MSO Queries on Trees

Enumerating Answers under Updates Using Forest Algebras

Matthias Niewerth

UNIVERSITÄT BAYREUTH

Problem Description

Query Answering

Given: MSO Query $\Psi(x_1, \dots, x_k)$, tree t

Answer: $\{\ (v_1, \dots, v_k)\ \mid\ t \models \Psi(v_1, \dots, v_k)\ \}$

Problem Description

Query Answering

Given: MSO Query $\Psi(x_1, \ldots, x_k)$, tree t

Answer: $\{ (v_1, \ldots, v_k) \mid t \models \Psi(v_1, \ldots, v_k) \}$

- Problem: up to $|t|^k$ answers

Problem Description

Query Answering

Given: MSO Query $\Psi(x_1, \dots, x_k)$, tree t

Answer: $\{\ (v_1, \dots, v_k)\ \mid\ t \models \Psi(v_1, \dots, v_k)\ \}$

- Problem: up to $|t|^k$ answers
- Approach: Enumeration
  1. build index structure
  2. enumerate with small delay
  3. fast updates of index structure

Problem Description

Query Answering

Given: MSO Query $\Psi(x_1, \ldots, x_k)$, tree t

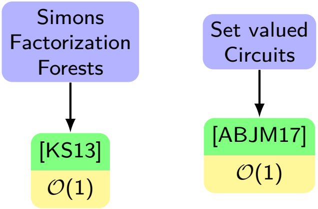
Answer: $\{\ (v_1, \ldots, v_k)\ \mid\ t \models \Psi(v_1, \ldots, v_k)\ \}$

- Problem: up to $|t|^k$ answers
- Approach: Enumeration
  1. build index structure
  2. enumerate with small delay
  3. fast updates of index structure
- Updates:
  - change of a label
  - insertion of a leaf
  - deletion of a leaf

Related Work

Reference
Delay
Update

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Reference

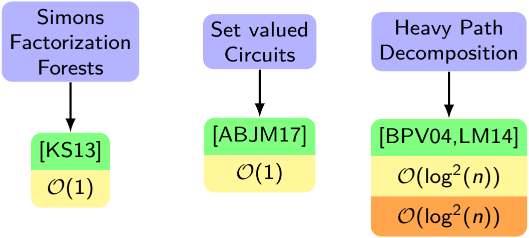
Delay

Update

Kazana and Segoufin
Enumeration of Monadic Second-Order Queries on Trees
TOCL 2013

Amarilli, Bourhis, Jachiet, Mengel
A Circuit-Based Approach to Efficient Enumeration
ICALP 2017

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

🟩 Reference

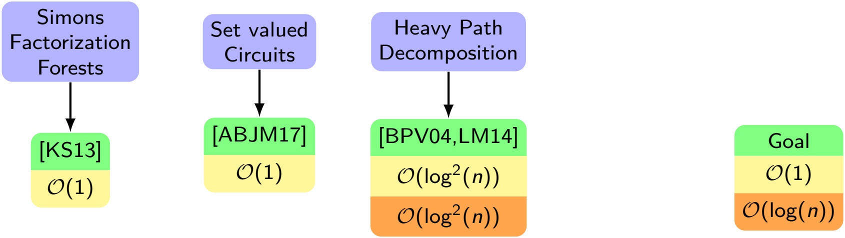
🟨 Delay

🟧 Update

Balmin, Papakonstantinou, Vianu
Incremental validation of XML documents
TODS 2004

Losemann and Martens
MSO Queries on Trees: Enumerating Answers under Updates
LICS 2014

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Goal

$\mathcal{O}(1)$

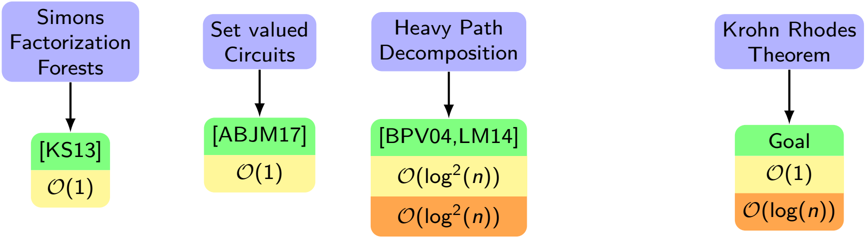
$\mathcal{O}(\log(n))$

Reference

Delay

Update

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Krohn Rhodes Theorem

Goal

$\mathcal{O}(1)$

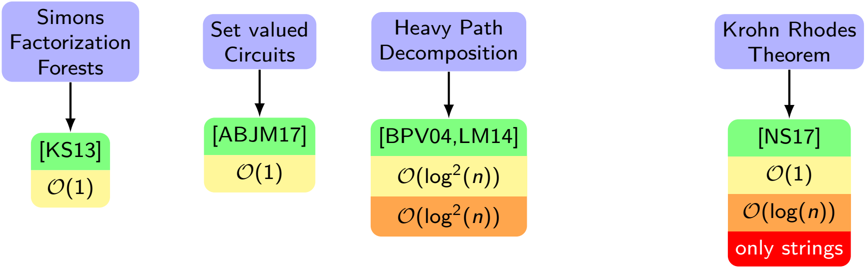
$\mathcal{O}(\log(n))$

Reference

Delay

Update

Related Work



Simons Factorization Forests → [KS13] $\mathcal{O}(1)$

Set valued Circuits → [ABJM17] $\mathcal{O}(1)$

Heavy Path Decomposition → [BPV04,LM14] $\mathcal{O}(\log^2(n))$ $\mathcal{O}(\log^2(n))$

Krohn Rhodes Theorem → [NS17] $\mathcal{O}(1)$ $\mathcal{O}(\log(n))$ only strings

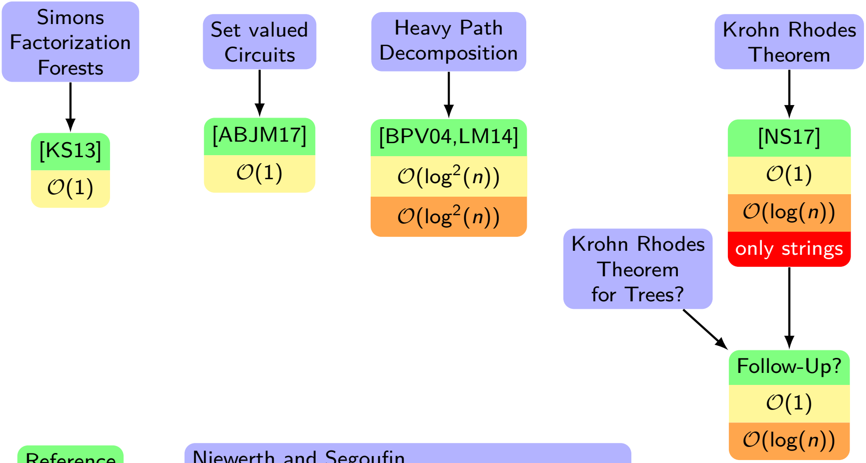
Reference

Delay

Update

Niewerth and Segoufin
Enumeration of MSO Queries on Strings with Constant Delay and Logarithmic Updates
PODS 2018

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

Follow-Up?

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

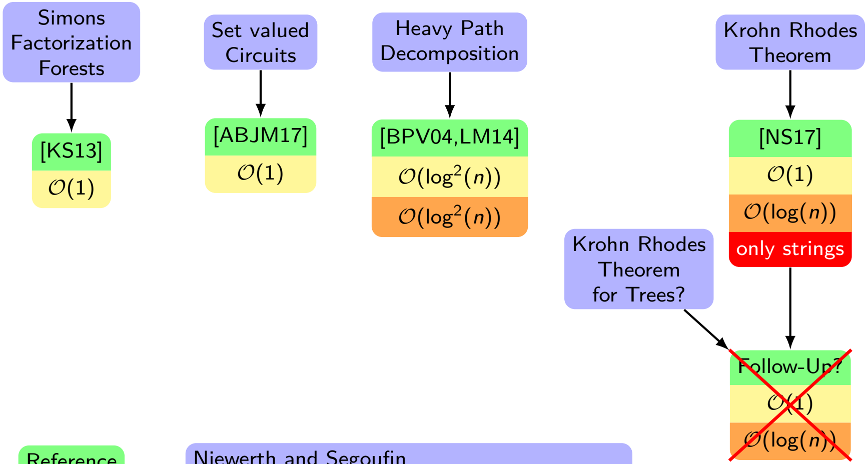
Reference

Delay

Update

Niewerth and Segoufin
Enumeration of MSO Queries on Strings with Constant Delay and Logarithmic Updates
PODS 2018

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

~~Follow-Up?~~

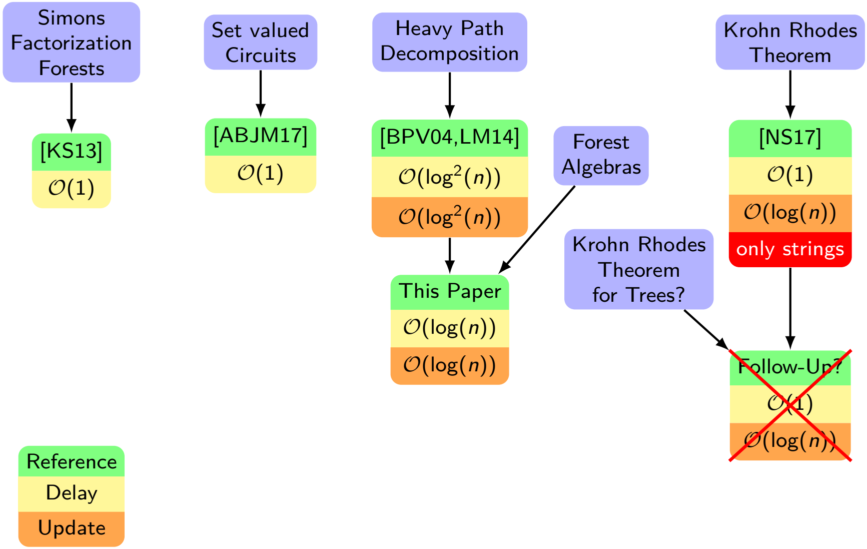
~~$\mathcal{O}(1)$~~

~~$\mathcal{O}(\log(n))$~~

Reference
Delay
Update

Niewerth and Segoufin
Enumeration of MSO Queries on Strings with Constant Delay and Logarithmic Updates
PODS 2018

