Decomposition of Weighted Multioperator Tree Automata

Torsten Stüber

Department of Computer Science
Dresden University of Technology

Szeged, May 30th, 2006

Outline

Outline

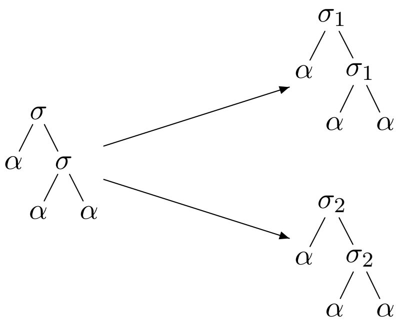
1. Motivation
2. Weighted Multioperator Tree Automata
3. Decomposition

Introductory Example

- Ranked alphabet $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$
- Tree transformation:
  - replace every σ by σ_1 or
  - replace every σ by σ_2

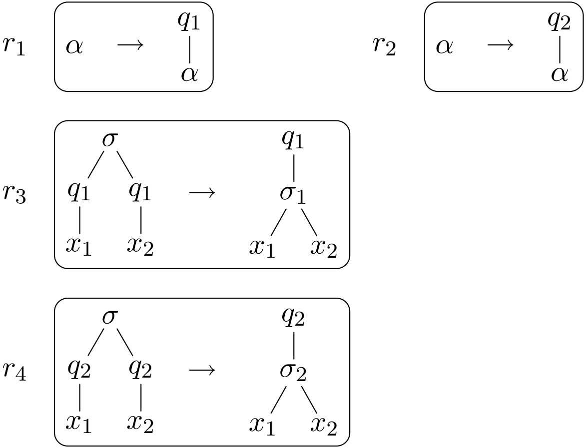
Introductory Example

- Ranked alphabet $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$
- Tree transformation:
  - replace every σ by σ_1 or
  - replace every σ by σ_2



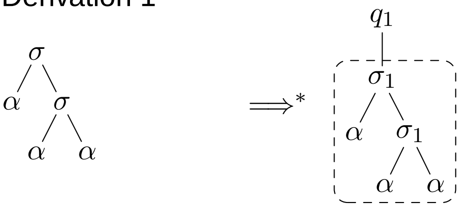
Use a Bottom-Up Tree Transducer

- State set: $\{q_1, q_2\}$, both accepting
- Rule set:

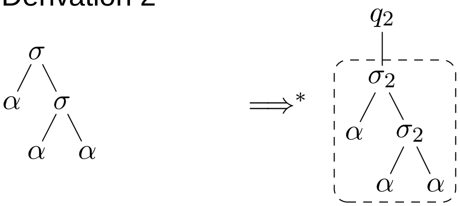


Example Derivation

- Derivation 1

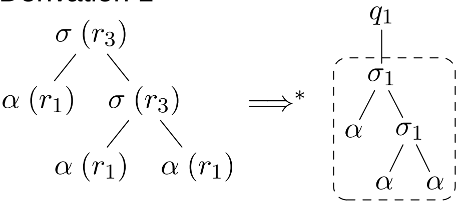


- Derivation 2



Example Derivation

- Derivation 1



- Derivation 2

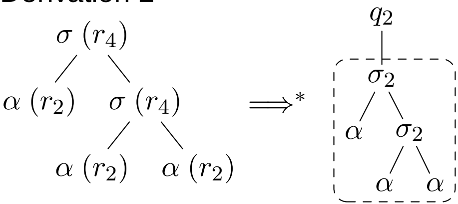


Illustration of the Decomposition

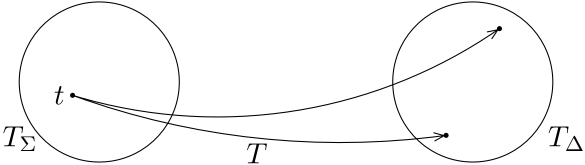


Illustration of the Decomposition

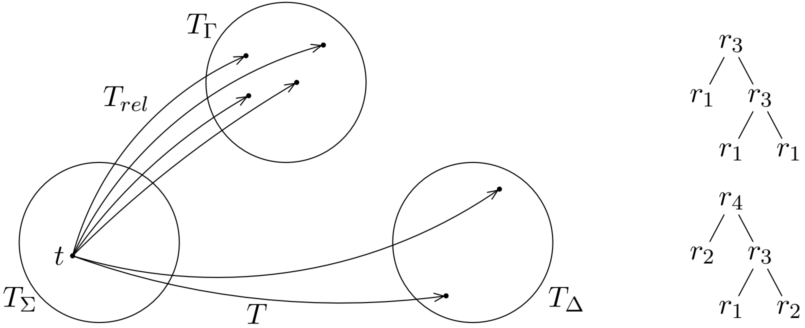


Illustration of the Decomposition

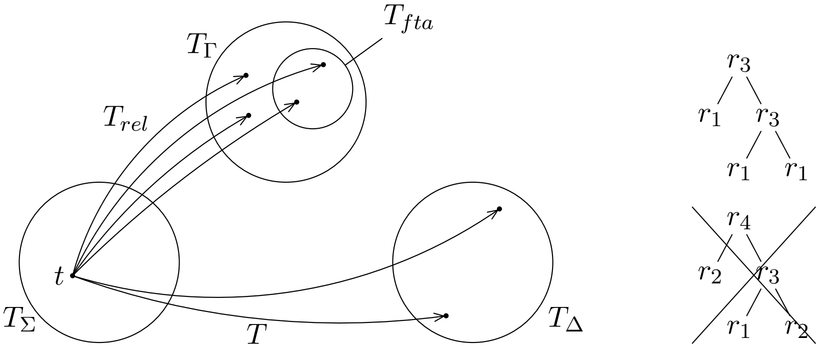
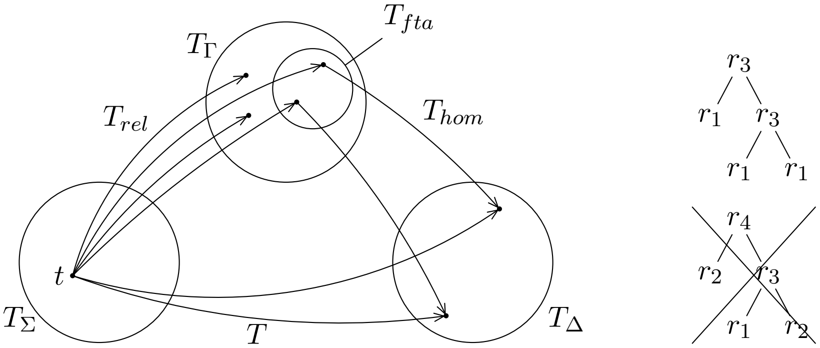


Illustration of the Decomposition



Decomposed Tree Transducer

- Tree Transducer T_{rel}
  - state set: $\{*\}$
  - new output alpabet: $\Gamma = \{r_1, r_2, r_3, r_4\}$
  - $\alpha \to *(r_1)$ $\quad\sigma(*(x_1), *(x_2)) \to *(r_3(x_1, x_2))$
    $\alpha \to *(r_2)$ $\quad\sigma(*(x_1), *(x_2)) \to *(r_4(x_1, x_2))$

- Tree Transducer T_{fta}
  - state set: $\{q_1, q_2\}$
  - $r_1 \to q_1(r_1)$ $\quad r_3(q_1(x_1), q_1(x_2)) \to q_1(r_3(x_1, x_2))$
    $r_2 \to q_2(r_2)$ $\quad r_4(q_2(x_1), q_2(x_2)) \to q_2(r_4(x_1, x_2))$

- Tree Transducer T_{hom}
  - state set: $\{*\}$
  - $r_1 \to *(\alpha)$ $\quad r_3(*(x_1), *(x_2)) \to *(\sigma_1(x_1, x_2))$
    $r_2 \to *(\alpha)$ $\quad r_4(*(x_1), *(x_2)) \to *(\sigma_2(x_1, x_2))$

