# Rational Tree Expressions and Compositions of Tree Series Transformations

Andreas Maletti

July 5, 2006

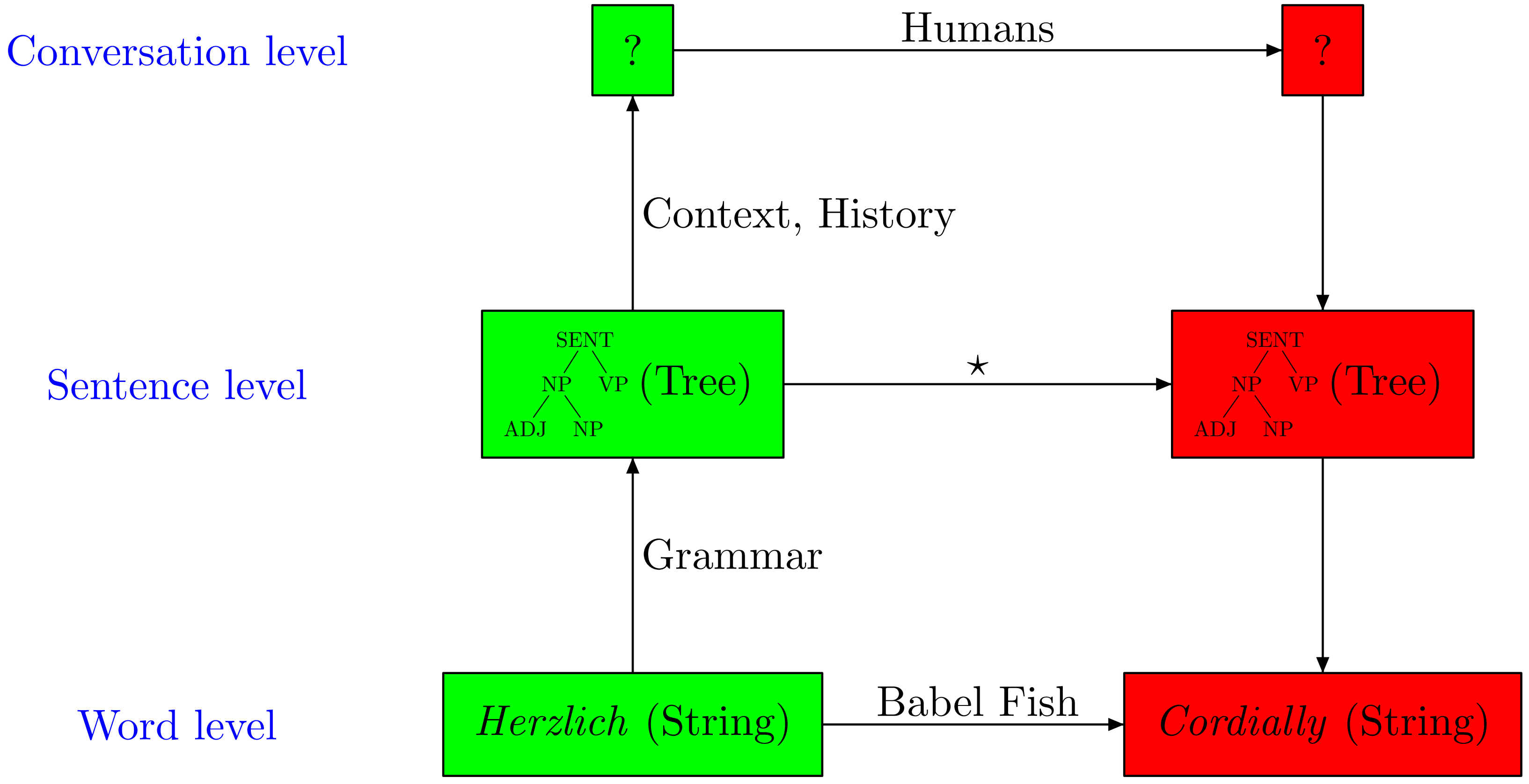
1. Motivation (from my point of view)
2. Tree Series and Weighted Tree Automata
3. Rational Tree Expressions
4. Tree Series Substitution and Tree Series Transducers
5. Compositions of Tree Series Transformations

# Motivation

Babel Fish Translation

| German | English |
| --- | --- |
| Ich möchte mich vorab bei den Organisatoren für die vortrefflich geleistete Arbeit bedanken. | I would like to thank you first the supervisors for the splendid carried out work. |
| Ich möchte mich vorab bei den Organisatoren, *die diese Veranstaltung erst ermöglicht haben*, für die vortrefflich geleistete Arbeit bedanken. | I would like me first the supervisors, *who made this meeting possible only*, for whom splendid carried out work thank you. |

# Motivation


Conversation level
Sentence level
Word level
?
Humans
?
Context, History
SENT
NP
VP
(Tree)
ADJ
NP
⋆
SENT
NP
VP
(Tree)
ADJ
NP
Grammar
*Herzlich* (String)
Babel Fish
*Cordially* (String)

# Motivation

- Automatic translation is widely used (even Microsoft uses it to translate English documentation into German)

  Beheben:
  Das globale Hooks muss es in mehreren Prozessen hinzufügen, die eine gültige, konsistente Funktion erfordern, um eine gültige, konsistente Funktion aufzurufen. Da diese Funktionszeiger sich Proxy befinden, die an den Flug erstellt werden, hat verwalteter Code kein Konzept eines konsistenten Werts für Funktionszeiger.

  Fix:
  The global hooks must add it in several processes, which demand a valid, consistent function to call a valid, consistent function. Because these function pointers are located proxy, which are created at the flight, managed code has no concept of a consistent value for function pointers.