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Forest Algebras

This Paper

$\mathcal{O}(\log(n))$

$\mathcal{O}(\log(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

~~Follow-Up?~~

~~$\mathcal{O}(1)$~~

~~$\mathcal{O}(\log(n))$~~

Reference

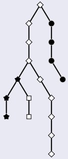
Delay

Update

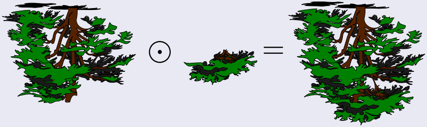
Contents

Monoids and Heavy Path Decomposition

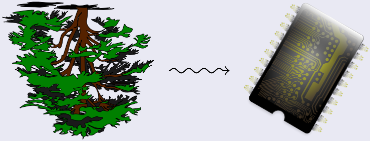
$(M, \odot, \varepsilon)$



Forest Algebras



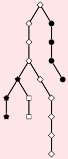
Enumeration using Circuits



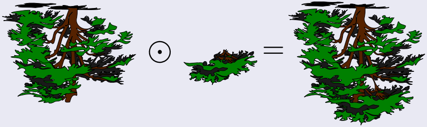
Contents

Monoids and Heavy Path Decomposition

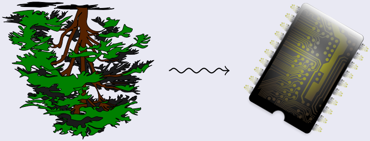
$$(M, \odot, \varepsilon)$$



Forest Algebras



Enumeration using Circuits



Boolean Queries: Monoids over Strings [BPV04]

Query Translation

MSO sentence Ψ → automaton A → transition monoid

Transition monoid: $(2^{Q^2}, \oplus, \mathrm{id}_Q\,)$

$m_1 \oplus m_2 = \{\ (q_1, q_3)\ \mid\ (q_1, q_2) \in m_1, (q_2, q_3) \in m_2\ \}$

$a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6 \quad a_7 \quad a_8$

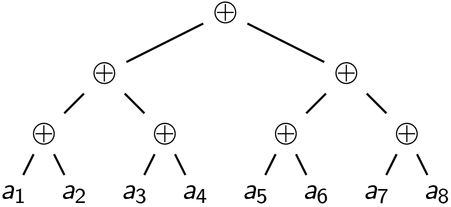
Boolean Queries: Monoids over Strings [BPV04]

Query Translation

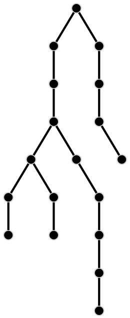
MSO sentence Ψ → automaton A → transition monoid

Transition monoid: $(2^{Q^2}, \oplus, \mathrm{id}_Q)$

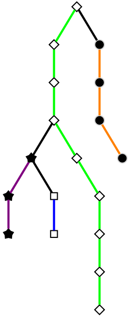
$m_1 \oplus m_2 = \{ (q_1, q_3) \mid (q_1, q_2) \in m_1, (q_2, q_3) \in m_2 \}$



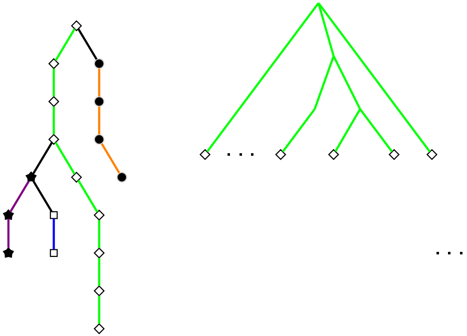
Heavy Path Decomposition [BPV04,LM14]



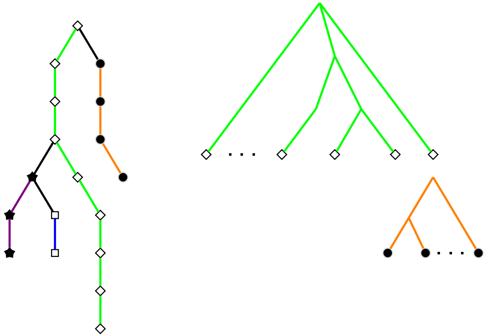
Heavy Path Decomposition [BPV04,LM14]



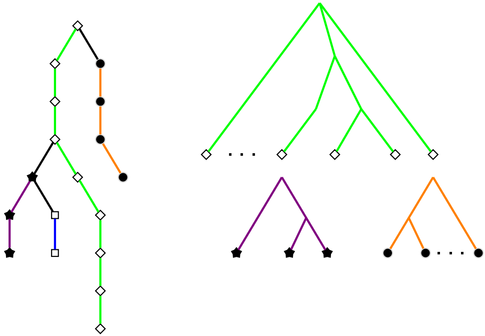
Heavy Path Decomposition [BPV04,LM14]



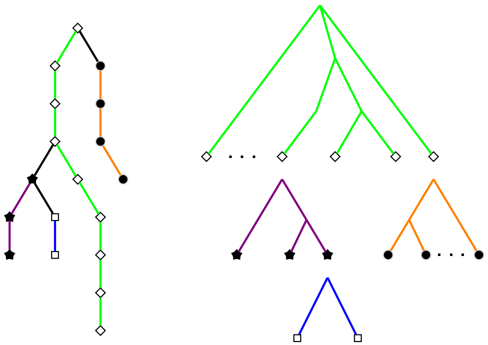
Heavy Path Decomposition [BPV04,LM14]



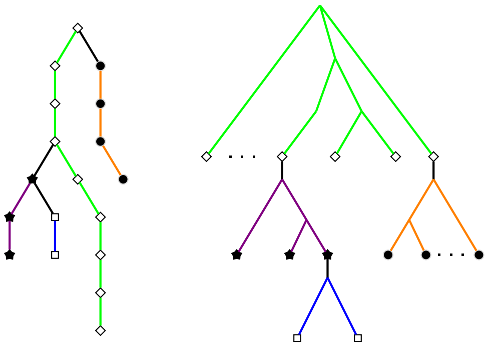
Heavy Path Decomposition [BPV04,LM14]



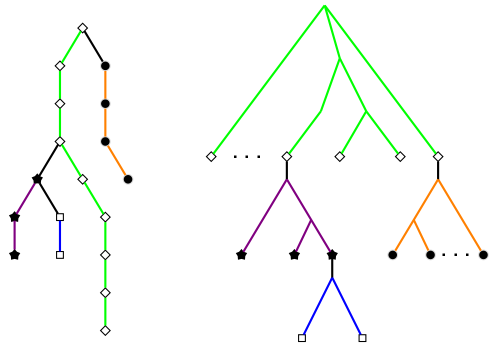
Heavy Path Decomposition [BPV04,LM14]



Heavy Path Decomposition [BPV04,LM14]



Heavy Path Decomposition [BPV04,LM14]



Results

Update time for Boolean queries:	$\mathcal{O}(\log^2(n))$	[BPV04]
Update time and delay for *k*-ary queries:	$\mathcal{O}(\log^2(n))$	[LM14]

From Boolean to k-ary Queries

Boolean Query

MSO sentence Ψ $\quad\rightarrow\quad$ automaton A

The approach works for strings and trees.

From Boolean to k-ary Queries

Boolean Query

MSO sentence Ψ → automaton A

k-ary Query

MSO formula $\Psi(x_1, \ldots, x_k)$ → node selecting automaton (A, S)

The approach works for strings and trees.

From Boolean to k-ary Queries

Boolean Query

MSO sentence $\Psi \quad \rightarrow \quad$ automaton A

k-ary Query

MSO formula $\Psi(x_1, \ldots, x_k) \rightarrow$ node selecting automaton (A, S)

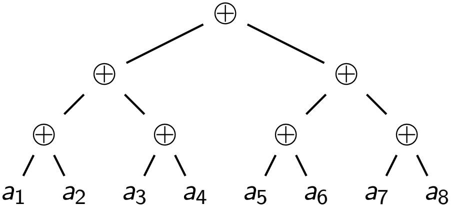
Node Selecting Automaton

A:	finite automaton
$S \subseteq Q^k$:	set of selecting tuples
Answer:	$\{ (v_1, \ldots, v_k) \mid \exists \text{ run } \lambda \text{ s.t. } (\lambda(v_1), \ldots, \lambda(v_k)) \in S \}$

The approach works for strings and trees.

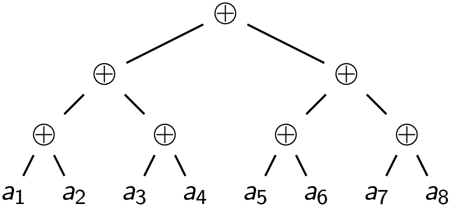
k-ary Queries: Monoids over Strings [LM14]

Transition monoid: $(2^{Q^2}, \oplus, \mathrm{id}_Q)$



k-ary Queries: Monoids over Strings [LM14]

Transition monoid: $(2^{Q^2 \times \mathbb{S}(S)}, \oplus, \mathrm{id}_Q \times \{\emptyset\})$



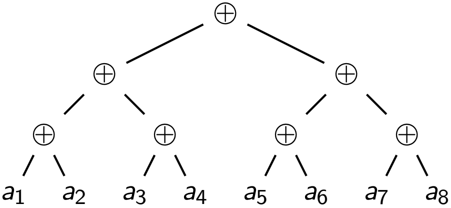
$\mathbb{S}(S) = \{Q' \subseteq Q \mid Q' \subseteq s \text{ for some } s \in S\}$

$$m_1 \oplus m_2 = \{ \; ((q_1, q_3), Q') \;\mid\; ((q_1, q_2), Q_1) \in m_1, \\ ((q_2, q_3), Q_2) \in m_2, Q' = Q_1 \cup Q_2 \; \}$$

Capture enough information about states to allow enumeration.

k-ary Queries: Monoids over Strings [LM14]

Transition monoid: $(2^{Q^2 \times \mathbb{S}(S)}, \oplus, \mathrm{id}_Q \times \{\emptyset\})$



Algorithm

1. Top-Down: search a position
2. Bottom-Up: filter tuples

$\mathbb{S}(S) = \{Q' \subseteq Q \mid Q' \subseteq s \text{ for some } s \in S\}$

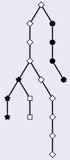
$$m_1 \oplus m_2 = \{ ((q_1, q_3), Q') \mid ((q_1, q_2), Q_1) \in m_1, \\ ((q_2, q_3), Q_2) \in m_2, Q' = Q_1 \cup Q_2 \}$$

Capture enough information about states to allow enumeration.

