Weighted Bottom-up and Top-down Tree Transformations are Incomparable

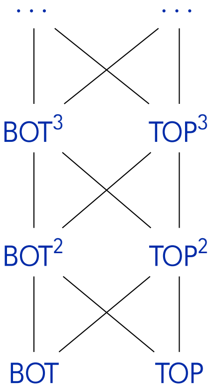
Andreas Maletti   Andreea-Teodora Nász

Faculty of Mathematics and Computer Science
Universität Leipzig, Germany
{maletti,nasz}@informatik.uni-leipzig.de

Famagusta, Cyprus — September 20, 2023

Bottom-up and Top-down Composition Hierarchy

Composition hierarchy [Engelfriet 1982]

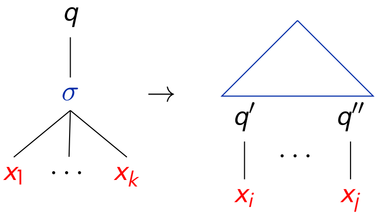


BOT — Bottom-up Tree Translations TOP — Top-down Tree Translations

Top-down Tree Transducer

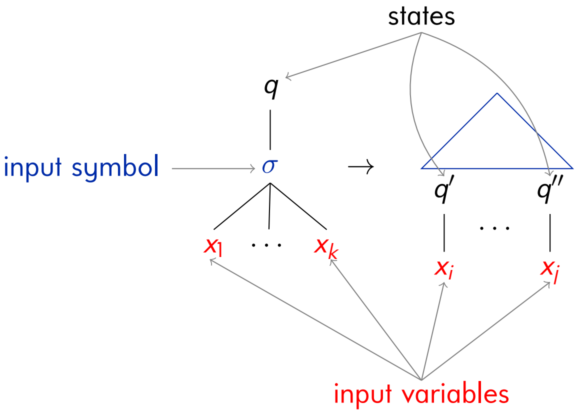
Rule shape TOP

[Rounds 1968], [Thatcher 1970]

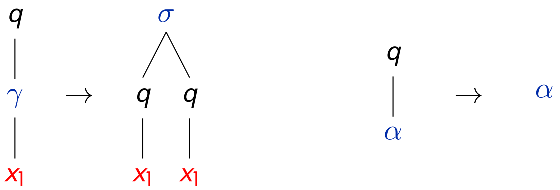


Rule shape TOP

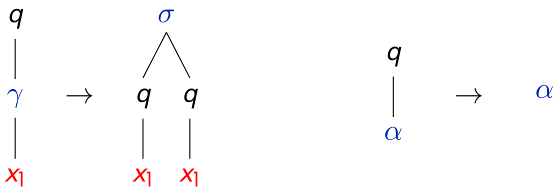
[Rounds 1968], [Thatcher 1970]



TOP example rules



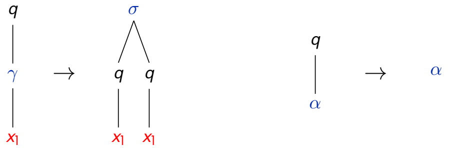
TOP example rules



Notes

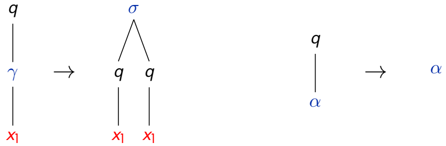
- 1st rule copies input tree
- Copies of input tree are independently processed