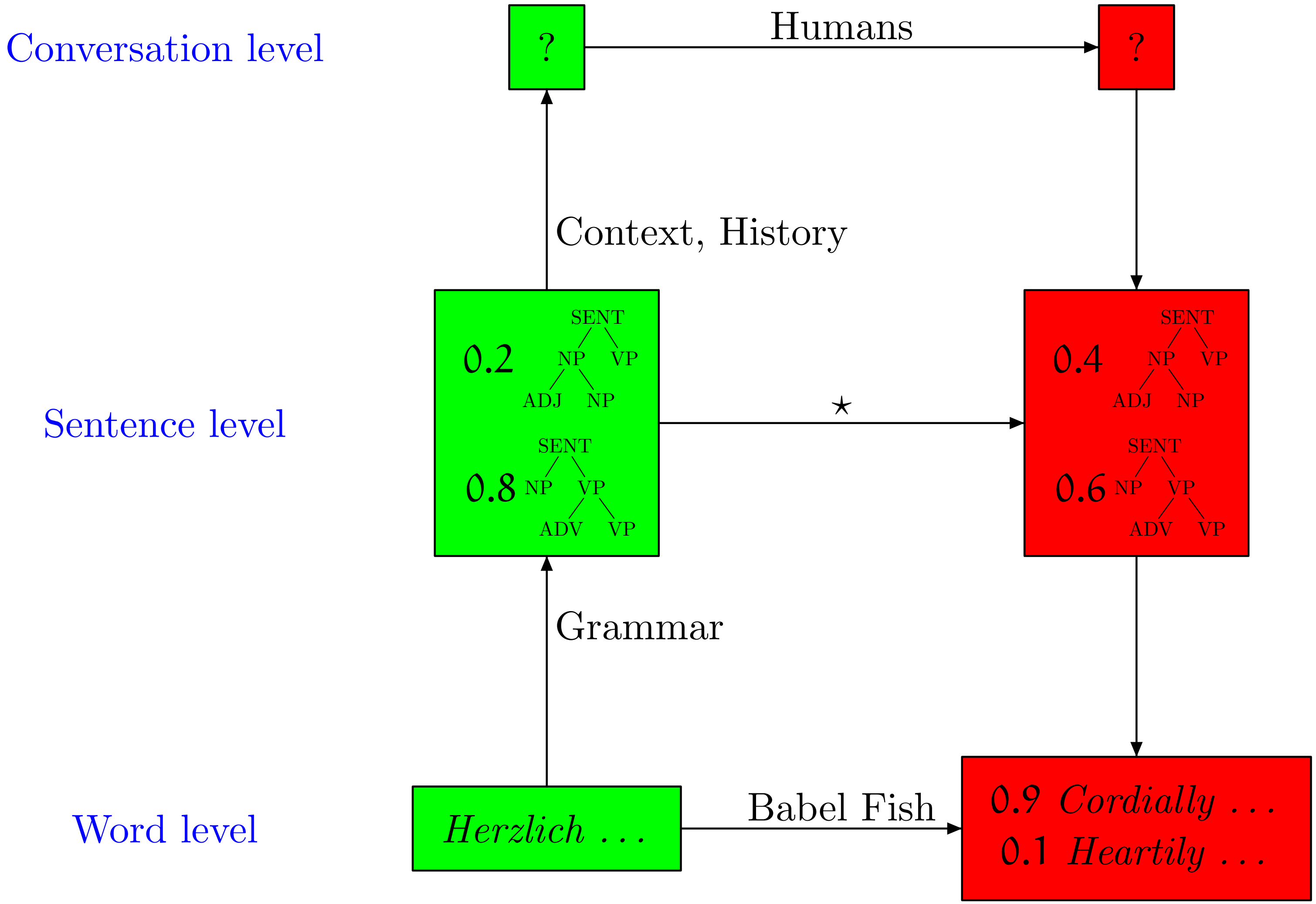
# Motivation

- Dictionaries are very powerful word-to-word translators; leave few words untranslated
- Outcome is nevertheless usually unhappy and ungrammatical
- Post-processing necessary

*Major problem:* Ambiguity of natural language

*Common approach:*
- "Soft output" (results equipped with a probability)
- Human choses the correct translation among the more likely ones

# Motivation



Conversation level
?
Humans
?
Context, History
Sentence level
0.2
SENT
NP VP
ADJ NP
0.8
SENT
NP VP
ADV VP
⋆
0.4
SENT
NP VP
ADJ NP
0.6
SENT
NP VP
ADV VP
Grammar
Word level
Herzlich . . .
Babel Fish
0.9 Cordially . . .
0.1 Heartily . . .

# Motivation

Tree series transducers are a straightforward generalization of

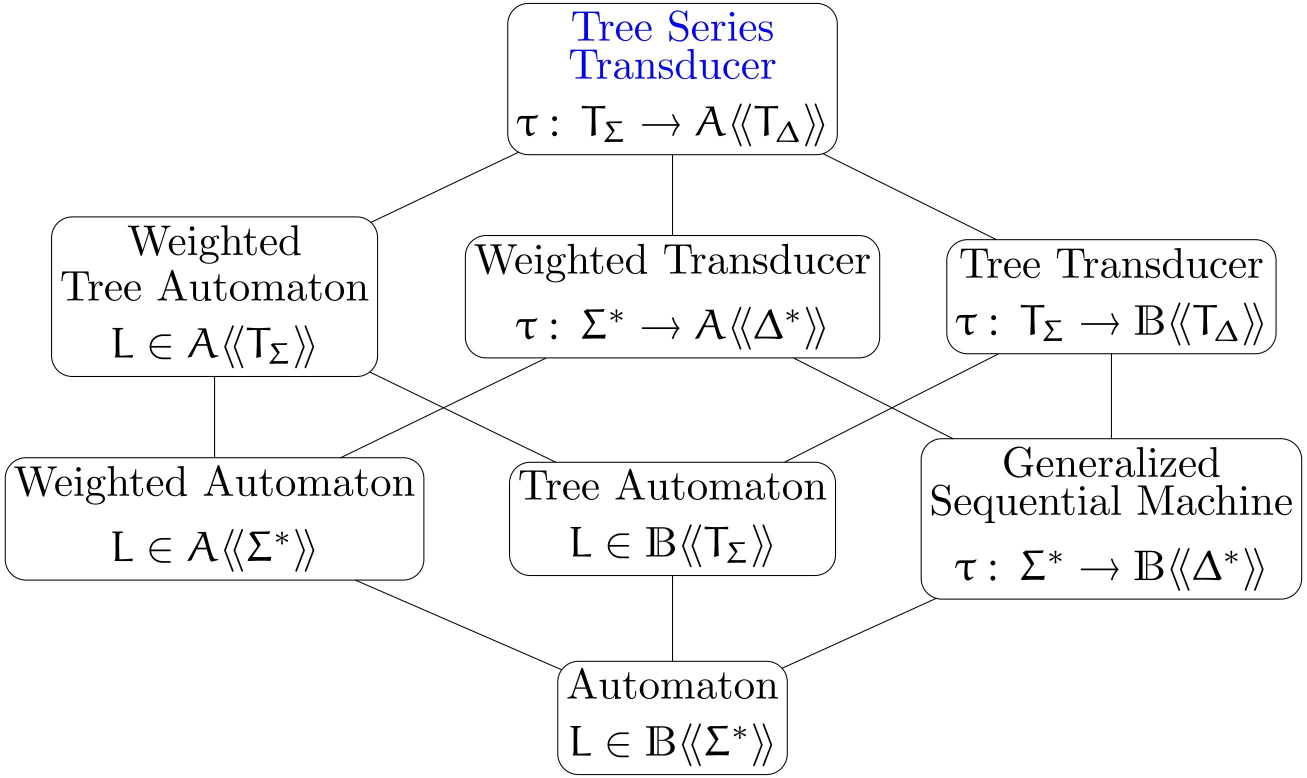
(i) tree transducers, which are applied in