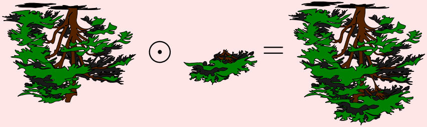
Contents

Monoids and Heavy Path Decomposition

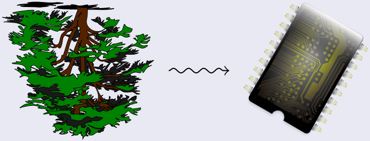
$(M, \odot, \varepsilon)$



Forest Algebras



Enumeration using Circuits

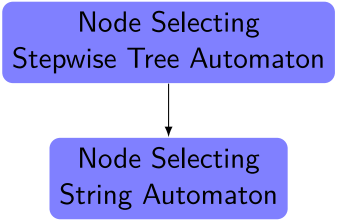


Structure of Algorithm

Structure of Algorithm

Node Selecting
Stepwise Tree Automaton

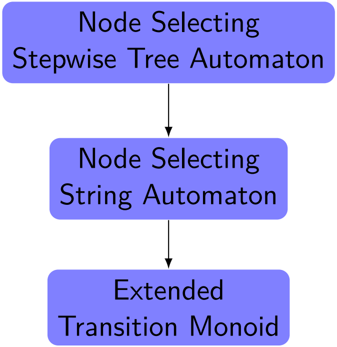
Structure of Algorithm



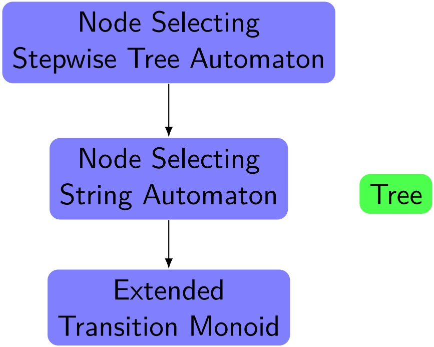
Node Selecting Stepwise Tree Automaton

Node Selecting String Automaton

Structure of Algorithm



Structure of Algorithm



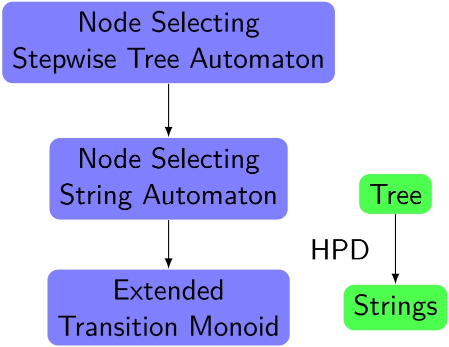
Node Selecting Stepwise Tree Automaton

Node Selecting String Automaton

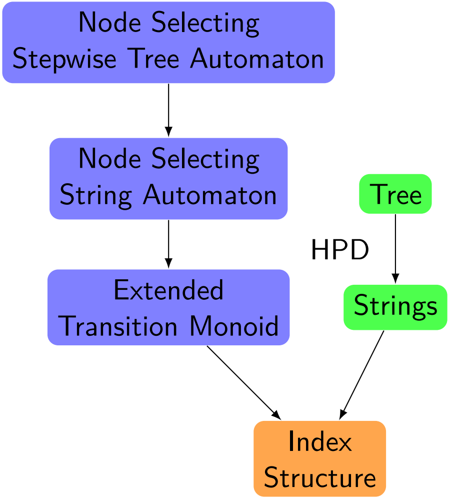
Extended Transition Monoid

Tree

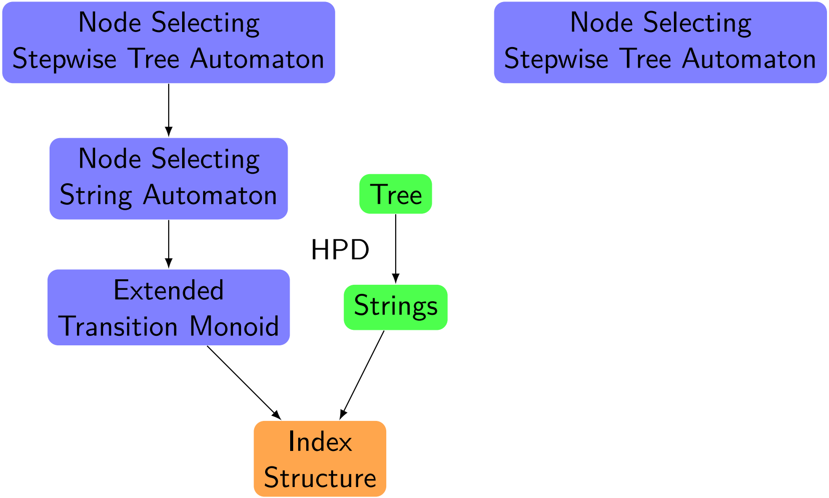
Structure of Algorithm



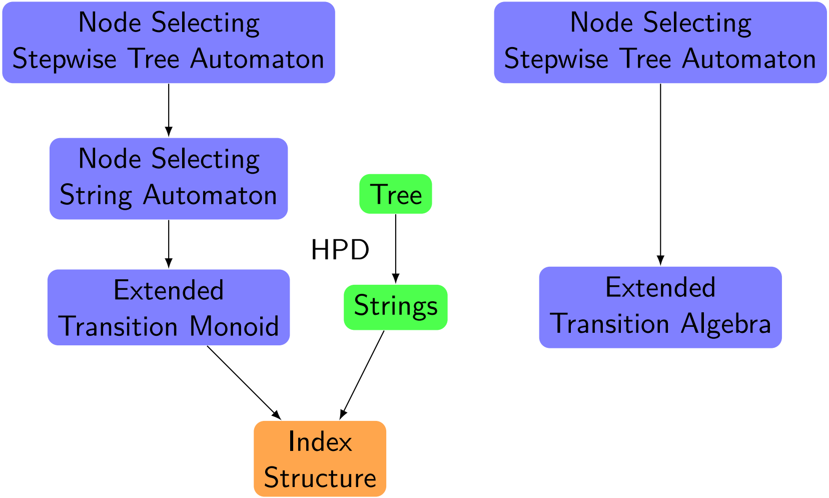
Structure of Algorithm



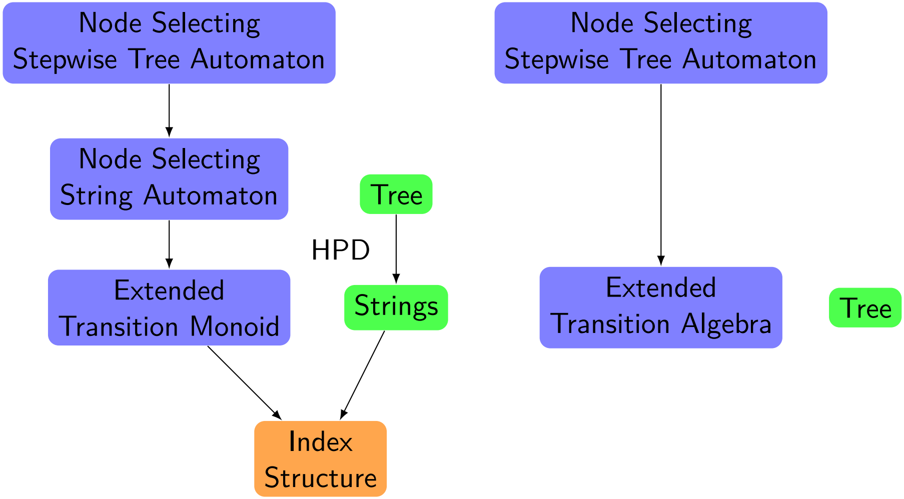
Structure of Algorithm



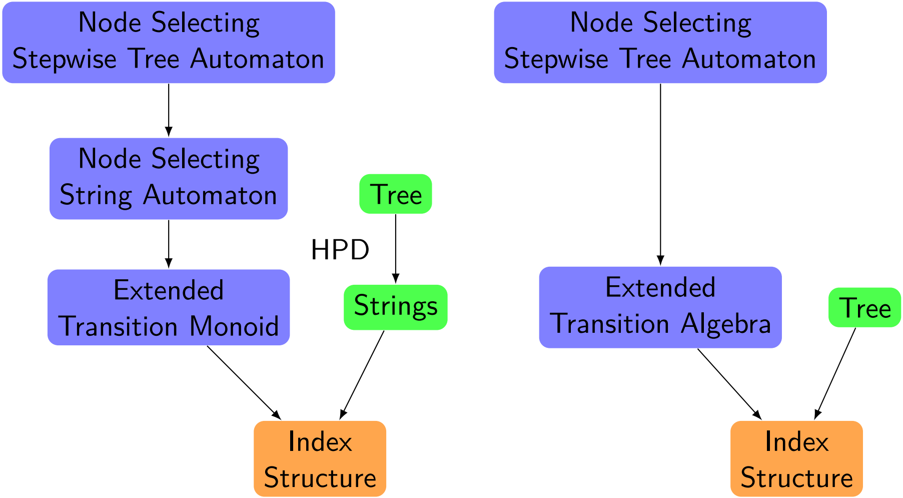
Structure of Algorithm



Structure of Algorithm

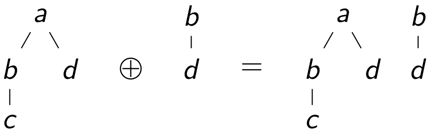


Structure of Algorithm


Node Selecting Stepwise Tree Automaton
Node Selecting String Automaton
Tree
HPD
Extended Transition Monoid
Strings
Index Structure
Node Selecting Stepwise Tree Automaton
Extended Transition Algebra
Tree
Index Structure