Decomposed Tree Transducer

- Tree Transducer T_{rel}
  - state set: $\{*\}$
  - new output alpabet: $\Gamma = \{r_1, r_2, r_3, r_4\}$
  - $\alpha \to *(r_1)$ $\quad$ $\sigma(*(x_1), *(x_2)) \to *(r_3(x_1, x_2))$
    $\alpha \to *(r_2)$ $\quad$ $\sigma(*(x_1), *(x_2)) \to *(r_4(x_1, x_2))$

- Tree Transducer T_{fta}
  - state set: $\{q_1, q_2\}$
  - $r_1 \to q_1(r_1)$ $\quad$ $r_3(q_1(x_1), q_1(x_2)) \to q_1(r_3(x_1, x_2))$
    $r_2 \to q_2(r_2)$ $\quad$ $r_4(q_2(x_1), q_2(x_2)) \to q_2(r_4(x_1, x_2))$

- Tree Transducer T_{hom}
  - state set: $\{*\}$
  - $r_1 \to *(\alpha)$ $\quad$ $r_3(*(x_1), *(x_2)) \to *(\sigma_1(x_1, x_2))$
    $r_2 \to *(\alpha)$ $\quad$ $r_4(*(x_1), *(x_2)) \to *(\sigma_2(x_1, x_2))$

Decomposed Tree Transducer

- Tree Transducer T_{rel}
  - state set: $\{*\}$
  - new output alpabet: $\Gamma = \{r_1, r_2, r_3, r_4\}$
  - $\alpha \to *(r_1)$ $\quad$ $\sigma(*(x_1), *(x_2)) \to *(r_3(x_1, x_2))$
    $\alpha \to *(r_2)$ $\quad$ $\sigma(*(x_1), *(x_2)) \to *(r_4(x_1, x_2))$

- Tree Transducer T_{fta}
  - state set: $\{q_1, q_2\}$
  - $r_1 \to q_1(r_1)$ $\quad$ $r_3(q_1(x_1), q_1(x_2)) \to q_1(r_3(x_1, x_2))$
    $r_2 \to q_2(r_2)$ $\quad$ $r_4(q_2(x_1), q_2(x_2)) \to q_2(r_4(x_1, x_2))$

- Tree Transducer T_{hom}
  - state set: $\{*\}$
  - $r_1 \to *(\alpha)$ $\quad$ $r_3(*(x_1), *(x_2)) \to *(\sigma_1(x_1, x_2))$
    $r_2 \to *(\alpha)$ $\quad$ $r_4(*(x_1), *(x_2)) \to *(\sigma_2(x_1, x_2))$

History of Results

Theorem (Engelfriet '75)

$BOT_{tt} \subseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

$BOT_{tt} \supseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

- Nivat '68
  Decomposition of generalized sequential machines
- Engelfriet, Fülöp, Vogler '02
  Decomposition of bottom-up tree series transducers

Prove a similar result for weighted multioperator tree automata!

History of Results

Theorem (Engelfriet '75)

$BOT_{tt} \subseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$
$BOT_{tt} \supseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

- Nivat '68
  Decomposition of generalized sequential machines
- Engelfriet, Fülöp, Vogler '02
  Decomposition of bottom-up tree series transducers

Prove a similar result for weighted multioperator tree automata!

History of Results

Theorem (Engelfriet '75)

$BOT_{tt} \subseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$
$BOT_{tt} \supseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

- Nivat '68
  Decomposition of generalized sequential machines
- Engelfriet, Fülöp, Vogler '02
  Decomposition of bottom-up tree series transducers

Prove a similar result for weighted multioperator tree automata!