- syntax-directed semantics,
- functional programming, and
- XML querying,

(ii) weighted automata, which are applied in

- (tree) pattern matching,
- image compression and speech-to-text processing.

# Generalization Hierarchy

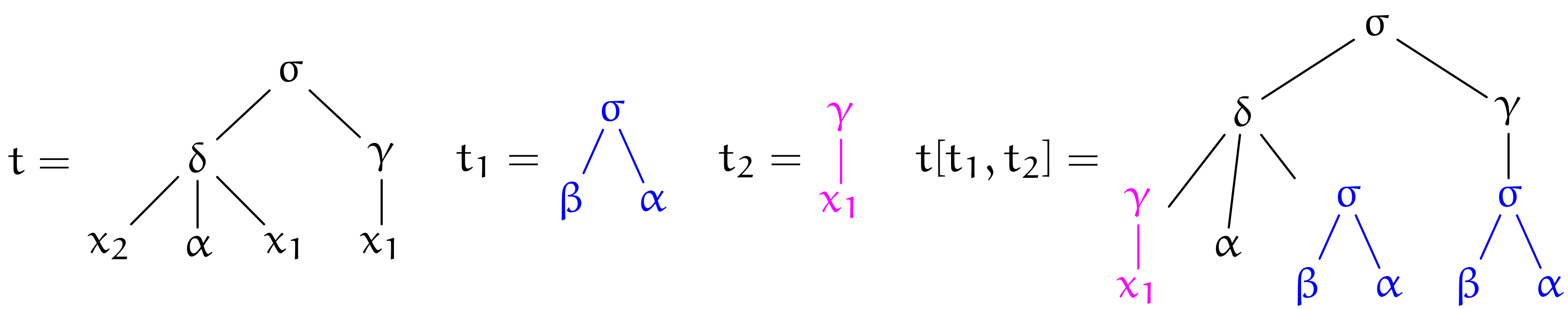


# Trees

$\Sigma$ ranked alphabet, $\Sigma_k \subseteq \Sigma$ symbols of rank $k$, $X = \{ x_i \mid i \in \mathbb{N}_+ \}$

- $T_\Sigma(X)$ set of $\Sigma$-trees indexed by $X$,
- $T_\Sigma = T_\Sigma(\emptyset)$,
- $t \in T_\Sigma(X)$ is *linear* (resp., *nondeleting*) in $Y \subseteq X$, if every $y \in Y$ occurs at most (resp., at least) once in $t$,
- $t[t_1, \ldots, t_k]$ denotes the tree substitution of $t_i$ for $x_i$ in $t$

Examples: $\Sigma = \{\sigma^{(2)}, \gamma^{(1)}, \alpha^{(0)}, \beta^{(0)}\}$ and $Y = \{x_1, x_2\}$



# Tree Series

$\Sigma$ ranked alphabet

Mappings $\varphi : T_\Sigma(X) \to \mathbb{R}$ are also called *tree series*

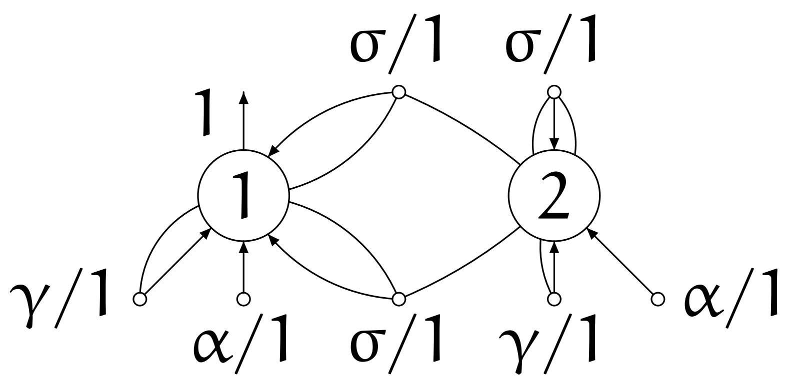
- the set of all tree series is $\mathbb{R}\langle\langle T_\Sigma(X)\rangle\rangle$,
- the *coefficient* of $t \in T_\Sigma(X)$ in $\varphi$, i.e., $\varphi(t)$, is denoted by $(\varphi, t)$,
- the *sum* is defined pointwise $(\varphi_1 + \varphi_2, t) = (\varphi_1, t) + (\varphi_2, t)$,
- the *support* of $\varphi$ is $\mathrm{supp}(\varphi) = \{t \in T_\Sigma(X) \mid (\varphi, t) \neq 0\}$,
- $\varphi$ is *linear* (resp., *nondeleting* in $Y \subseteq X$), if $\mathrm{supp}(\varphi)$ is a set of trees, which are linear (resp., nondeleting in Y),
- the series $\varphi$ with $\mathrm{supp}(\varphi) = \emptyset$ is denoted by $\widetilde{0}$.