Forest Algebras in a Nutshell

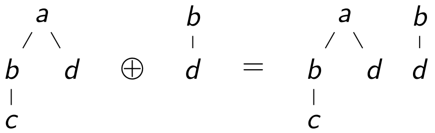
Examples



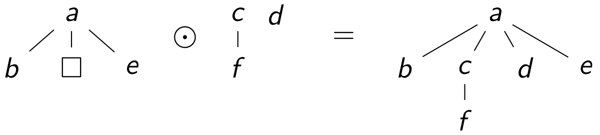
concatenation

Forest Algebras in a Nutshell

Examples



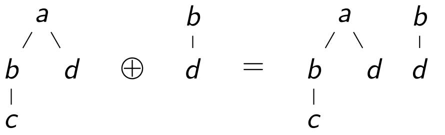
concatenation



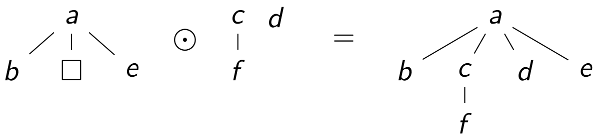
context application

Forest Algebras in a Nutshell

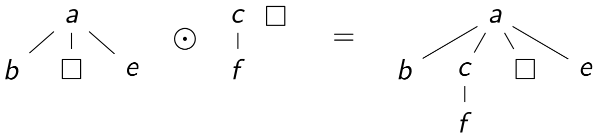
Examples



concatenation



context application



context application

Forest Algebras in a Nutshell

Free Forest Algebra

Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

Forest Algebras in a Nutshell

Free Forest Algebra

Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

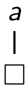
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \dots$	atomic forests
$a_\square, b_\square, \dots$	atomic contexts

Forest Algebras in a Nutshell

Free Forest Algebra

$a_\square =$



Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

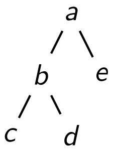
Atomic Formulas

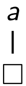
ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

Forest Algebras in a Nutshell

Free Forest Algebra

tree



$a_\square =$ 

Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

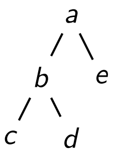
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

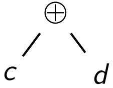
Forest Algebras in a Nutshell

Free Forest Algebra

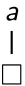
tree



formula



$a_\square$ =



Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

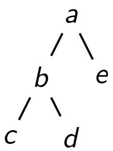
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

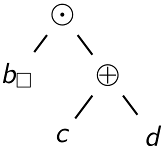
Forest Algebras in a Nutshell

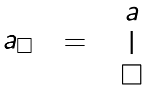
Free Forest Algebra

tree



formula





Free Forest Algebra

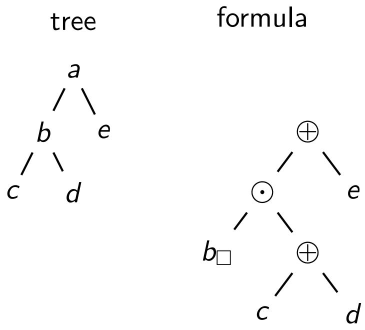
H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

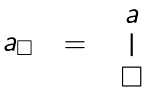
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

Forest Algebras in a Nutshell

Free Forest Algebra





Free Forest Algebra

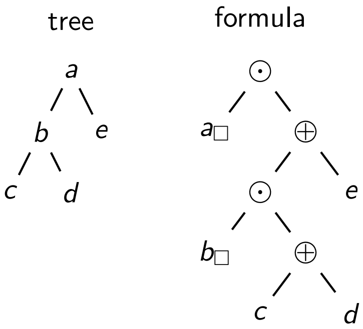
H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

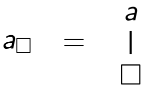
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

Forest Algebras in a Nutshell

Free Forest Algebra





Free Forest Algebra

H	all forests
V	all contexts
$\oplus$	concatenation
$\odot$	context application

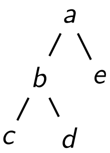
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \dots$	atomic forests
$a_\square, b_\square, \dots$	atomic contexts

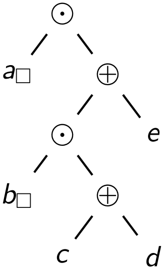
Forest Algebras in a Nutshell

Free Forest Algebra

tree



formula



$$a_\square \;=\; \begin{array}{c} a \\ | \\ \square \end{array}$$

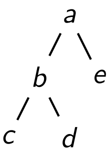
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

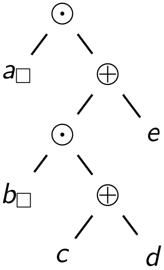
Forest Algebras in a Nutshell

Free Forest Algebra

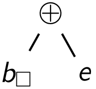
tree

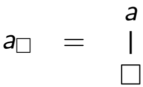


formula



alternative formula





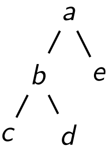
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

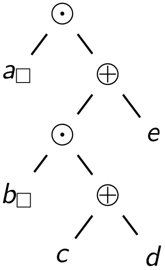
Forest Algebras in a Nutshell

Free Forest Algebra

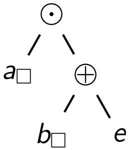
tree

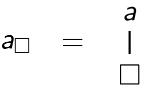


formula



alternative formula



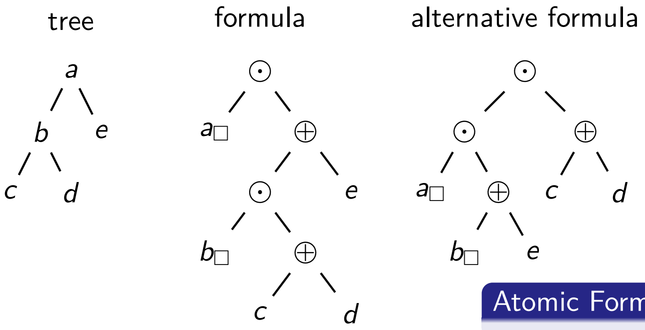


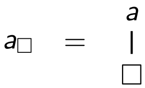
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

Forest Algebras in a Nutshell

Free Forest Algebra





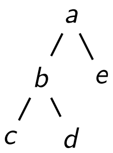
Atomic Formulas

ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

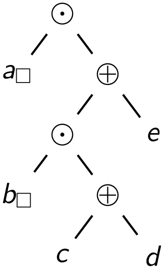
Forest Algebras in a Nutshell

Free Forest Algebra

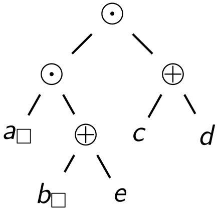
tree

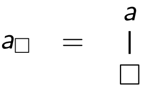


formula



alternative formula



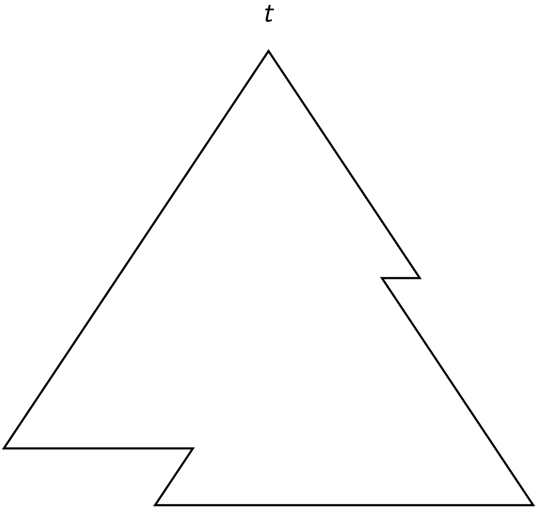


Atomic Formulas

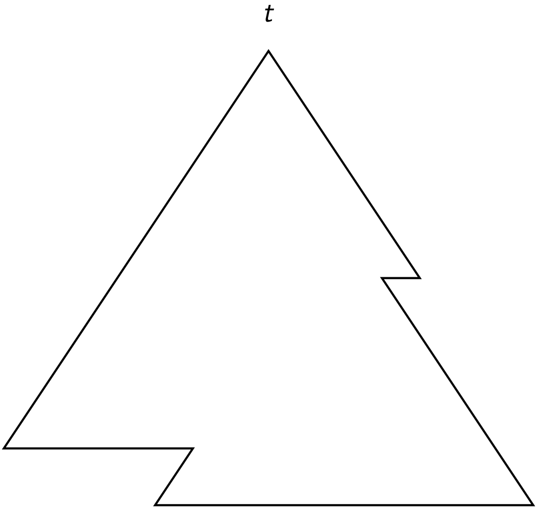
ε	empty forest
$\square$	empty context
$a, b, \ldots$	atomic forests
$a_\square, b_\square, \ldots$	atomic contexts

The leaves of the formula correspond to the nodes of the tree.