Top-down Tree Transducer

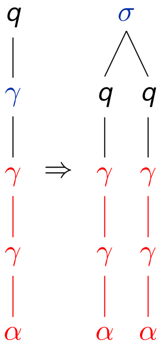


TOP example derivation

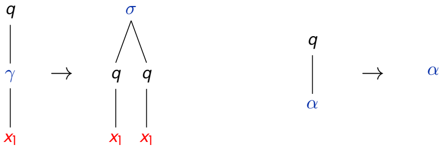
Top-down Tree Transducer



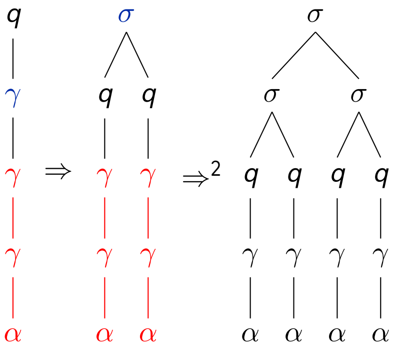
TOP example derivation

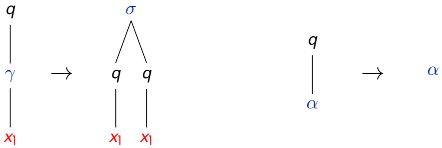


Top-down Tree Transducer

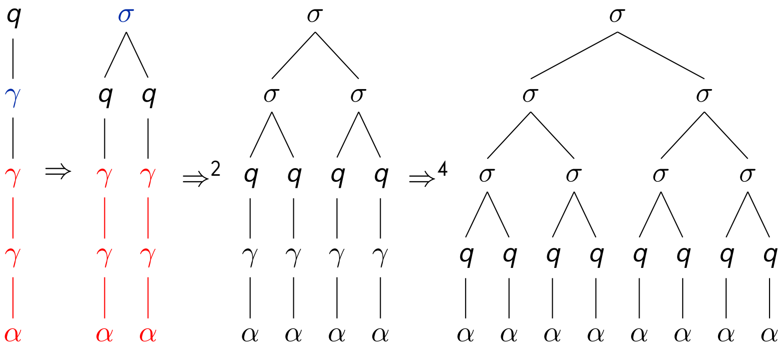


TOP example derivation

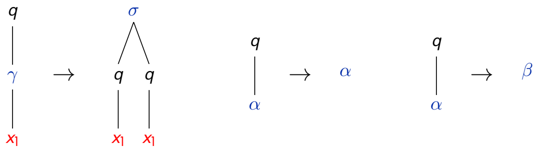




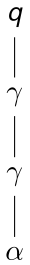
TOP example derivation



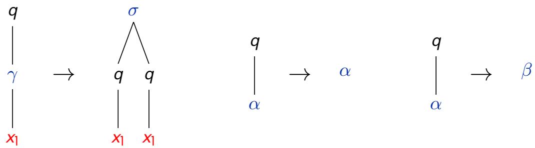
Top-down Tree Transducer



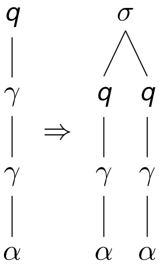
TOP example derivation

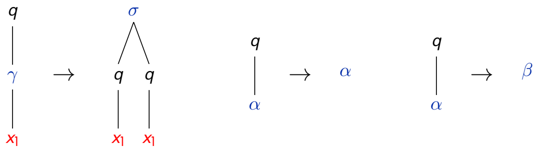


Top-down Tree Transducer

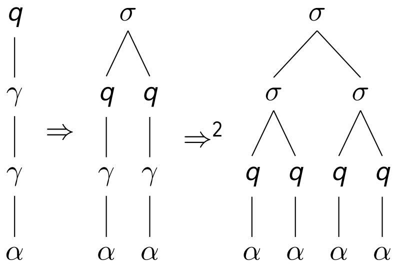


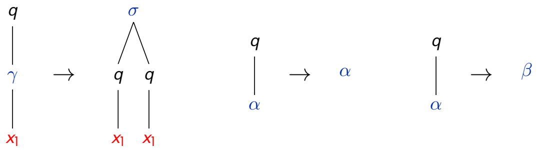
TOP example derivation



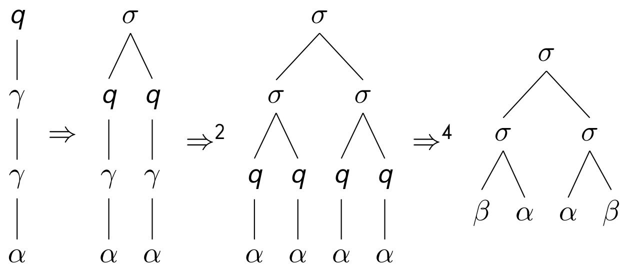


TOP example derivation

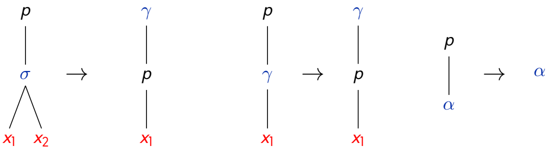




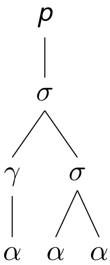
TOP example derivation



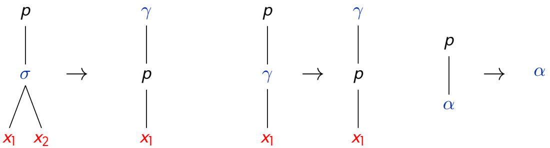
Top-down Tree Transducer



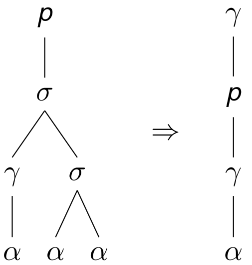
TOP example derivation

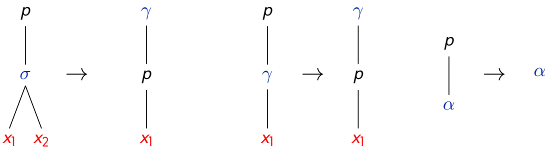