Example: $\varphi = 1\,\alpha + 1\,\beta + 3\,\sigma(\alpha, \alpha) + \ldots + 3\,\sigma(\beta, \beta) + 5\,\sigma(\alpha, \sigma(\alpha, \alpha)) + \ldots$

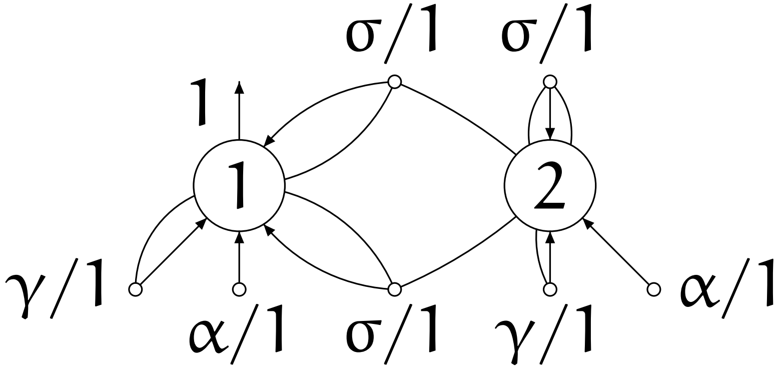
# Weighted Tree Automata

Weighted Tree Automaton $M = (Q, \Sigma, F, \mu)$

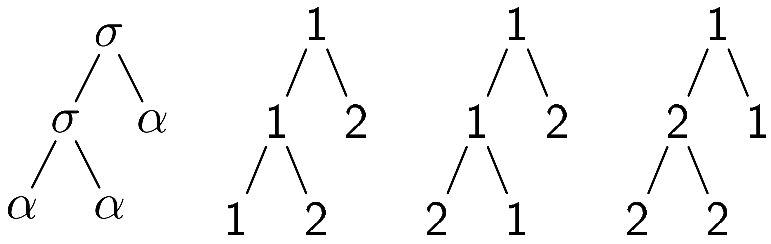
- $Q$ finite set
- $\Sigma$ input ranked alphabet
- $F : Q \longrightarrow \mathbb{R}$ final distribution
- $\mu_k : \Sigma_k \longrightarrow \mathbb{R}^{Q \times Q^k}$ tree representation



# Run Semantics of WTA




Input tree and some runs on it:

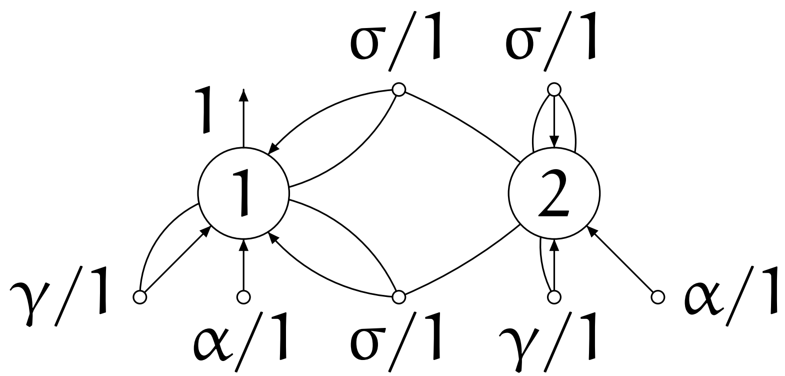


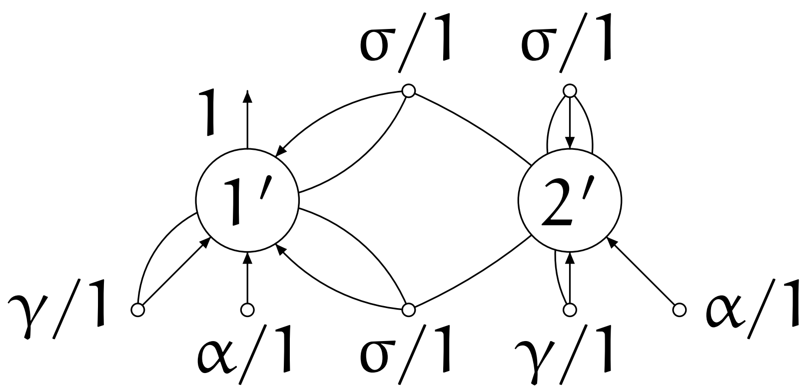

all runs have weight 1.

$$(\|M\|, \sigma(\sigma(\alpha, \alpha), \alpha)) = 3$$

# Rational Operations—Sum

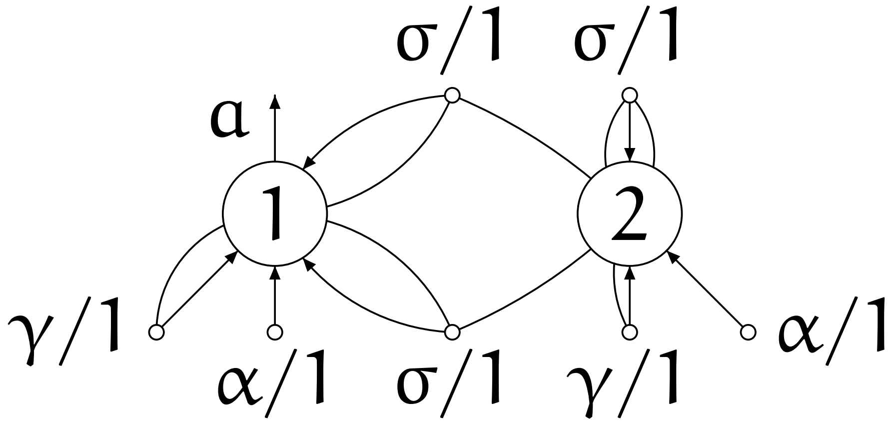
$$(\|\eta_1 + \eta_2\|, t) = (\|\eta_1\|, t) + (\|\eta_2\|, t)$$