How to Decompose a Tree (to a log-depth Formula)

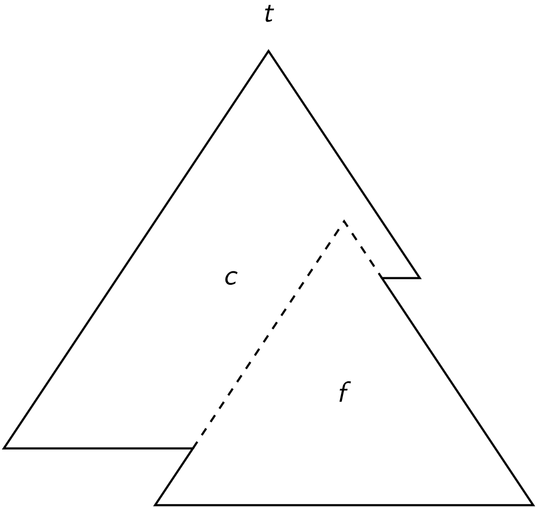


How to Decompose a Tree (to a log-depth Formula)



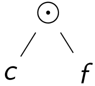
How to decompose a tree?

How to Decompose a Tree (to a log-depth Formula)

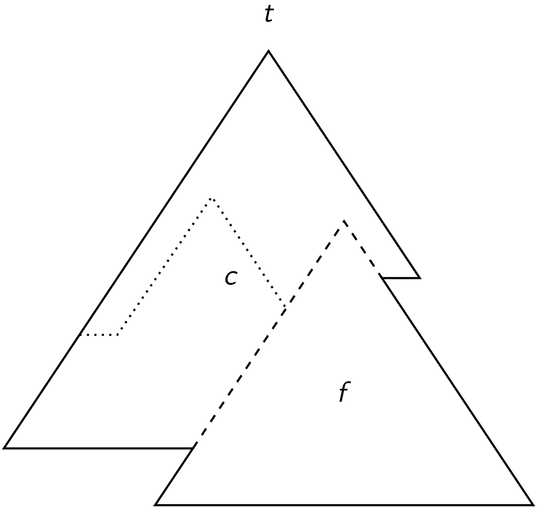


How to decompose a tree?

1. $t = c \odot f$

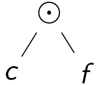


How to Decompose a Tree (to a log-depth Formula)

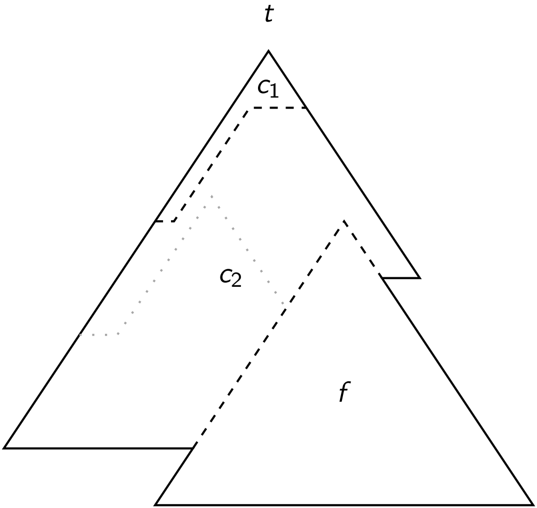


How to decompose a tree?

1. $t = c \odot f$

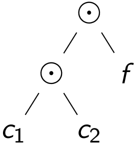


How to Decompose a Tree (to a log-depth Formula)

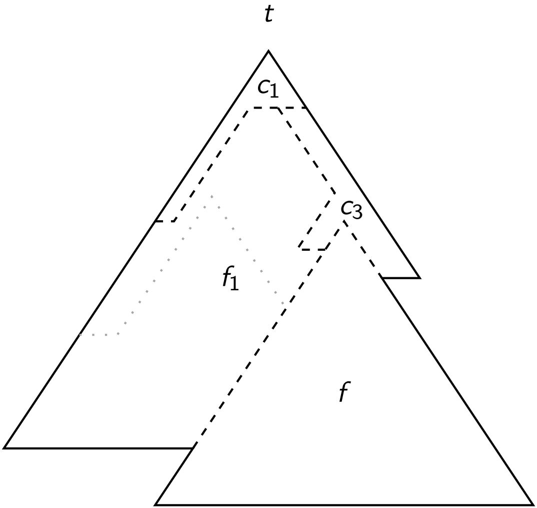


How to decompose a tree?

1. $t = c \odot f$
2. $c = c_1 \odot c_2$

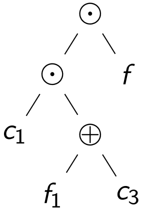


How to Decompose a Tree (to a log-depth Formula)

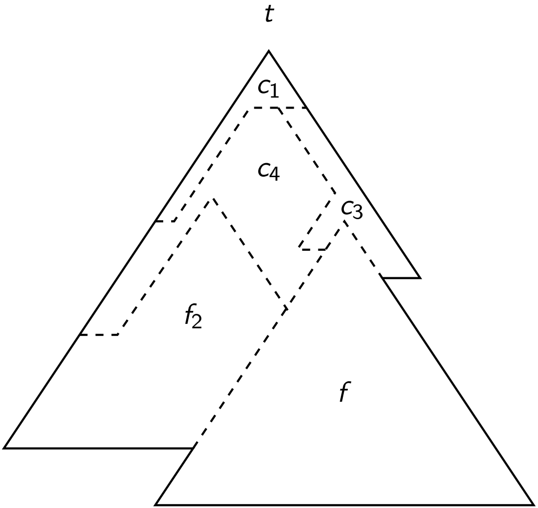


How to decompose a tree?

1. $t = c \odot f$
2. $c = c_1 \odot c_2$
3. $c_2 = f_1 \oplus c_3$

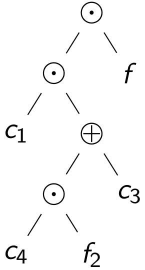


How to Decompose a Tree (to a log-depth Formula)

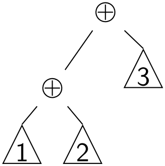


How to decompose a tree?

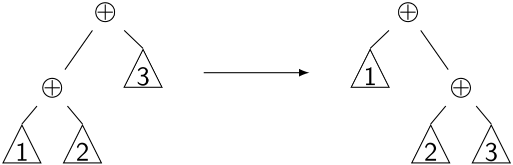
1. $t = c \odot f$
2. $c = c_1 \odot c_2$
3. $c_2 = f_1 \oplus c_3$
4. $f_1 = c_4 \odot f_2$



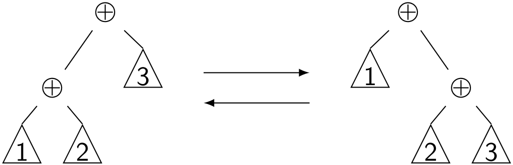
Rebalancing Forest Algebra Formulas



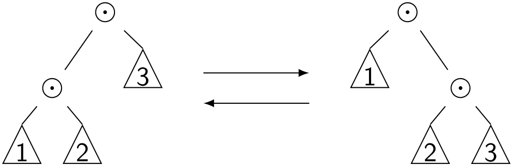
Rebalancing Forest Algebra Formulas



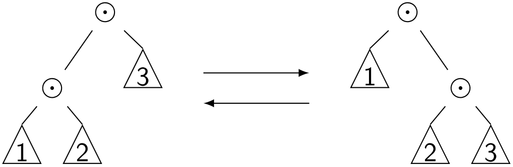
Rebalancing Forest Algebra Formulas

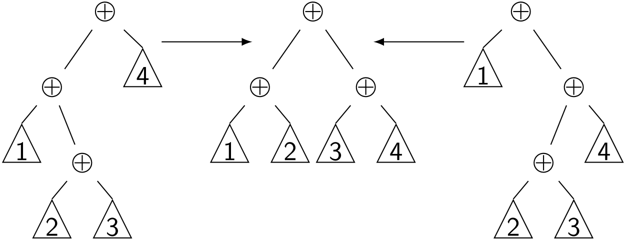


Rebalancing Forest Algebra Formulas

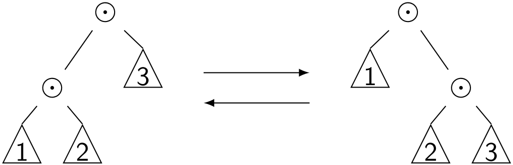


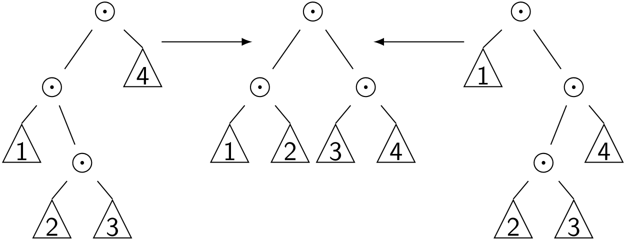
Rebalancing Forest Algebra Formulas



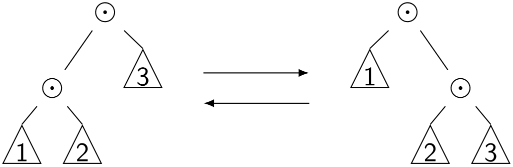


Rebalancing Forest Algebra Formulas

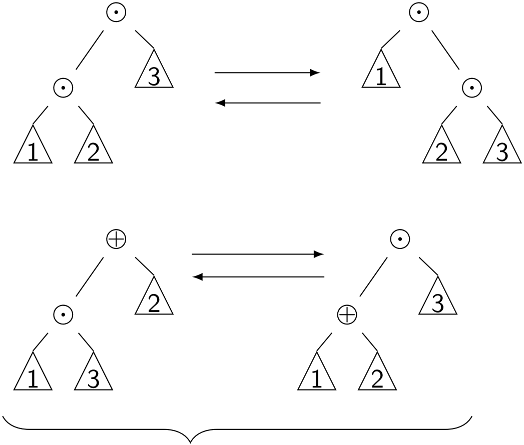




Rebalancing Forest Algebra Formulas

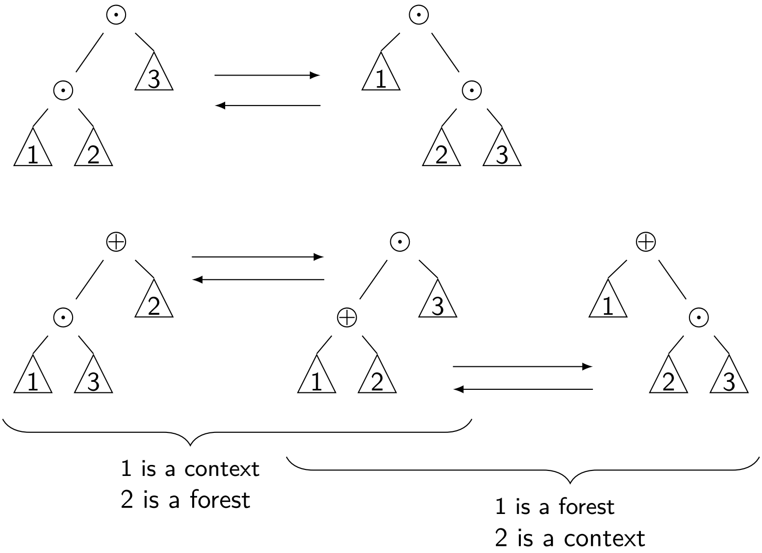


Rebalancing Forest Algebra Formulas



1 is a context
2 is a forest

Rebalancing Forest Algebra Formulas



1 is a context
2 is a forest

1 is a forest
2 is a context

Result

Theorem

Enumertion of MSO formulas on trees can be done in time:

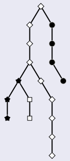
Preprocessing	$\mathcal{O}(\ \lvert Q\rvert^6 \cdot \lvert S\rvert \cdot 2^k \cdot n\ )$
Delay	$\mathcal{O}(\ \lvert Q\rvert^6 \cdot \lvert S\rvert \cdot 2^k \cdot k \cdot \log(n)\ )$
Updates	$\mathcal{O}(\ \lvert Q\rvert^6 \cdot \lvert S\rvert \cdot 2^k \cdot \log(n)\ )$

n	size of tree
$\lvert Q\rvert$	number of states
$\lvert S\rvert$	number of accepting tuples
k	arity of the query

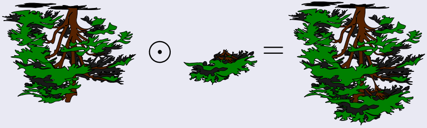
Contents

Monoids and Heavy Path Decomposition

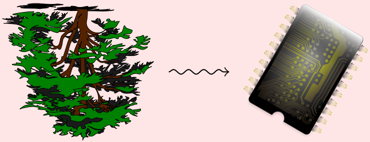
$$(M, \odot, \varepsilon)$$



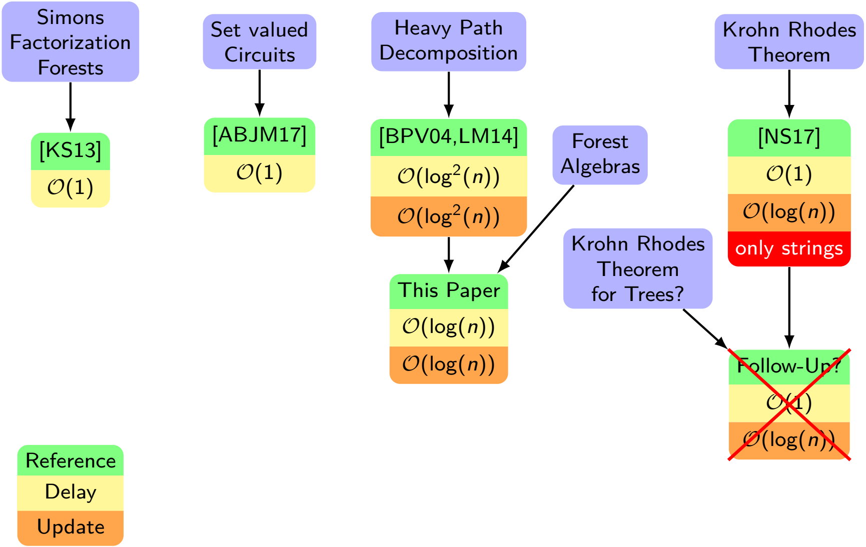
Forest Algebras



Enumeration using Circuits



Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Forest Algebras

This Paper

$\mathcal{O}(\log(n))$

$\mathcal{O}(\log(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

~~Follow-Up?~~

~~$\mathcal{O}(1)$~~

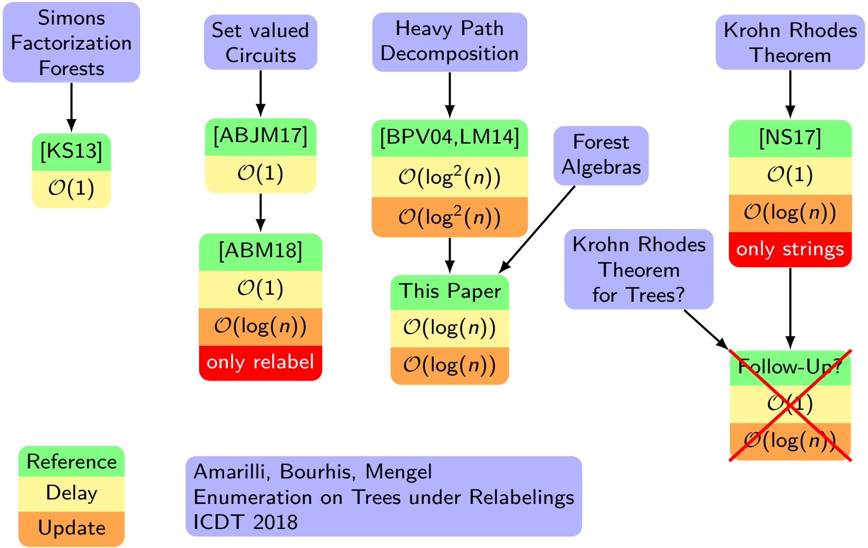
~~$\mathcal{O}(\log(n))$~~

Reference

Delay

Update

Related Work



Simons Factorization Forests

[KS13]
$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]
$\mathcal{O}(1)$

[ABM18]
$\mathcal{O}(1)$
$\mathcal{O}(\log(n))$
only relabel

Heavy Path Decomposition

[BPV04,LM14]
$\mathcal{O}(\log^2(n))$
$\mathcal{O}(\log^2(n))$

Forest Algebras

This Paper
$\mathcal{O}(\log(n))$
$\mathcal{O}(\log(n))$

Krohn Rhodes Theorem

[NS17]
$\mathcal{O}(1)$
$\mathcal{O}(\log(n))$
only strings

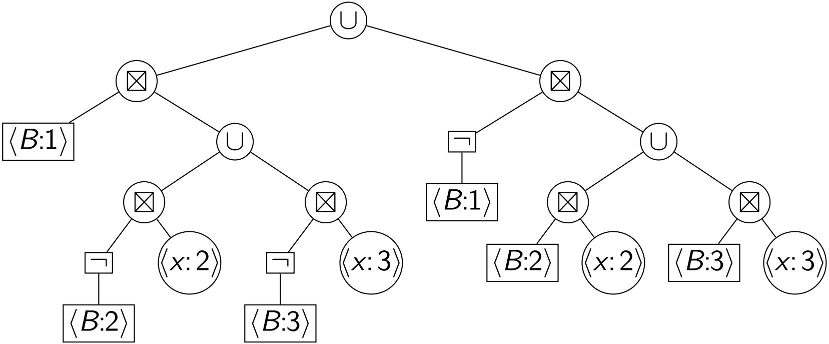
Krohn Rhodes Theorem for Trees?

~~Follow-Up?~~
~~$\mathcal{O}(1)$~~
~~$\mathcal{O}(\log(n))$~~

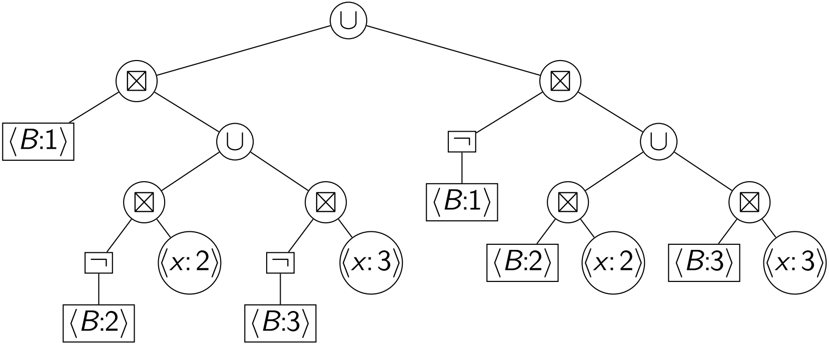
Reference
Delay
Update

Amarilli, Bourhis, Mengel
Enumeration on Trees under Relabelings
ICDT 2018

Enumeration using Circuits [ABM18]

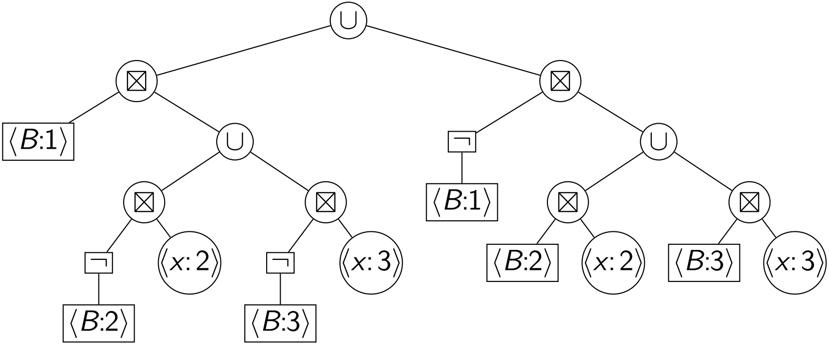


Enumeration using Circuits [ABM18]