Outline

Basic Notions

- ranked alphabet (Σ, rk_Σ)
- let Σ ranked alphabet
  - trees over Σ: T_Σ
  - $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$
    $\rightsquigarrow T_\Sigma = \{\alpha, \sigma(\alpha, \alpha), \sigma(\alpha, \sigma(\alpha, \alpha)), \ldots\}$
- let $t \in T_\Sigma$
  - $pos(t) \subseteq \mathbb{N}^*$
  - $t = \sigma(\alpha, \sigma(\alpha, \alpha))$
    $\rightsquigarrow pos(t) = \{\varepsilon, 1, 2, 21, 22\}$
- let $p \in pos(t)$
  - label $t(p)$
  - $p = 21 \rightsquigarrow t(p) = \alpha$

Basic Notions

- ranked alphabet (Σ, rk_Σ)
- let Σ ranked alphabet
  - trees over Σ: T_Σ
  - $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$
    $\rightsquigarrow T_\Sigma = \{\alpha, \sigma(\alpha,\alpha), \sigma(\alpha,\sigma(\alpha,\alpha)), \ldots\}$
- let $t \in T_\Sigma$
  - $pos(t) \subseteq \mathbb{N}^*$
  - $t = \sigma(\alpha, \sigma(\alpha,\alpha))$
    $\rightsquigarrow pos(t) = \{\varepsilon, 1, 2, 21, 22\}$
- let $p \in pos(t)$
  - label $t(p)$
  - $p = 21 \rightsquigarrow t(p) = \alpha$

Basic Notions

- ranked alphabet (Σ, rk_Σ)
- let Σ ranked alphabet
  - trees over Σ: T_Σ
  - $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$
    $\rightsquigarrow T_\Sigma = \{\alpha, \sigma(\alpha,\alpha), \sigma(\alpha, \sigma(\alpha,\alpha)), \ldots\}$
- let $t \in T_\Sigma$
  - $pos(t) \subseteq \mathbb{N}^*$
  - $t = \sigma(\alpha, \sigma(\alpha,\alpha))$
    $\rightsquigarrow pos(t) = \{\varepsilon, 1, 2, 21, 22\}$
- let $p \in pos(t)$
  - label $t(p)$
  - $p = 21 \rightsquigarrow t(p) = \alpha$

Multioperator Monoids

Definition ($Ops(A)$)

Let A be a set, k natural number.

- $Ops^k(A)$: k-ary operations on A,
- $Ops(A)$: operations on A.

Definition (Multioperator monoid)

A multioperator monoid (M-monoid) is a tuple $\underline{A} = (A, \oplus, 0_A, \Omega)$, where

- $(A, \oplus, 0_A)$ is a commutative monoid,
- $\Omega \subseteq Ops(A)$.

Multioperator Monoids

Definition ($Ops(A)$)

Let A be a set, k natural number.

- $Ops^k(A)$: k-ary operations on A,
- $Ops(A)$: operations on A.

Definition (Multioperator monoid)

A multioperator monoid (M-monoid) is a tuple $\underline{A} = (A, \oplus, 0_A, \Omega)$, where

- $(A, \oplus, 0_A)$ is a commutative monoid,
- $\Omega \subseteq Ops(A)$.

Top Concatentation

Let Σ ranked alphabet, $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$

Definition (Top Concatenation)

Let $t_1, \ldots, t_k \in T_\Sigma$.
$\bar{\sigma}(t_1, \ldots, t_k) = \sigma(t_1, \ldots, t_k)$.

Definition (Language Top Concatenation)

Let $L_1, \ldots, L_k \subseteq T_\Sigma$.
$\bar{\sigma}^L(L_1, \ldots, L_k) = \{\sigma(t_1, \ldots, t_k) | t_i \in L_i\}$.

- Drop the L in $\bar{\sigma}^L$.
- $L_1 = \{\alpha_1, \beta_1\}, L_2 = \{\alpha_2, \beta_2\}$
  $\rightsquigarrow \bar{\sigma}(L_1, L_2) = \{\sigma(\alpha_1, \alpha_2), \sigma(\beta_1, \alpha_2), \sigma(\alpha_1, \beta_2), \sigma(\beta_1, \beta_2)\}$

Top Concatentation

Let Σ ranked alphabet, $k \in \mathbb{N}$, $\sigma \in \Sigma^{(k)}$

Definition (Top Concatenation)

Let $t_1, \ldots, t_k \in T_\Sigma$.
$\bar{\sigma}(t_1, \ldots, t_k) = \sigma(t_1, \ldots, t_k)$.