Top-down Tree Transducer

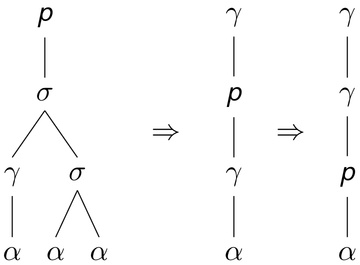


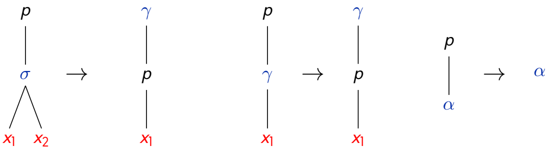
TOP example derivation



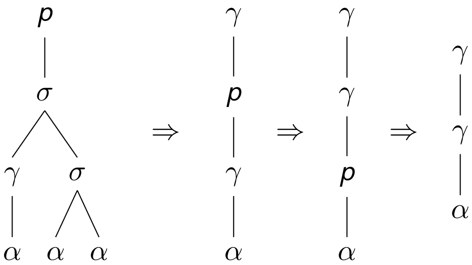


TOP example derivation





TOP example derivation



Top-down Tree Transducer

General features TOP [Engelfriet 1975]

✓ Can copy and nondeterministically process copies
(copies input subtrees)

Top-down Tree Transducer

General features TOP [Engelfriet 1975]

- ✓ Can copy and nondeterministically process copies
  (copies input subtrees)
- ✗ Cannot output identical trees for nondetermininstically processed inputs
  (except by chance) (due to independent processing of copies)

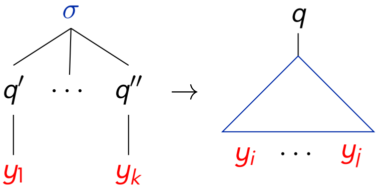
Top-down Tree Transducer

General features TOP [Engelfriet 1975]

- ✓ Can copy and nondeterministically process copies
  (copies input subtrees)
- ✗ Cannot output identical trees for nondetermininstically processed inputs (except by chance) (due to independent processing of copies)
- ✗ Cannot restrict subtrees before deletion
  (deletes by ignoring input subtrees)

Rule shape BOT

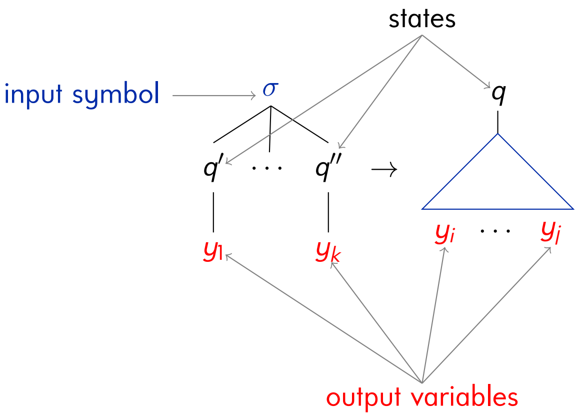
[Rounds 1968], [Engelfriet 1975]