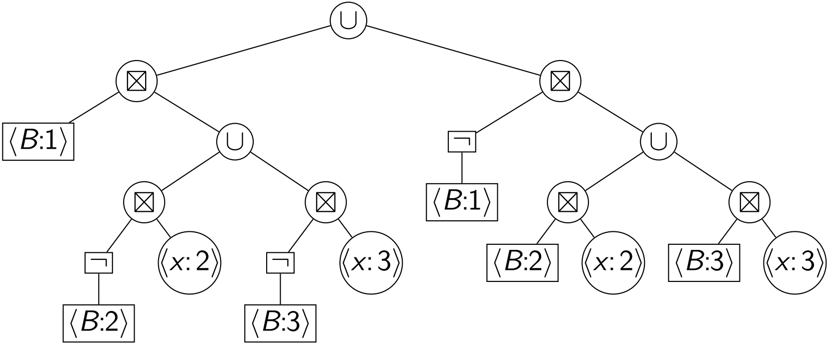
Problem: $\text{depth}(\text{circuit}) \in \Omega(\text{depth}(\text{tree}))$

Enumeration using Circuits [ABM18]



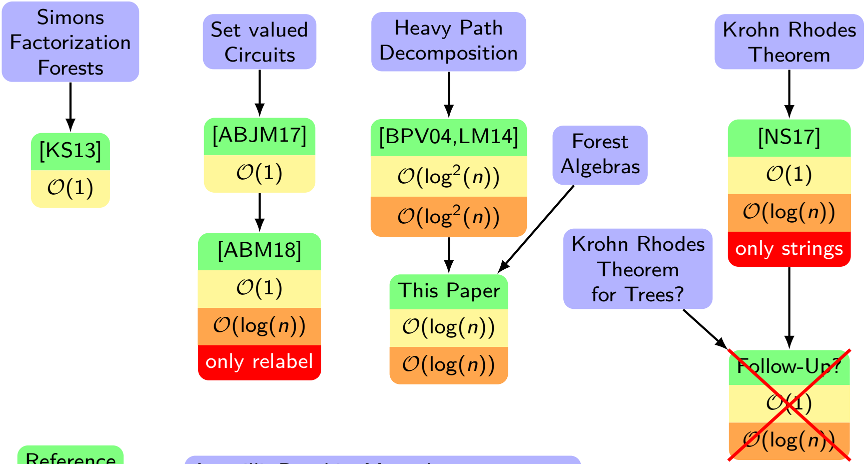
Problem: depth(circuit) $\in \Omega$(depth(tree))
Solution: tree decomposition of input tree

Enumeration using Circuits [ABM18]



Problem:	depth(circuit) $\in \Omega$(depth(tree))
Solution:	tree decomposition of input tree
Better Solution:	use forest algebra formulas

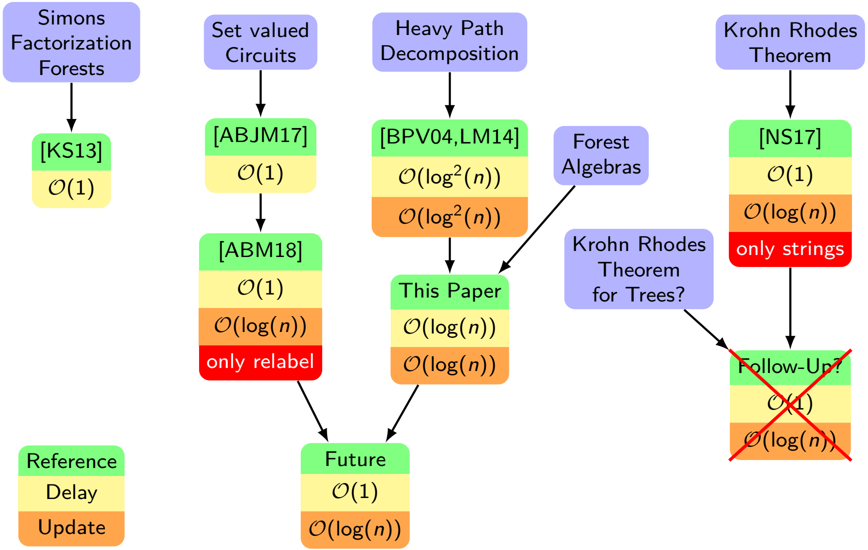
Related Work



Reference
Delay
Update

Amarilli, Bourhis, Mengel
Enumeration on Trees under Relabelings
ICDT 2018

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

[ABM18]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only relabel

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Forest Algebras

This Paper

$\mathcal{O}(\log(n))$

$\mathcal{O}(\log(n))$

Future

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

Follow-Up?

$\mathcal{O}(1)$

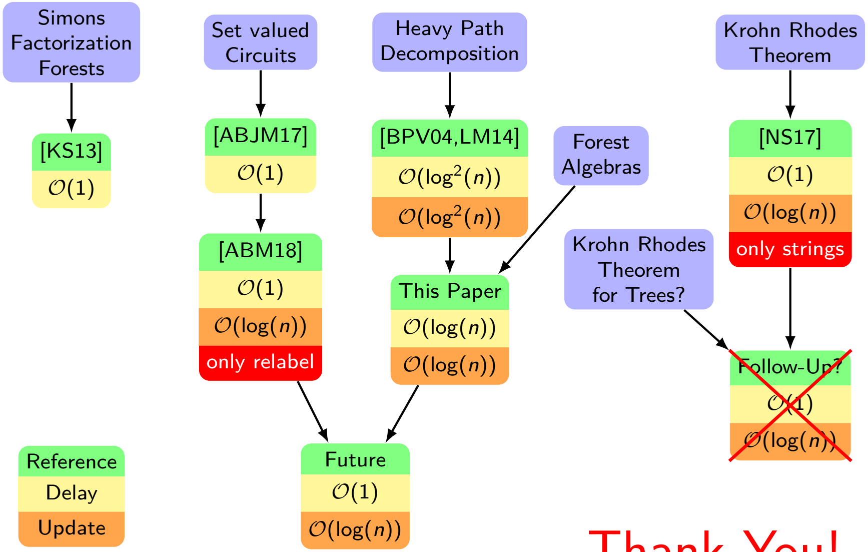
$\mathcal{O}(\log(n))$

Reference

Delay

Update

Related Work



Simons Factorization Forests

[KS13]

$\mathcal{O}(1)$

Set valued Circuits

[ABJM17]

$\mathcal{O}(1)$

[ABM18]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only relabel

Heavy Path Decomposition

[BPV04,LM14]

$\mathcal{O}(\log^2(n))$

$\mathcal{O}(\log^2(n))$

Forest Algebras

This Paper

$\mathcal{O}(\log(n))$

$\mathcal{O}(\log(n))$

Krohn Rhodes Theorem

[NS17]

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

only strings

Krohn Rhodes Theorem for Trees?

Follow-Up?

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

Future

$\mathcal{O}(1)$

$\mathcal{O}(\log(n))$

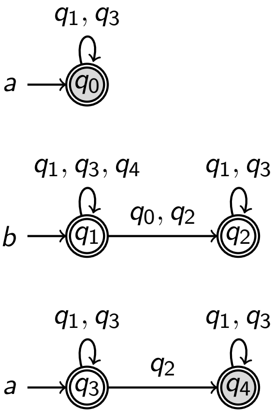
Reference

Delay

Update

Thank You!

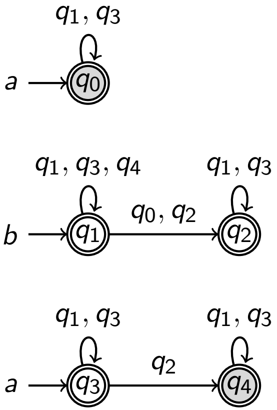
Node Selecting Stepwise Tree Automaton (Example)

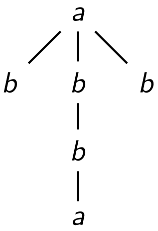


$S = \{(q_4, q_0)\}$

Output all pairs of a-nodes that are connected by a path of b-nodes.

Node Selecting Stepwise Tree Automaton (Example)

