1. $\mathcal{F}$ signature
  2. $Q$ states
  3. $Q_f \subseteq Q$ final states
  4. $\Delta$ transition rules

$$\mathsf{f}(\alpha_1, \dots, \alpha_n) \to \beta$$
$$\alpha \to \beta \qquad \text{epsilon transition}$$

- language accepted by $\mathcal{A}$:

$$L(\mathcal{A}) = \{ t \in \mathcal{T}(\mathcal{F}) \mid \exists\, \alpha \in Q_f : t \xrightarrow[\Delta]{*} \alpha \}$$

- $L \subseteq \mathcal{T}(\mathcal{F})$ is regular if $L = L(\mathcal{A})$ for some tree automaton $\mathcal{A}$





tree automaton $\mathcal{A}$

- signature: $\mathsf{a}$ (constant) $\mathsf{f}$ $\mathsf{g}$ (unary)
- states: $\alpha$ $\underline{\beta}$ final
- transition rules:

$$\mathsf{a} \to \alpha \qquad \mathsf{g}(\alpha) \to \alpha$$
$$\mathsf{f}(\alpha) \to \beta \qquad \beta \to \alpha$$

$$\mathsf{f}(\mathsf{f}(\mathsf{a})) \in L(\mathcal{A}): \quad \mathsf{f}(\mathsf{f}(\mathsf{a})) \to \mathsf{f}(\mathsf{f}(\alpha)) \to \mathsf{f}(\beta) \to \mathsf{f}(\alpha) \to \beta$$

$$\mathsf{g}(\mathsf{a}) \notin L(\mathcal{A}): \quad \mathsf{g}(\mathsf{a}) \to \mathsf{g}(\alpha) \to \alpha$$

$$L(\mathcal{A}) = \{ t \mid \mathsf{root}(t) = \mathsf{f} \}$$





## Examples of regular tree languages

- set of all ground terms
- set of well-typed terms with respect to order-sorted signature

sorts: $\mathtt{nat} < \mathtt{int}$ $\quad \mathtt{bool}$

signature:

$$\begin{array}{ll} 0 : \mathtt{nat} & + : \mathtt{nat} \times \mathtt{nat} \to \mathtt{nat} \\ \mathsf{s} : \mathtt{nat} \to \mathtt{nat} & + : \mathtt{int} \times \mathtt{int} \to \mathtt{int} \\ \mathsf{s} : \mathtt{int} \to \mathtt{int} & \leqslant : \mathtt{int} \times \mathtt{int} \to \mathtt{bool} \\ \mathsf{p} : \mathtt{int} \to \mathtt{int} & \end{array}$$

tree automaton:

$$\begin{array}{ll} 0 \to \mathtt{nat} & +(\mathtt{nat}, \mathtt{nat}) \to \mathtt{nat} \\ \mathsf{s}(\mathtt{nat}) \to \mathtt{nat} & +(\mathtt{int}, \mathtt{int}) \to \mathtt{int} \\ \mathsf{s}(\mathtt{int}) \to \mathtt{int} & \leqslant(\mathtt{int}, \mathtt{int}) \to \mathtt{bool} \\ \mathsf{p}(\mathtt{int}) \to \mathtt{int} & \mathtt{nat} \to \mathtt{int} \end{array}$$

# Deterministic Tree Automata

**Theorem**

every regular language is accepted by deterministic completely defined tree automaton $\mathcal{A} = (\mathcal{F}, Q, Q_f, \Delta)$

- ➡ no epsilon transitions
- ➡ no different transition rules with same left-hand sides
- ➡ $\forall f \in \mathcal{F} \ \ \forall \alpha_1, \dots, \alpha_n \in Q \ \ \exists f(\alpha_1, \dots, \alpha_n) \to \beta \in \Delta$





**Proof by Example**

$$\mathcal{A}_1 \quad \begin{array}{rcl} \mathsf{a} & \to & \alpha \\ \mathsf{f}(\alpha) & \to & \beta \\ \mathsf{g}(\alpha) & \to & \alpha \\ \beta & \to & \alpha \end{array} \qquad\qquad \begin{array}{rcl} \mathsf{a} & \to & \alpha \\ \mathsf{f}(\alpha) & \to & \beta \\ \mathsf{g}(\alpha) & \to & \alpha \\ \mathsf{f}(\alpha) & \to & \alpha \end{array} \quad \mathcal{A}_2$$