Definition (Language Top Concatenation)

Let $L_1, \ldots, L_k \subseteq T_\Sigma$.
$\bar{\sigma}^L(L_1, \ldots, L_k) = \{\sigma(t_1, \ldots, t_k) | t_i \in L_i\}$.

- Drop the L in $\bar{\sigma}^L$.
- $L_1 = \{\alpha_1, \beta_1\}, L_2 = \{\alpha_2, \beta_2\}$
  $\rightsquigarrow \bar{\sigma}(L_1, L_2) = \{\sigma(\alpha_1, \alpha_2), \sigma(\beta_1, \alpha_2), \sigma(\alpha_1, \beta_2), \sigma(\beta_1, \beta_2)\}$

Bottom-Up Weighted Multioperator Tree Automata

Definition (Bottom-Up Weighted Multioperator Tree Automaton)

$M = (Q, \Sigma, \underline{A}, \mu, F)$ bottom-up weighted multioperator tree automaton (wta) iff

- Q finite set (of *states*)
- Σ ranked alphabet
- $\underline{A} = (A, \oplus, 0_A, \Omega)$ M-monoid
- $\mu = (\mu_k)_{k \in \mathbb{N}}$ with $\mu_k : \Sigma^k \to (\Omega^{(k)})^{Q^k \times Q}$
- $F \in (\Omega^{(1)})^Q$

Tree Series

Definition (Tree Series)

Let Σ ranked alphabet, $\underline{A} = (A, \oplus, 0_A, \Omega)$ M-monoid.
A tree series over T_Σ and A is a mapping $\varphi : T_\Sigma \to A$.
Notation:

- (φ, s) denotes $\varphi(s)$,
- $A\langle\langle T_\Sigma \rangle\rangle$ set of tree series over T_Σ and A.

Semantics (1)

Let $(Q, \Sigma, \underline{A}, \mu, F)$ be wta, $t \in T_\Sigma$.

Definition (Run)

A run of M on t is a mapping $r : pos(t) \to Q$.
$R_M(t)$ set of all runs of M on t.

Definition (Weight of a run)

Let $r \in R_M(t)$. Define weight mapping $C_r : pos(t) \to A$:

$$C_r(p) = \mu_k(t(p))_{r(p1)\cdots r(pk), r(p)}(C_r(p1), \ldots, C_r(pk)),$$

where $k = rk(t(p))$.

Semantics (1)

Let $(Q, \Sigma, \underline{A}, \mu, F)$ be wta, $t \in T_\Sigma$.

Definition (Run)

A run of M on t is a mapping $r : pos(t) \to Q$.
$R_M(t)$ set of all runs of M on t.

Definition (Weight of a run)

Let $r \in R_M(t)$. Define weight mapping $C_r : pos(t) \to A$:

$$\begin{aligned} & C_r(p) \\ & \quad = \quad \mu_k(t(p))_{r(p1)\cdots r(pk), r(p)}(C_r(p1), \ldots, C_r(pk)), \end{aligned}$$

where $k = rk(t(p))$.

Semantics(2)

Let $M = (Q, \Sigma, \underline{A}, \mu, F)$ be wta

Definition (Run Semantics)

Tree series $\varphi_M \in A\langle\langle T_\Sigma \rangle\rangle$ recognized by M

$$(\varphi_M, t) = \bigoplus_{r \in R_M(t)} F_{r(\varepsilon)}(C_r(\varepsilon)) \; .$$

Example

Example

$M_{ex} = (Q, \Sigma, \underline{A}, \mu, F)$ with

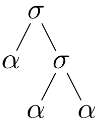
- $Q = \{q_1, q_2\}$,
- $\Sigma = \{\alpha^{(0)}, \sigma^{(2)}\}$,
- $\underline{A} = (\mathcal{P}(T_\Delta), \cup, \emptyset, \{id_{\mathcal{P}(T_\Delta)}, \widetilde{\emptyset}^{(2)}, r_1, r_2, r_3, r_4\})$,
  - $\Delta = \{\alpha^{(0)}, \sigma_1^{(2)}, \sigma_2^{(2)}\}$
  - $r_1 = r_2$ is the constant $\{\alpha\}$,
  - $r_3 : (L_1, L_2) \mapsto \overline{\sigma_1}(L_1, L_2)$,
  - $r_4 : (L_1, L_2) \mapsto \overline{\sigma_2}(L_1, L_2)$,