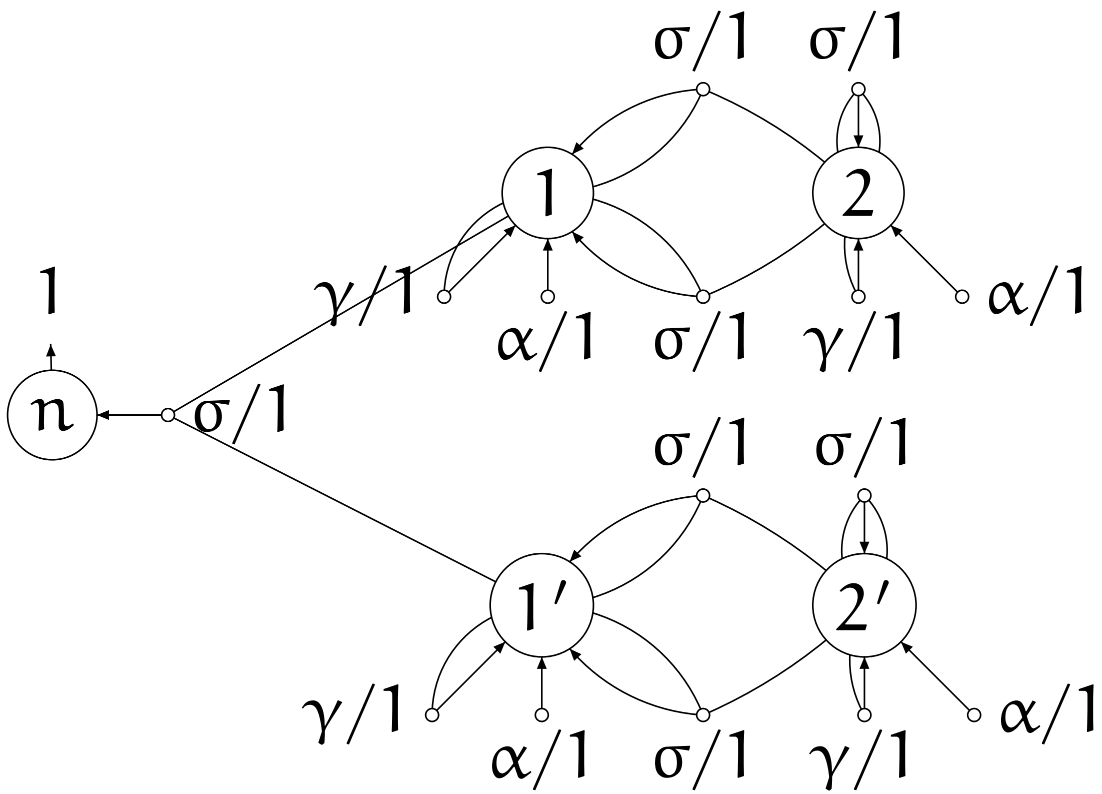
# Rational Operations—Scalar Product

$$(\|a \cdot \eta\|, t) = a \cdot (\|\eta\|, t)$$



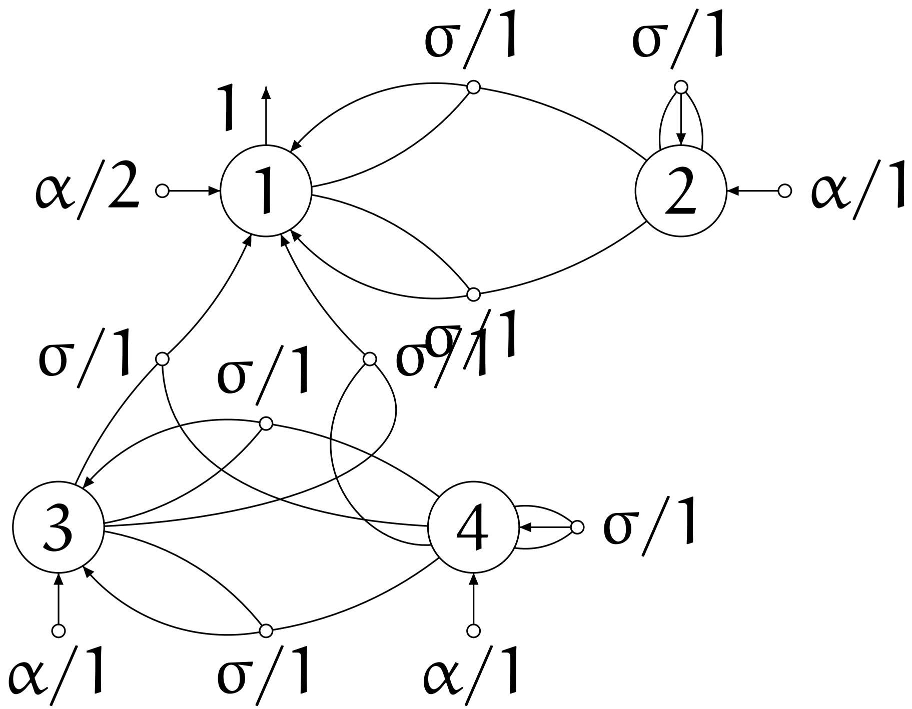
# Rational Operations—Top Concatenation

$$(\|\sigma(\eta_1,\ldots,\eta_k)\|,\sigma(t_1,\ldots,t_k)) = (\|\eta_1\|,t_1)\cdot\ldots\cdot(\|\eta_k\|,t_k)$$



# Rational Operations—Concatenation

$$(\|\eta_1 \cdot_z \eta_2\|, t) = (\|\eta_1\| \overset{\mathrm{OI}}{\leftarrow} \|\eta_2\|, t) = \sum_{\substack{t' \in T_\Sigma(z), l = |t'|_z \\ t_1, \ldots, t_l \in T_\Sigma \\ t = t'[z \leftarrow (t_1, \ldots, t_l)]}} (\|\eta_1\|, t') \cdot (\|\eta_2\|, t_1) \cdot \ldots \cdot (\|\eta_2\|, t_l)$$