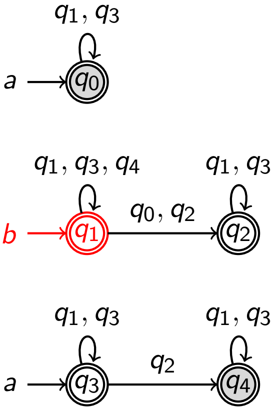


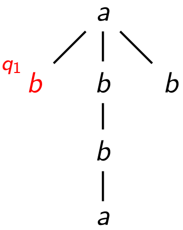


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

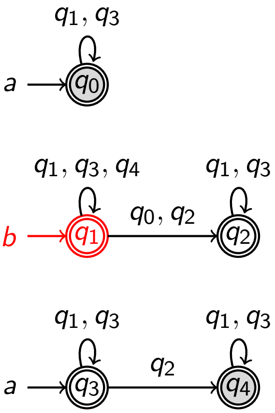


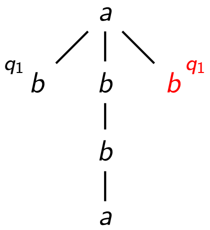


$S = \{(q_4, q_0)\}$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

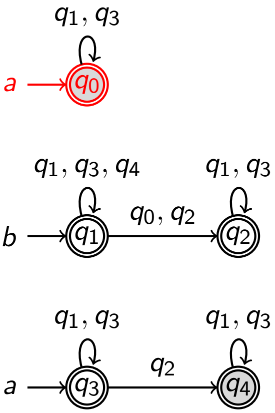


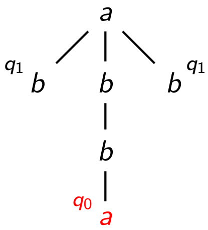


$S = \{(q_4, q_0)\}$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

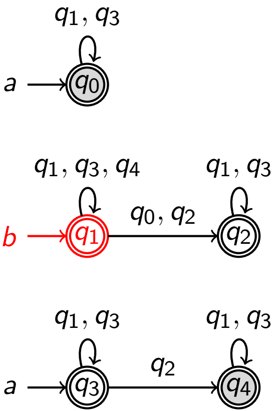


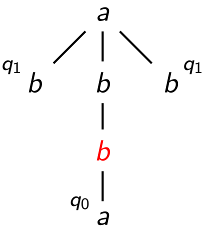


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

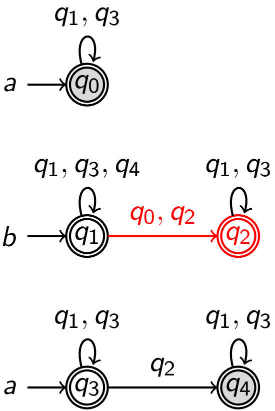


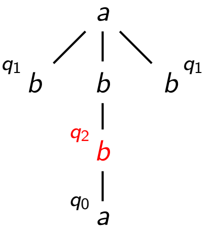


$S = \{(q_4, q_0)\}$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

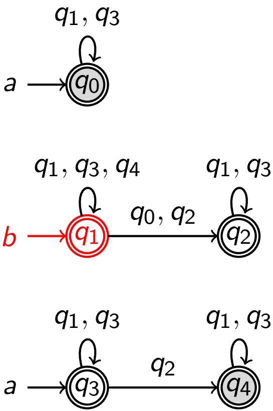


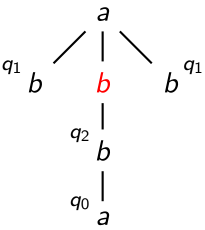


$S = \{(q_4, q_0)\}$

Output all pairs of a-nodes that are connected by a path of b-nodes.

Node Selecting Stepwise Tree Automaton (Example)

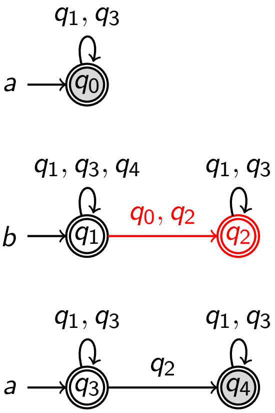


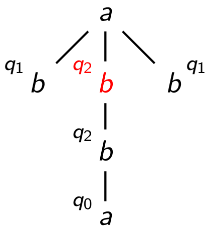


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

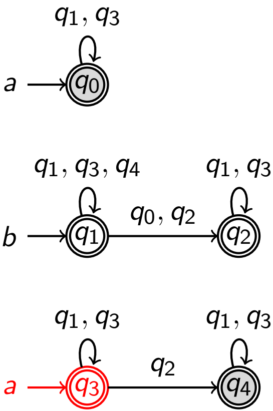


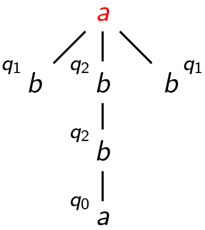


$S = \{(q_4, q_0)\}$

Output all pairs of a-nodes that are connected by a path of b-nodes.

Node Selecting Stepwise Tree Automaton (Example)

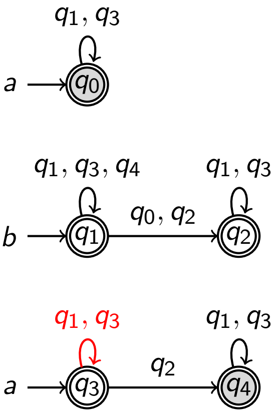


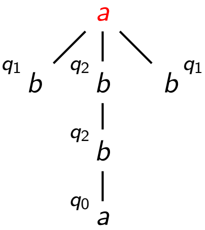


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

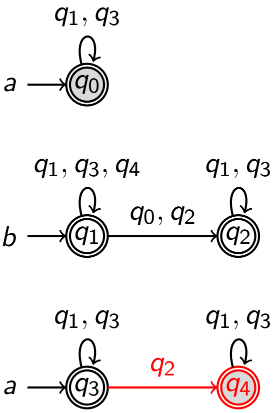


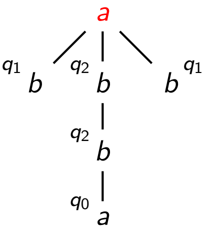


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

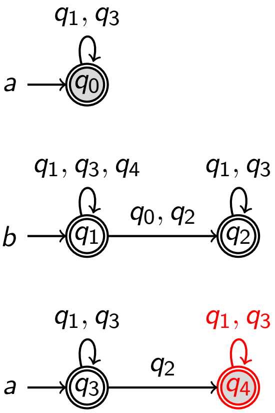


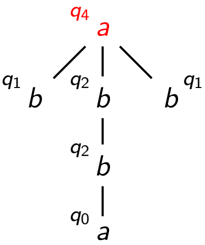


$S = \{(q_4, q_0)\}$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

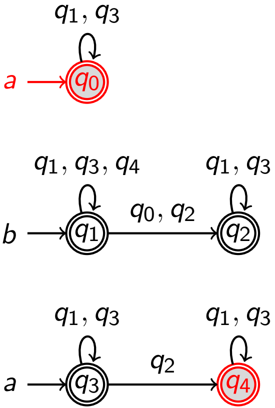


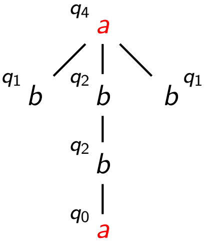


$$S = \{(q_4, q_0)\}$$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Node Selecting Stepwise Tree Automaton (Example)

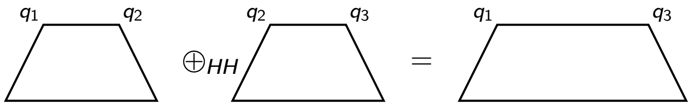




$S = \{(q_4, q_0)\}$

Output all pairs of *a*-nodes that are connected by a path of *b*-nodes.

Transition Algebra



Transition Algebra

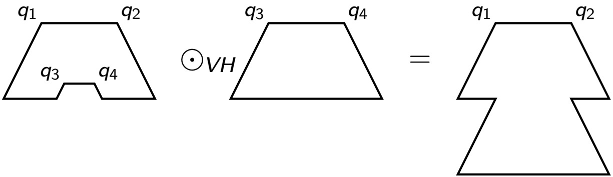
horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$
vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

$$f_1 \oplus_{HH} f_2 = \{ \quad (q_1, q_3) \quad | \quad (q_1, q_2) \in f_1 \text{ and } (q_2, q_3) \in f_2 \quad \}$$

forest $\hat{=}$ string of trees

horizontal monoid $\hat{=}$ transition monoid of a string automaton

Transition Algebra



Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

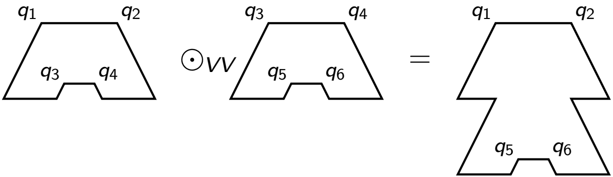
vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

$$c \odot_{VH} f = \{ \ (q_1, q_2) \ \mid \ ((q_1, q_2), (q_3, q_4)) \in c \text{ and } (q_3, q_4) \in f \ \}$$

forest $\hat{=}$ string of trees

horizontal monoid $\hat{=}$ transition monoid of a string automaton

Transition Algebra



Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

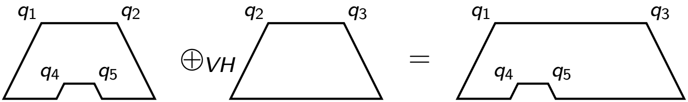
vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

$$c_1 \odot_{VV} c_2 = \{\ ((q_1, q_2), (q_5, q_6))\ \mid\ ((q_1, q_2), (q_3, q_4)) \in c_1 \text{ and } ((q_3, q_4), (q_5, q_6)) \in c_2\ \}$$

forest $\triangleq$ string of trees

horizontal monoid $\triangleq$ transition monoid of a string automaton

Transition Algebra



Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

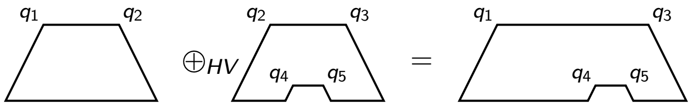
vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

$$f \oplus_{HV} c = \{\ ((q_1,q_3),(q_4,q_5))\ \mid\ (q_1,q_2) \in f \text{ and } ((q_2,q_3),(q_4,q_5)) \in c\ \}$$

forest $\triangleq$ string of trees

horizontal monoid $\triangleq$ transition monoid of a string automaton

Transition Algebra



Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

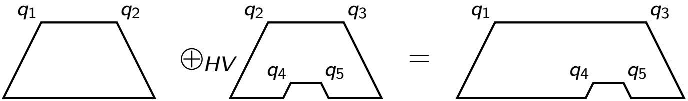
vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

$$c \oplus_{VH} f = \{\ ((q_1,q_3),(q_4,q_5)) \ \mid\ ((q_1,q_2),(q_4,q_5)) \in c \text{ and } (q_2,q_3) \in f\ \}$$

forest $\triangleq$ string of trees

horizontal monoid $\triangleq$ transition monoid of a string automaton

Transition Algebra

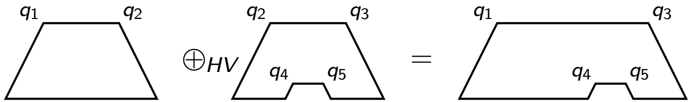


Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

Transition Algebra



Transition Algebra

horizontal monoid: $(2^{Q^2}, \oplus_{HH}, \mathrm{id}_Q)$

vertical monoid: $(2^{(Q^2)^2}, \odot_{VV}, \mathrm{id}_{Q^2})$

Extended Transition Algebra

horizontal monoid: $(2^{Q^2 \times \mathbb{S}(S)}, \oplus_{HH}, \mathrm{id}_Q \times \{\emptyset\})$

vertical monoid: $(2^{(Q^2)^2 \times \mathbb{S}(S)}, \odot_{VV}, \mathrm{id}_{Q^2} \times \{\emptyset\})$