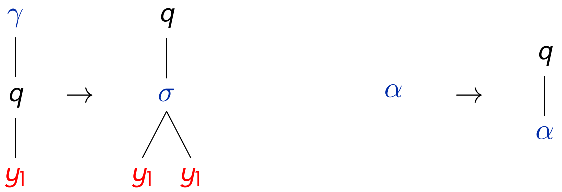
Bottom-up Tree Transducer

Rule shape BOT

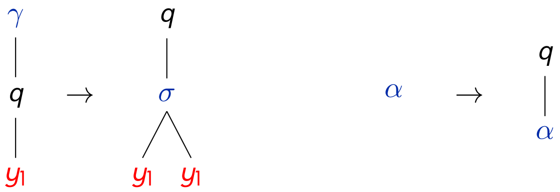
[Rounds 1968], [Engelfriet 1975]



BOT example rules



BOT example rules



Notes

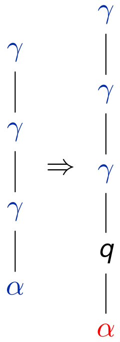
- 1st rule copies output tree
- Results in exact copies of output

Bottom-up Tree Transducer

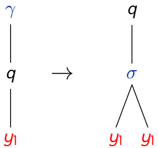
BOT example derivation

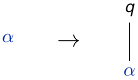
Bottom-up Tree Transducer

BOT example derivation

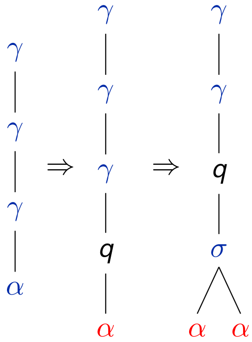


Bottom-up Tree Transducer



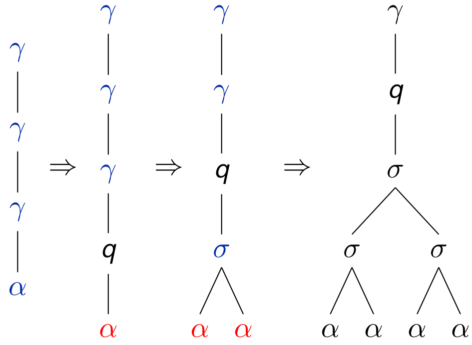


BOT example derivation



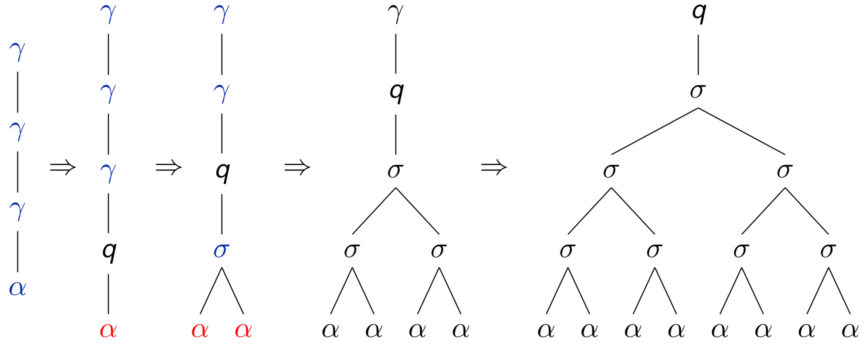
Bottom-up Tree Transducer

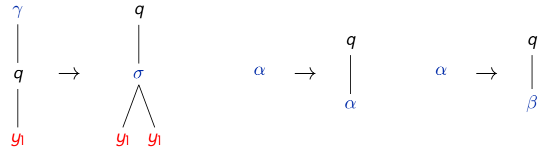
BOT example derivation



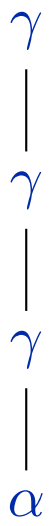
Bottom-up Tree Transducer

BOT example derivation

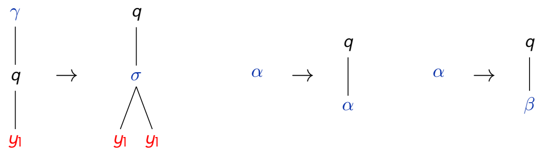




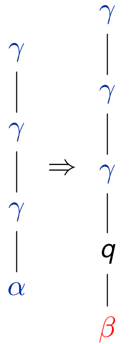
BOT example derivation

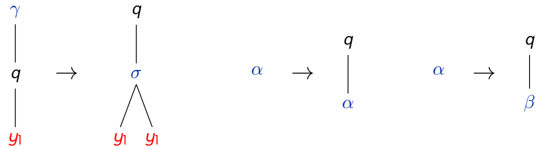


Bottom-up Tree Transducer

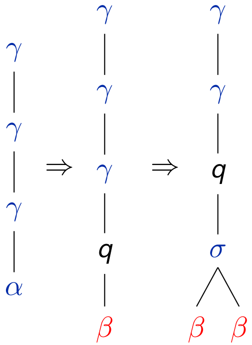


BOT example derivation

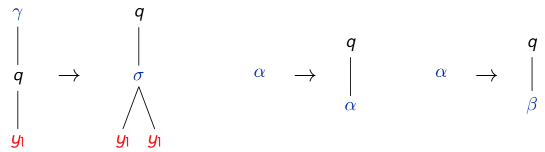




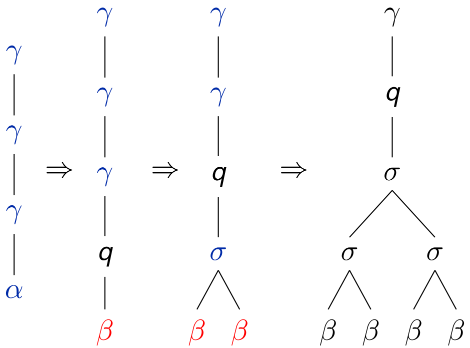
BOT example derivation



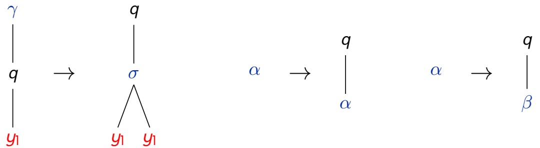
Bottom-up Tree Transducer



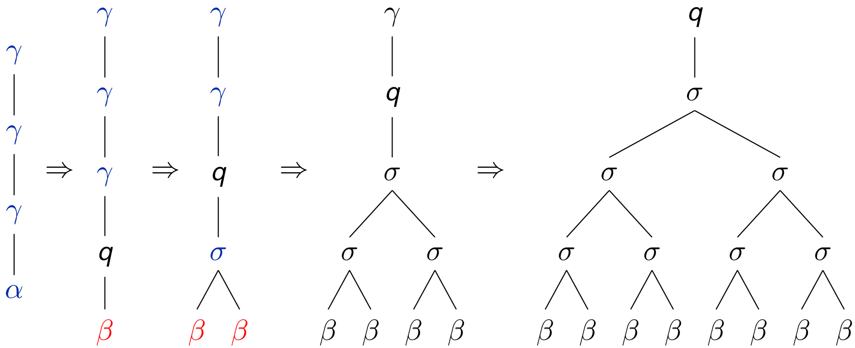
BOT example derivation

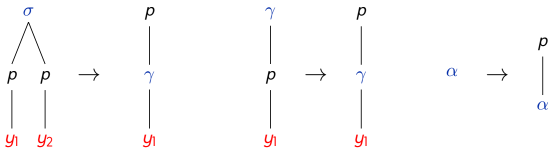


Bottom-up Tree Transducer

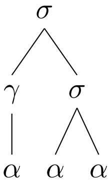