Example (cont'd)

- $\mu = (\mu_k)_{k \in \mathbb{N}}$,
  - $\mu_0(\alpha)_{\varepsilon, q_1} = r_1$
  - $\mu_0(\alpha)_{\varepsilon, q_2} = r_2$,
  - $\mu_2(\sigma)_{q_1 q_1, q_1} = r_3$,
  - $\mu_2(\sigma)_{q_2 q_2, q_2} = r_4$,
  - otherwise $\mu_2(\sigma)_{\_\_,\_} = \widetilde{\emptyset}^{(2)}$
- $F_{q_1} = F_{q_2} = id_{\mathcal{P}(T_\Delta)}$.

Example Computation (1)

- Given input tree:



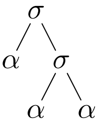
- Example run r:

- Weight mapping of r:

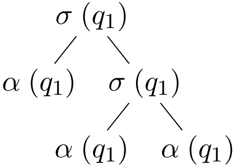
- Result:

Example Computation (1)

- Given input tree:



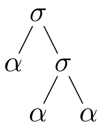
- Example run r:



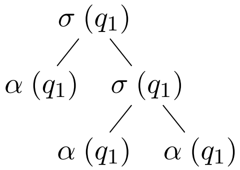
- Weight mapping of r:
- Result:

Example Computation (1)

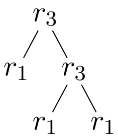
- Given input tree:



- Example run r:



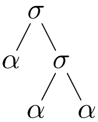
- Weight mapping of r:



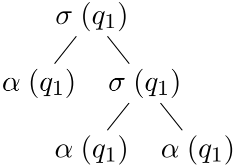
- Result:

Example Computation (1)

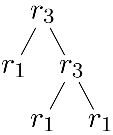
- Given input tree:



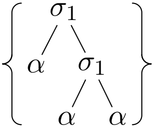
- Example run r:



- Weight mapping of r:

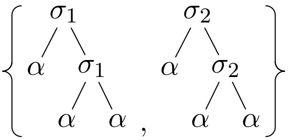


- Result:



Example Computation (2)

- Summing up over all runs:

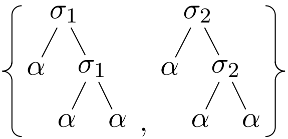


Theorem (Maletti '05)

Every bottom-up tree transducer can be simulated by a wta.

Example Computation (2)

- Summing up over all runs:

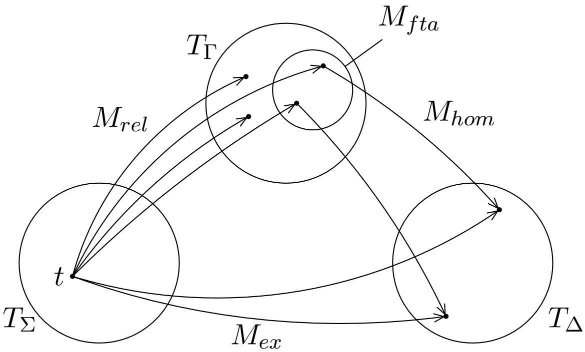


Theorem (Maletti '05)

Every bottom-up tree transducer can be simulated by a wta.