① remove epsilon transitions

$$\begin{array}{rcl} \mathsf{a} & \to & \{\alpha\} \\ \mathsf{f}(\varnothing) & \to & \varnothing \\ \mathsf{f}(\{\alpha\}) & \to & \{\alpha, \beta\} \\ \mathsf{f}(\{\beta\}) & \to & \varnothing \\ \mathsf{f}(\{\alpha, \beta\}) & \to & \{\alpha, \beta\} \end{array} \qquad\qquad \begin{array}{rcl} \\ \mathsf{g}(\varnothing) & \to & \varnothing \\ \mathsf{g}(\{\alpha\}) & \to & \{\alpha\} \\ \mathsf{g}(\{\beta\}) & \to & \varnothing \\ \mathsf{g}(\{\alpha, \beta\}) & \to & \{\alpha\} \end{array} \quad \mathcal{A}_3$$

② subset construction

③ remove inaccessible states (optional)





# Closure Properties

**Theorem**

regular languages are effectively closed under union, intersection and difference

**Proof** union

tree automaton $\mathcal{A}_1 = (\mathcal{F}, Q_1, Q_f^1, \Delta_1)$

tree automaton $\mathcal{A}_2 = (\mathcal{F}, Q_2, Q_f^2, \Delta_2)$

wlog $Q_1 \cap Q_2 = \varnothing$

$L(\mathcal{A}_1) \cup L(\mathcal{A}_2) = L(\mathcal{A})$ for tree automaton $\mathcal{A} = (\mathcal{F}, Q, Q_f, \Delta)$ with

- ➡ $Q = Q_1 \cup Q_2$
- ➡ $Q_f = Q_f^1 \cup Q_f^2$
- ➡ $\Delta = \Delta_1 \cup \Delta_2$

□





**Proof** intersection

tree automaton $\mathcal{A}_1 = (\mathcal{F}, Q_1, Q_f^1, \Delta_1)$
tree automaton $\mathcal{A}_2 = (\mathcal{F}, Q_2, Q_f^2, \Delta_2)$

$L(\mathcal{A}_1) \cap L(\mathcal{A}_2) = L(\mathcal{A})$ for tree automaton $\mathcal{A} = (\mathcal{F}, Q, Q_f, \Delta)$ with

- ➡ $Q = Q_1 \times Q_2$
- ➡ $Q_f = Q_f^1 \times Q_f^2$
- ➡ $\Delta$:

$$\begin{array}{rll} \mathsf{f}([\alpha_1, \beta_1], \dots, [\alpha_n, \beta_n]) \to [\alpha, \beta] & \forall\, \mathsf{f}(\alpha_1, \dots, \alpha_n) \to \alpha \in \Delta_1 \\ & \forall\, \mathsf{f}(\beta_1, \dots, \beta_n) \to \beta \in \Delta_2 \\ [\alpha, \beta] \to [\alpha', \beta] & \forall\, \alpha \to \alpha' \in \Delta_1 \\ [\alpha, \beta] \to [\alpha, \beta'] & \forall\, \beta \to \beta' \in \Delta_2 \end{array}$$

□

**Proof** difference

tree automaton $\mathcal{A}_1 = (\mathcal{F}, Q_1, Q_f^1, \Delta_1)$
tree automaton $\mathcal{A}_2 = (\mathcal{F}, Q_2, Q_f^2, \Delta_2)$

deterministic completely defined

$L(\mathcal{A}_1) \setminus L(\mathcal{A}_2) = L(\mathcal{A})$ for tree automaton $\mathcal{A} = (\mathcal{F}, Q, Q_f, \Delta)$ with

- $Q = Q_1 \times Q_2$
- $Q_f = Q_f^1 \times (Q_2 \setminus Q_f^2)$
- $\Delta$:

$$f([\alpha_1, \beta_1], \ldots, [\alpha_n, \beta_n]) \to [\alpha, \beta] \quad \forall\, f(\alpha_1, \ldots, \alpha_n) \to \alpha \in \Delta_1, \ \forall\, f(\beta_1, \ldots, \beta_n) \to \beta \in \Delta_2$$

$$[\alpha, \beta] \to [\alpha', \beta] \quad \forall\, \alpha \to \alpha' \in \Delta_1$$

□





# Decidability

**Theorem**

- membership — instance: tree automaton $\mathcal{A}$ and term $t$; question: $t \in L(\mathcal{A})$?
- emptiness — instance: tree automaton $\mathcal{A}$; question: $L(\mathcal{A}) = \varnothing$?
- finiteness — instance: tree automaton $\mathcal{A}$; question: $L(\mathcal{A})$ is finite?
- inclusion — instance: tree automata $\mathcal{A}_1$ and $\mathcal{A}_2$; question: $L(\mathcal{A}_1) \subseteq L(\mathcal{A}_2)$?

are decidable problems