BOT example derivation

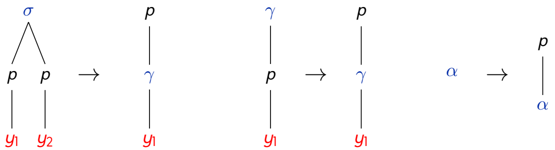




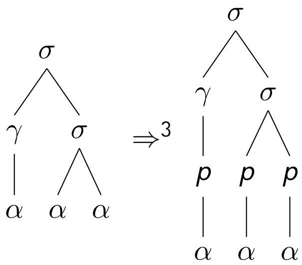
BOT example derivation



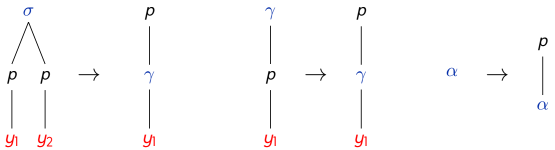
Bottom-up Tree Transducer



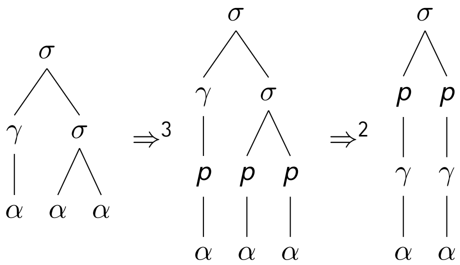
BOT example derivation



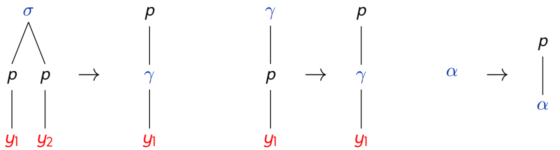
Bottom-up Tree Transducer



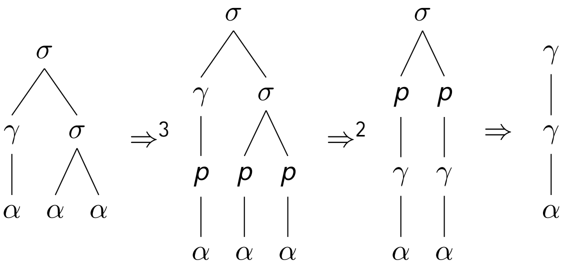
BOT example derivation



Bottom-up Tree Transducer



BOT example derivation



Bottom-up Tree Transducer

General features BOT [Engelfriet 1975]

- ✓ Can inspect input subtrees before deletion
  (deletes output subtrees)

Bottom-up Tree Transducer

General features BOT [Engelfriet 1975]

- ✓ Can inspect input subtrees before deletion
  (deletes output subtrees)
- ✓ Can output identical trees for nondetermininstically processed inputs
  (copies output subtrees)

Bottom-up Tree Transducer

General features BOT [Engelfriet 1975]

- ✓ Can inspect input subtrees before deletion
  (deletes output subtrees)
- ✓ Can output identical trees for nondetermininstically processed inputs
  (copies output subtrees)
- ✗ Cannot output different translations of same input subtree
  (cannot copy input subtrees)

Weights

Extension to TOP and BOT [Kuich 1999], [Engelfriet, Fülöp, Vogler 2002]

- Weights used in practice to resolve nondeterminism
- Each rule is assigned weight
- Weights multiplied along derivation
- Weights of alternatives are added

Typical weights