Outline

Decomposition of M_{ex}

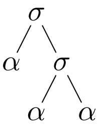


Decomposition (1)

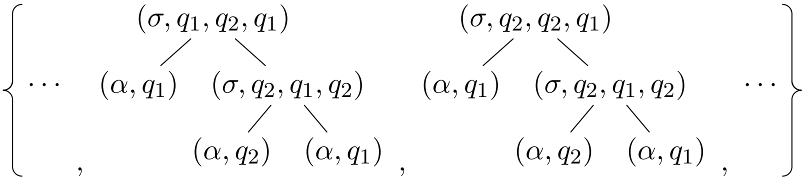
- Define ranked alphabet
  $\Gamma = \{(\alpha, q_1), (\alpha, q_2)\} \cup \{(\sigma, p_1, p_2, q) \mid p_1, p_2, q \in Q\}$.
- Define a relabeling wta

  $M_{rel} = (\{*\}, \Sigma, \underline{A}_{rel}, \mu_{rel}, F_{rel})$ where
  - $\underline{A}_{rel} = (\mathcal{P}(T_\Gamma), \cup, \emptyset, \{id_{\mathcal{P}(T_\Gamma)}, \widetilde{\alpha}, \widetilde{\sigma}\})$
    - $\widetilde{\alpha} = \{(\alpha, q_1), (\alpha, q_2)\}$,
    - $\widetilde{\sigma}(L_1, L_2) = \bigcup_{p_1, p_2, q \in Q} \overline{(\sigma, p_1, p_2, q)}(L_1, L_2)$
  - μ_{rel}:
    - $\mu_{rel}(\alpha)_{\varepsilon,*} = \widetilde{\alpha}$,
    - $\mu_{rel}(\sigma)_{**,*} = \widetilde{\sigma}$,
  - $(F_{rel})_* = id_{\mathcal{P}(T_\Gamma)}$.

Example Computation of M_{rel}



$\rightsquigarrow \quad (M_{rel})$



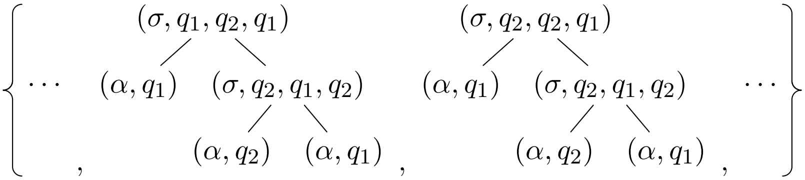
Decomposition (2)

- Define finite state tree automaton

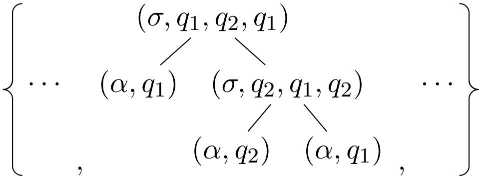
$$M_{fta} = (Q, \Gamma, \underline{A}_{fta}, \mu_{fta}, F_{fta})$$

  - $\underline{A}_{fta} = (\mathcal{P}(T_\Gamma), \cup, \emptyset, \{\widetilde{\emptyset}^{(0)}, \widetilde{\emptyset}^{(2)}, \bar{\alpha}, \bar{\sigma}\})$,
  - μ_{fta}: for every $q, q', p_1, p_1', p_2, p_2' \in Q$:
    - $\mu_{fta}((\alpha, q))_{\varepsilon, q'} = \begin{cases} \bar{\alpha}, & q = q'; \\ \widetilde{\emptyset}^{(0)}, & q \neq q' \end{cases}$
    - $\mu_{fta}((\sigma, p_1, p_2, q))_{p_1' p_2', q'} = \begin{cases} \bar{\sigma}, & q = q' \wedge p_1 = p_1' \wedge p_2 = p_2'; \\ \widetilde{\emptyset}^{(2)}, & \text{otherwise} \end{cases}$
  - $(F_{fta})_q = id_{\mathcal{P}(T_\Gamma)}$ for every $q \in Q$.