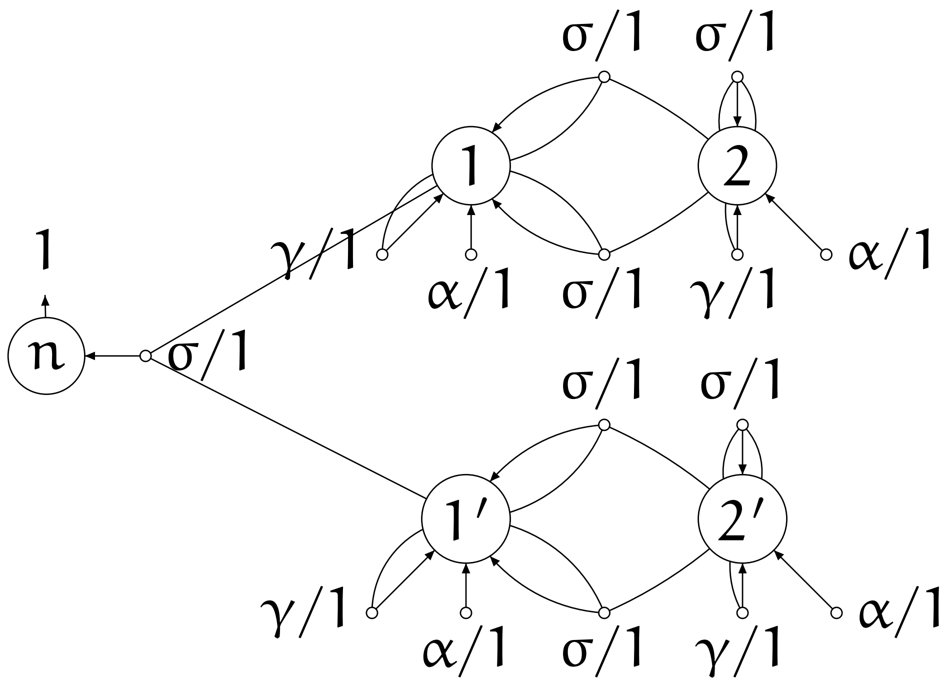


# Rational Operations—Iteration

$$(\|\eta_z^*\|, t) = (\|\eta_z^{\text{height}(t)+1}\|, t)$$

with

- $(\|\eta_z^0\|, t) = \widetilde{0}$
- $(\|\eta_z^{n+1}\|, t) = (\|\eta\| \cdot_z \|\eta_z^n\|) + 1\, z$



# Main Theorem

**Theorem.** [Droste, Pech, Vogler 05]
Rational tree expressions characterize exactly the recognizable tree series.

# Tree Series Substitution

$\varphi, \psi_1, \ldots, \psi_k \in \mathbb{R}\langle\langle T_\Sigma(X)\rangle\rangle$ with finite support

*Pure substitution* of $(\psi_1, \ldots, \psi_k)$ into $\varphi$:

$$\varphi \leftarrow (\psi_1, \ldots, \psi_k) = \sum_{\substack{t \in \mathrm{supp}(\varphi),\\ (\forall i \in [k]):\, t_i \in \mathrm{supp}(\psi_i)}} (\varphi, t) \cdot (\psi_1, t_1) \cdot \ldots \cdot (\psi_k, t_k)\; t[t_1, \ldots, t_k]$$

Example: $5\;\sigma(x_1, x_1) \leftarrow (2\;\alpha + 3\;\beta) = 10\;\sigma(\alpha, \alpha) + 15\;\sigma(\beta, \beta)$

$$5 \; \overset{\sigma}{x_1 \diagup\diagdown x_1} \leftarrow (2\;\alpha + 3\;\beta) = 10 \; \overset{\sigma}{\alpha \diagup\diagdown \alpha} + 15 \; \overset{\sigma}{\beta \diagup\diagdown \beta}$$

# Tree Series Transducers

Definition: A *(bottom-up) tree series transducer* (tst) is a system $M = (Q, \Sigma, \Delta, F, \mu)$

- $Q$ is a non-empty set of *states*,
- $\Sigma$ and $\Delta$ are input and output ranked alphabets,
- $F \in \mathbb{R}\langle\langle T_\Delta(X_1)\rangle\rangle^Q$ is a vector of linear and nondeleting tree series, also called *final output*,
- *tree representation* $\mu = (\mu_k)_{k\in\mathbb{N}}$ with $\mu_k : \Sigma_k \rightarrow \mathbb{R}\langle\langle T_\Delta(X_k)\rangle\rangle^{Q\times Q^k}$.

If $Q$ is finite and $\mu_k(\sigma)_{q,\vec{q}}$ is polynomial, then $M$ is called *finite*.

# Semantics of Tree Series Transducers

Mapping $r : \mathrm{pos}(t) \to Q$ is a *run* of $M$ on the input tree $t \in T_\Sigma$

$\mathrm{Run}(t)$ set of all runs on $t$

Evaluation mapping: $\mathrm{eval}_r : \mathrm{pos}(t) \to \mathbb{R}\langle\langle T_\Delta \rangle\rangle$ defined for every $k \in \mathbb{N}$, $\mathrm{lab}_t(p) \in \Sigma_k$ by

$$\mathrm{eval}_r(p) = \mu_k(\mathrm{lab}_t(p))_{r(p),r(p\cdot 1)\dots r(p\cdot k)} \leftarrow \big(\mathrm{eval}_r(p\cdot 1), \dots, \mathrm{eval}_r(p\cdot k)\big)$$

*Tree-series transformation* induced by $M$ is $\|M\| : \mathbb{R}\langle\langle T_\Sigma \rangle\rangle \to \mathbb{R}\langle\langle T_\Delta \rangle\rangle$ defined

$$\|M\|(\varphi) = \sum_{t \in T_\Sigma} \Big( \sum_{r \in \mathrm{Run}(t)} \mathrm{eval}_r(\varepsilon) \Big)$$