- Probabilities
- Costs
- Flows
- Profits

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Weight structure

Definition

Commutative semiring $(C, +, \cdot, 0, 1)$ if

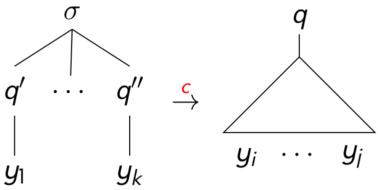
- $(C, +, 0)$ and $(C, \cdot, 1)$ commutative monoids
- $\cdot$ distributes over finite (incl. empty) sums

Example

- Boolean semiring $(\{0, 1\}, \max, \min, 0, 1)$ (unweighted case)
- semiring $(\mathbb{N}, +, \cdot, 0, 1)$ of nonnegative integers
- field $(\mathbb{Q}, +, \cdot, 0, 1)$ of rational numbers
- any field, ring, etc.

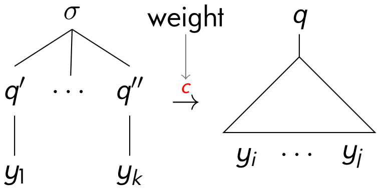
Weighted Bottom-up Tree Transducer

Rule shape Weighted BOT [Engelfriet, Fülöp, Vogler 2002]



Weighted Bottom-up Tree Transducer

Rule shape Weighted BOT [Engelfriet, Fülöp, Vogler 2002]



Weighted Extended Top-down Tree Transducer

Adjustment of semantics

- Weight of derivation is product of rule weights
  (rule weight taken as often as rule is used)

Weighted Extended Top-down Tree Transducer

Adjustment of semantics

- Weight of derivation is product of rule weights
  (rule weight taken as often as rule is used)
- Weight of translation is sum of all derivations for that translation
  (actually only left-most derivations to normalize rewrite order)

Weighted Composition

Given $\tau_1 \colon T_\Sigma \times T_\Gamma \to C$ and $\tau_2 \colon T_\Gamma \times T_\Delta \to C$

$$(\tau_1 \,;\, \tau_2)(s, u) = \sum_{t \in T_\Gamma} \tau_1(s, t) \cdot \tau_2(t, u)$$