Example Computation of M_{fta}



$\rightsquigarrow\ (M_{fta})$



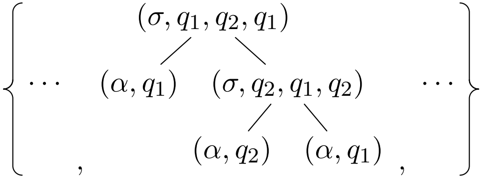
Decomposition (3)

- Define homomorphism

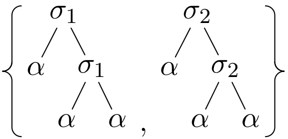
  $M_{hom} = (\{*\}, \Gamma, \underline{A}, \mu_{hom}, F_{hom})$

  - μ_{hom}: for every $q, p_1, p_2 \in Q$:
    - $\mu_{hom}((\alpha, q))_{\varepsilon,*} = \mu(\alpha)_{\varepsilon,q}$,
    - $\mu_{hom}((\sigma, p_1, p_2, q))_{*,*} = \mu(\sigma)_{p_1 p_2, q}$,
  - $(F_{hom})_* = id_A$.

Example Computation of M_{hom}



$\rightsquigarrow \quad (M_{hom})$



Definition

Let $\underline{A}$ be M-monoid.

- $BOT(\underline{A}) = \{\varphi_M \mid M$ is a wta over $\underline{A}\}$
- $HOM(\underline{A}) = \{\varphi_M \mid M$ is a homomorphism wta over $\underline{A}\}$
- $REL = \{\varphi_M \mid M$ is a relabeling wta$\}$
- $FTA = \{\varphi_M \mid M$ is a fta$\}$
- $BOT^f(\underline{A}) = \{\varphi_M \mid M$ is a fwp wta over $\underline{A}\}$
- $HOM^f(\underline{A}) = \{\varphi_M \mid M$ is a fwp homomorphism wta over $\underline{A}\}$

Definition

A wta $M = (Q, \Sigma, \underline{A}, \mu, F)$ is called final weight preserving (fwp) iff $F_q = id_A$ for every $q \in Q$.

Definition

Let $\underline{A}$ be M-monoid.

- $BOT(\underline{A}) = \{\varphi_M \mid M$ is a wta over $\underline{A}\}$
- $HOM(\underline{A}) = \{\varphi_M \mid M$ is a homomorphism wta over $\underline{A}\}$
- $REL = \{\varphi_M \mid M$ is a relabeling wta$\}$
- $FTA = \{\varphi_M \mid M$ is a fta$\}$
- $BOT^f(\underline{A}) = \{\varphi_M \mid M$ is a fwp wta over $\underline{A}\}$
- $HOM^f(\underline{A}) = \{\varphi_M \mid M$ is a fwp homomorphism wta over $\underline{A}\}$

Definition

A wta $M = (Q, \Sigma, \underline{A}, \mu, F)$ is called final weight preserving (fwp) iff $F_q = id_A$ for every $q \in Q$.

Summary

Main Result

Theorem (Stüber '06)

For every M-monoid $\underline{A}$:

$$BOT^f(\underline{A}) \subseteq REL \circ FTA \circ HOM^f(\underline{A}) \ .$$

This implies

Theorem (Engelfriet '75)

$BOT_{tt} \subseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

because of

Theorem (Maletti '05)

Every bottom-up tree transducer can be simulated by a wta.

Summary

Main Result

Theorem (Stüber '06)

For every M-monoid $\underline{A}$:

$$BOT^{f}(\underline{A}) \subseteq REL \circ FTA \circ HOM^{f}(\underline{A}) \ .$$

This implies

Theorem (Engelfriet '75)

$BOT_{tt} \subseteq REL_{tt} \circ FTA_{tt} \circ HOM_{tt}$

because of

Theorem (Maletti '05)

Every bottom-up tree transducer can be simulated by a wta.

Future work

Future work

- For every M-monoid $\underline{A}$
  $BOT^{f}(\underline{A}) \supseteq REL \circ FTA \circ HOM^{f}(\underline{A})$.
- Determine all $\underline{A}$ such that
  $BOT(\underline{A}) = REL \circ FTA \circ HOM(\underline{A})$.

References I

J. Engelfriet.
Bottom-up and top-down tree transformations - a comparison.
Math. Systems Theory, 9(3):198–231, 1975.

Z. Fülöp, A. Maletti, H. Vogler.
Weighted tree automata over multioperator monoids, 2006.

A. Maletti.
Relating tree series transducers and weighted tree automata.
Foundations of Computer Science, 16(4):723–741, 2005.