# Semantics — Example

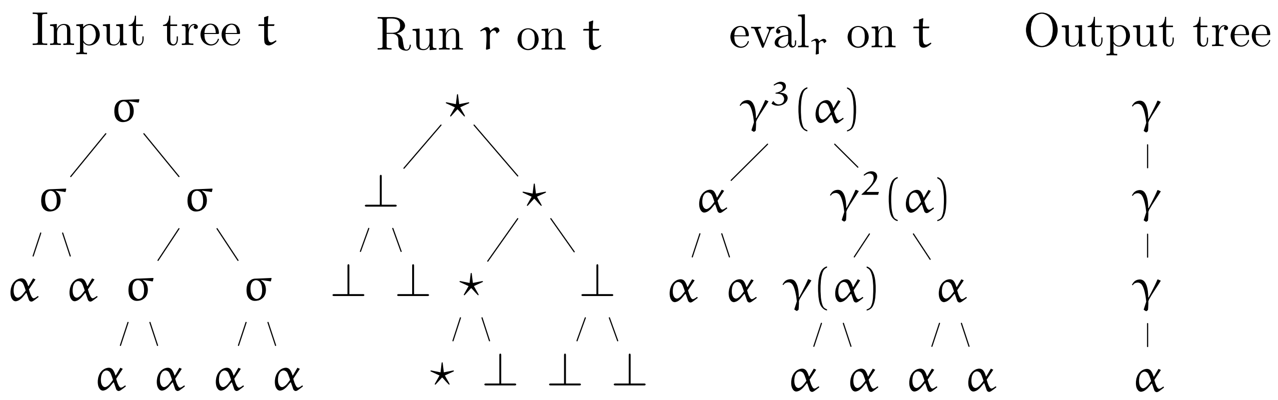
$M = (Q, \Sigma, \Delta, F, \mu)$ with

- $Q = \{\perp, \star\}$,
- $\Sigma = \{\sigma^{(2)}, \alpha^{(0)}\}$ and $\Delta = \{\gamma^{(1)}, \alpha^{(0)}\}$,
- $F_\perp = \widetilde{0}$ and $F_\star = 1\, x_1$,
- and tree representation

$$\mu_0(\alpha)_\perp = 1\,\alpha \qquad \mu_0(\alpha)_\star = 1\,\alpha$$

$$\mu_2(\sigma)_{\perp,\perp\perp} = 1\,\alpha \qquad \mu_2(\sigma)_{\star,\star\perp} = 1\,\gamma(x_1) \qquad \mu_2(\sigma)_{\star,\perp\star} = 1\,\gamma(x_2)$$

# Semantics — Example (cont.)




$\|M\|(1\ t) = 2\,\gamma^2(\alpha) + 4\,\gamma^3(\alpha)$

# Extension

$(Q, \Sigma, \Delta, F, \mu)$ tree series transducer, $\vec{q} \in Q^k$, $q \in Q$, $\varphi \in \mathbb{R}\langle\langle T_\Sigma(X_k)\rangle\rangle$ with finite support

Definition: We define $h_\mu^{\vec{q}} : T_\Sigma(X_k) \to \mathbb{R}\langle\langle T_\Delta(X_k)\rangle\rangle^Q$

$$h_\mu^{\vec{q}}(x_i)_q = \begin{cases} 1\, x_i & \text{, if } q = q_i \\ \widetilde{0} & \text{, otherwise} \end{cases}$$

$$h_\mu^{\vec{q}}(\sigma(t_1, \ldots, t_k))_q = \sum_{p_1, \ldots, p_k \in Q} \mu_k(\sigma)_{q, p_1 \ldots p_k} \leftarrow (h_\mu^{\vec{q}}(t_1)_{p_1}, \ldots, h_\mu^{\vec{q}}(t_k)_{p_k})$$

We define $h_\mu^{\vec{q}} : \mathbb{R}\langle\langle T_\Sigma(X_k)\rangle\rangle \to \mathbb{R}\langle\langle T_\Delta(X_k)\rangle\rangle^Q$ by

$$h_\mu^{\vec{q}}(\varphi)_q = \sum_{t \in T_\Sigma(X_k)} (\varphi, t) \cdot h_\mu^{\vec{q}}(t)_q$$

# Composition Construction

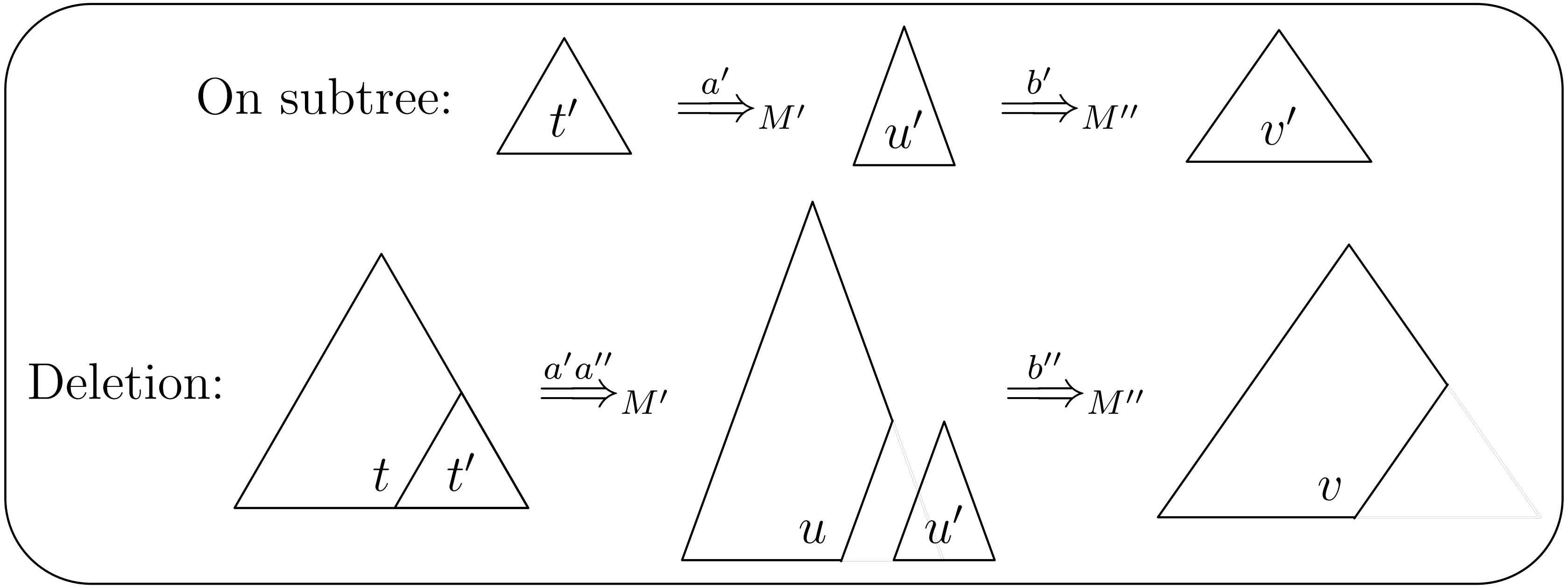
$M_1 = (Q_1, \Sigma, \Delta, F_1, \mu_1)$ and $M_2 = (Q_2, \Delta, \Gamma, F_2, \mu_2)$ tree series transducer