Weighted Composition

Given $\tau_1 \colon T_\Sigma \times T_\Gamma \to C$ and $\tau_2 \colon T_\Gamma \times T_\Delta \to C$

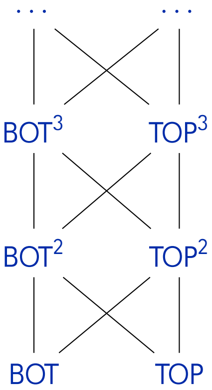
$$(\tau_1 \,;\, \tau_2)(s, u) = \sum_{t \in T_\Gamma} \tau_1(s, t) \cdot \tau_2(t, u)$$

Note

- Weighted composition is standard matrix product
  (identify $\tau_1 \in C^{T_\Sigma \times T_\Gamma}$ and $\tau_2 \in C^{T_\Gamma \times T_\Delta}$)

Weighted Composition Hierarchy

Existing hierarchy [Fülöp, Gazdag, Vogler 2004]



Weight structure **must not** be ring

(Non-rings permit semiring homomorphism into Boolean semiring)

Weighted Composition Hierarchy

Notes

- Results for rings or fields sometimes “better” than in unweighted case
- Automata (NFA) minimization
  - in P for fields [Flouret, Laugerotte 1997]
  - Unweighted: PSpace-complete [Jiang, Ravikumar 1993]

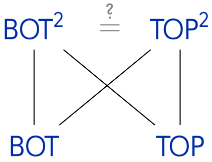
Weighted Composition Hierarchy

Notes

- Results for rings or fields sometimes “better” than in unweighted case
- Automata (NFA) minimization
  - in P for fields [Flouret, Laugerotte 1997]
  - Unweighted: PSpace-complete [Jiang, Ravikumar 1993]
- HOM-problem
  - in NL for certain semirings that embed into field [M., Nász, Paul 2023]
  - Unweighted: ExpTime-complete [Creus, Gascón, Godoy, Ramos 2016]

Weighted Composition Hierarchy

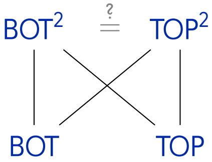
Main result



for all commutative semirings

Weighted Composition Hierarchy

Main result



for all commutative semirings

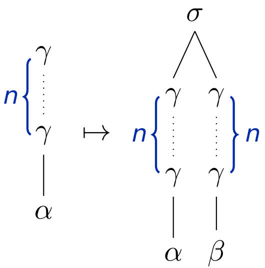
Main proof obligations

TOP $\not\subseteq$ BOT   BOT $\not\subseteq$ TOP

Weighted Composition Hierarchy

Obligation TOP $\not\subseteq$ BOT

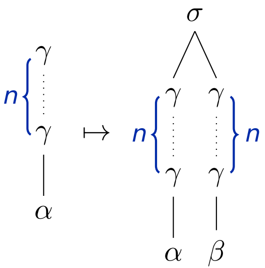
- Straightforward due to BOT's structural inability to output different trees for same input subtree
- Support (nonzero-weighted pairs) cannot be mapping



Weighted Composition Hierarchy

Obligation TOP $\not\subseteq$ BOT

- Straightforward due to BOT's structural inability to output different trees for same input subtree
- Support (nonzero-weighted pairs) cannot be mapping

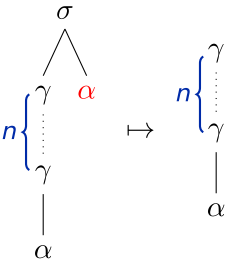


- Easily possible with weighted TOP

Weighted Composition Hierarchy

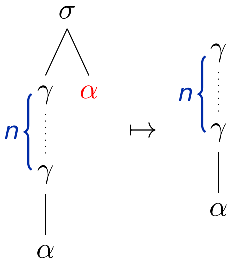
Obligation BOT $\not\subseteq$ TOP

- Utilize TOP's inability to inspect subtrees before deletion
- Support (nonzero-weighted pairs) cannot be mapping f



Obligation BOT $\not\subseteq$ TOP

- Utilize TOP's inability to inspect subtrees before deletion
- Support (nonzero-weighted pairs) cannot be mapping *f*



- Easily possible with weighted BOT

Weighted Composition Hierarchy

Proof.