Definition: The *product of* $M_1$ and $M_2$, denoted by $M_1\,; M_2$, is the tree series transducer

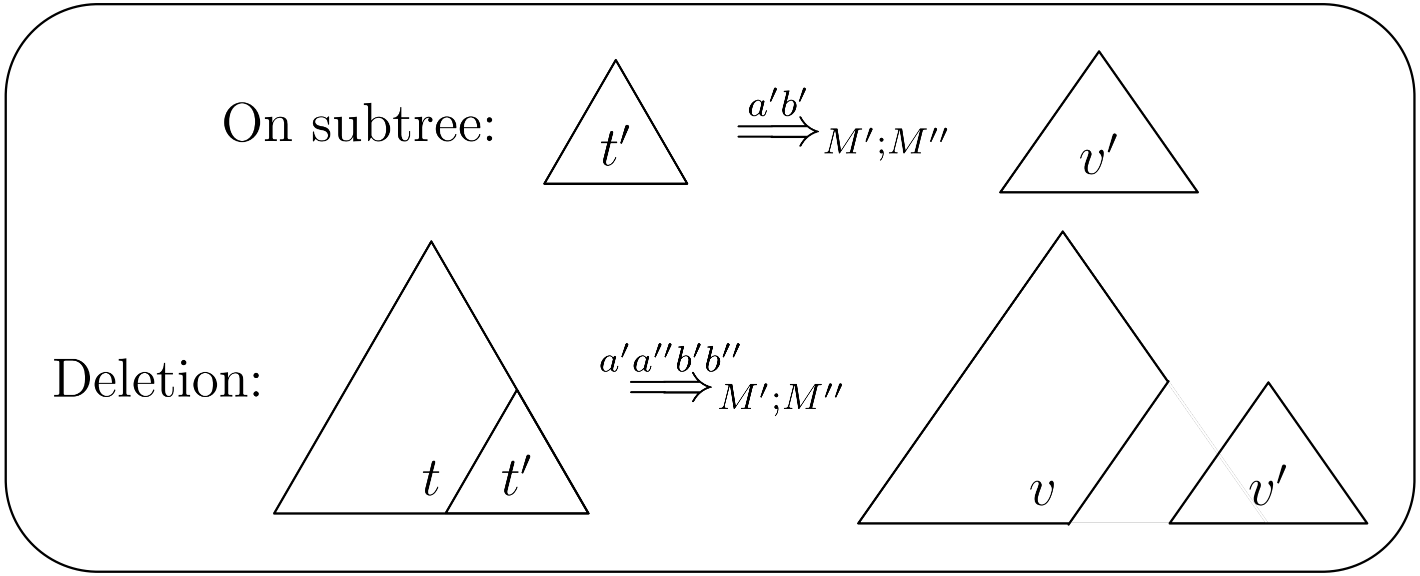
$$M = (Q_1 \times Q_2, \Sigma, \Gamma, F, \mu)$$

- $F_{pq} = \sum_{i \in Q_2} (F_2)_i \leftarrow h^q_{\mu_2}\big((F_1)_p\big)_i$
- $\mu_k(\sigma)_{pq,(p_1 q_1, \dots, p_k q_k)} = h^{q_1 \cdots q_k}_{\mu_2}\big((\mu_1)_k(\sigma)_{p, p_1 \dots p_k}\big)_q.$

# Composition



# Composition (cont.)



# Main Theorem

**Theorem.** [M. 05]

- $\text{l-BOT}_{\text{ts-ts}}(\mathbb{R}) \,;\, \text{BOT}_{\text{ts-ts}}(\mathbb{R}) = \text{BOT}_{\text{ts-ts}}(\mathbb{R})$.
- $\text{BOT}_{\text{ts-ts}}(\mathbb{R}) \,;\, \text{db-BOT}_{\text{ts-ts}}(\mathbb{R}) = \text{BOT}_{\text{ts-ts}}(\mathbb{R})$,

# Top-down Tree Series Transducers?

Why not top-down tree series transducers?

Few known results are proved for special cases where bottom-up device can simulate top-down device!

# References


[Borchardt 04] B. Borchardt: *Code Selection by Tree Series Transducers*. CIAA'04, Kingston, Canada, 2004.

[Engelfriet et al 02] J. Engelfriet, Z. Fülöp, and H. Vogler: *Bottom-up and Top-down Tree Series Transformations*. Journal of Automata, Languages, and Combinatorics 7:11–70, 2002

[Fülöp et al 03] Z. Fülöp and H. Vogler: *Tree Series Transformations that Respect Copying*. Theory of Computing Systems 36:247–293, 2003

[Kuich 99] W. Kuich: *Tree Transducers and Formal Tree Series*. Acta Cybernetica 14:135–149, 1999