- Suppose TOP computes τ with $\mathrm{supp}(\tau) = f$
- Let n be maximal size of rhs

Weighted Composition Hierarchy

Proof.

- Suppose TOP computes τ with $\text{supp}(\tau) = f$
- Let n be maximal size of rhs
- Consider inputs $s = \sigma(t, \alpha)$ and $s' = \sigma(t, \beta)$ with $t = \gamma^{2n+1}(\alpha)$

$$(s, t) \in f = \text{supp}(\tau) \qquad\qquad (s', t) \notin f = \text{supp}(\tau) \tag{1}$$

Weighted Composition Hierarchy

Proof.

- Suppose TOP computes τ with $\mathrm{supp}(\tau) = f$
- Let n be maximal size of rhs
- Consider inputs $s = \sigma(t, \alpha)$ and $s' = \sigma(t, \beta)$ with $t = \gamma^{2n+1}(\alpha)$

$$(s, t) \in f = \mathrm{supp}(\tau) \qquad\qquad (s', t) \notin f = \mathrm{supp}(\tau) \tag{1}$$

- Show $\tau(s, t) = \tau(s', t)$ by showing that they have same derivations

Weighted Composition Hierarchy

Proof.

- Suppose TOP computes τ with $\text{supp}(\tau) = f$
- Let n be maximal size of rhs
- Consider inputs $s = \sigma(t, \alpha)$ and $s' = \sigma(t, \beta)$ with $t = \gamma^{2n+1}(\alpha)$

$$(s, t) \in f = \text{supp}(\tau) \qquad\qquad (s', t) \notin f = \text{supp}(\tau) \tag{1}$$

- Show $\tau(s, t) = \tau(s', t)$ by showing that they have same derivations
- Case 1: Initial rule deletes 1st subtree. Then TOP applies 2 rules but output t too large $\lightning$. No derivations in either case

Weighted Composition Hierarchy

Proof.

- Suppose TOP computes τ with $\text{supp}(\tau) = f$
- Let n be maximal size of rhs
- Consider inputs $s = \sigma(t, \alpha)$ and $s' = \sigma(t, \beta)$ with $t = \gamma^{2n+1}(\alpha)$

$$(s, t) \in f = \text{supp}(\tau) \qquad\qquad (s', t) \notin f = \text{supp}(\tau) \tag{1}$$

- Show $\tau(s, t) = \tau(s', t)$ by showing that they have same derivations
- Case 1: Initial rule deletes 1st subtree. Then TOP applies 2 rules but output t too large $\lightning$. No derivations in either case
- Case 2: Initial rule deletes 2nd subtree. Since s and s' only differ in 2nd subtree, same derivations in both cases

Weighted Composition Hierarchy

Proof.

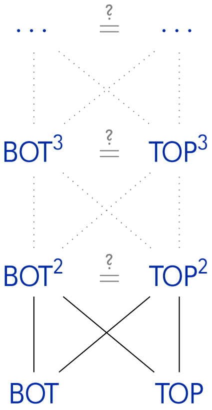
- Suppose TOP computes τ with $\text{supp}(\tau) = f$
- Let n be maximal size of rhs
- Consider inputs $s = \sigma(t, \alpha)$ and $s' = \sigma(t, \beta)$ with $t = \gamma^{2n+1}(\alpha)$

$$(s, t) \in f = \text{supp}(\tau) \qquad (s', t) \notin f = \text{supp}(\tau) \tag{1}$$

- Show $\tau(s, t) = \tau(s', t)$ by showing that they have same derivations
- Case 1: Initial rule deletes 1st subtree. Then TOP applies 2 rules but output t too large $\lightning$. No derivations in either case
- Case 2: Initial rule deletes 2nd subtree. Since s and s' only differ in 2nd subtree, same derivations in both cases
- $\tau(s, t) = \tau(s', t)$ contradicts (1) $\square$

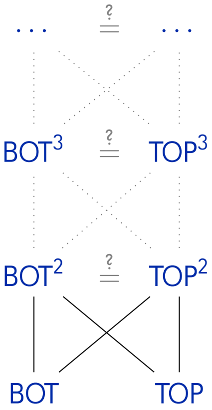
Summary

Status quo



Summary

Status quo



Thank you for your attention!