How to Remove the Look-Ahead of Top-Down Tree Transducers

Sebastian Maneth

University of Edinburgh

with Joost Engefriet and Helmut Seidl

DLT 2014 - Екатеринбург

Outline

1. Top-Down Tree Transducers (with and wo Look-Ahead)
2. Difference Trees
3. Canonical Transducers
4. Difference Tuples
5. Main Results
6. Future

1. Tree Transducers

Generalize to ranked (ordered) trees the
finite-state transducer aka “Generalized Sequential Machine” GSM
[Ginsburg 1962]

→ Top-Down Tree Transducer TOP
[Thatcher 1970, “Generalized^2 Sequential Machine Maps”]
[Rounds 1970, “Mappings and Grammars on Trees”]

→ Bottom-Up Tree Transducer BU
[Thatcher 1970]

invented as model for

→ *linguistics* transformational grammars
→ *compiler theory* syntax-directed translations of context-free languages

Recent application: XML

1. Tree Transducers

Generalize to ranked (ordered) trees the
finite-state transducer aka “Generalized Sequential Machine” GSM
[Ginsburg 1962]

→ Top-Down Tree Transducer TOP

Cave TOPs are more complicated than GSMs

→ input subtrees may be processed multiple times (“copying”)
or not at all (“deletion”)

→ size increase is **polynomial** or **exponential** [Aho,Ullman1971]

1. Tree Transducers

→ Top-Down Tree Transducer TOP [Thatcher 1970, Rounds 1970]

→ TOP with **regular Look-Ahead** TLA [Engelfriet 1977]

1. Tree Transducers

→ Top-Down Tree Transducer TOP [Thatcher 1970, Rounds 1970]

→ TOP with **regular Look-Ahead** TLA [Engelfriet 1977]

Advantages of TLA

Composition Closure: linear TLA's
deterministic TLA's

Power: DBU ⊆ DTLA

Functions: TLA* ∩ FUNCT = DTLA [Engelfriet 1978]

→ Natural for XPath filters (often depend on input trees that are deleted)

1. Tree Transducers

→ Top-Down Tree Transducer TOP [Thatcher 1970, Rounds 1970]

→ TOP with regular Look-Ahead TLA [Engelfriet 1977]

Advantages of TLA

Composition Closure: linear TLA's
deterministic TLA's

Power: DBU $\subseteq$ DTLA

Functions: TLA* $\cap$ FUNCT = DTLA [Engelfriet 1978]

→ Natural for XPath filters (often depend on input trees that are deleted)

Advantages of TOP

→ 1-Pass Computable (allows bounded memory stream processing)

→ Canonical Normal Form [Engelfriet,Maneth,Seidl 2008]

→ Myhill-Nerode & Gold-style learning [Lemay,Maneth,Niehren 2010]

Our Problem

Is it decidable for a DTLA (Deterministic Top-down tree transducer with Look-Ahead) whether there exists an equivalent DTOP?

Our Problem

Is it decidable for a DTLA whether there exists an equivalent DTOP?

On strings (DGSMs) easy:

→ transform DGSM-LA into GSM
→ try to determinize (“twinning property” [Choffrut 1977])

Twinning property: (runs do “the same” in a loop)

If $q_0(uwx) \Rightarrow^* a_1 q(wx) \Rightarrow^* a_1 b_1 q(x)$
$q_0(uwx) \Rightarrow^* a_2 p(wx) \Rightarrow^* a_2 b_2 p(x)$
then
$a_1\, b_1\, a_1^{-1} = a_2\, b_2\, a_2^{-1}$

→ only check uw of length $<2|Q|^2$

Note
DTLA $\not\subseteq$ TOP

DTLAs

M = (Q, S, Δ, R, A, **P**, d)

Q: states

S: ranked alphabet of input symbols

Δ: ranked alphabet of output symbols

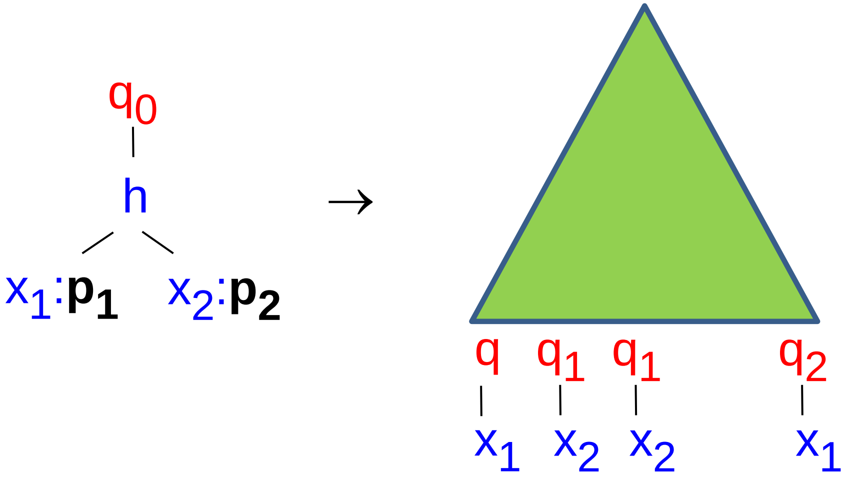
R: rules

A: axioms A(**p**) for every **p** in **P**

P: look-ahead states

d: look-ahead transitions (deterministic and total)

Rules:



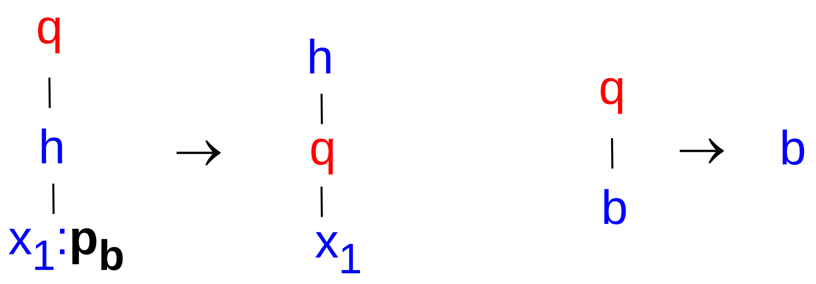
DTLAs

$M_1 = (Q, S, \Delta, R, A, \mathbf{P}, d)$

$d(a) = \mathbf{p_a}$
$d(b) = \mathbf{p_b}$
$d(h(\mathbf{p_a}/\mathbf{p_b})) = \mathbf{p_a}/\mathbf{p_b}$

$A(\mathbf{p_a}) = a$
$A(\mathbf{p_b}) = q(x_0)$

$M_1(h^n(a)) = a$
$M_1(h^n(b)) = h^n(b)$



DTLAs

$M_1 = (Q, S, \Delta, R, A, \mathbf{P}, d)$

$d(a) = \mathbf{p_a}$
$d(b) = \mathbf{p_b}$
$d(h(\mathbf{p_a}/\mathbf{p_b})) = \mathbf{p_a}/\mathbf{p_b}$

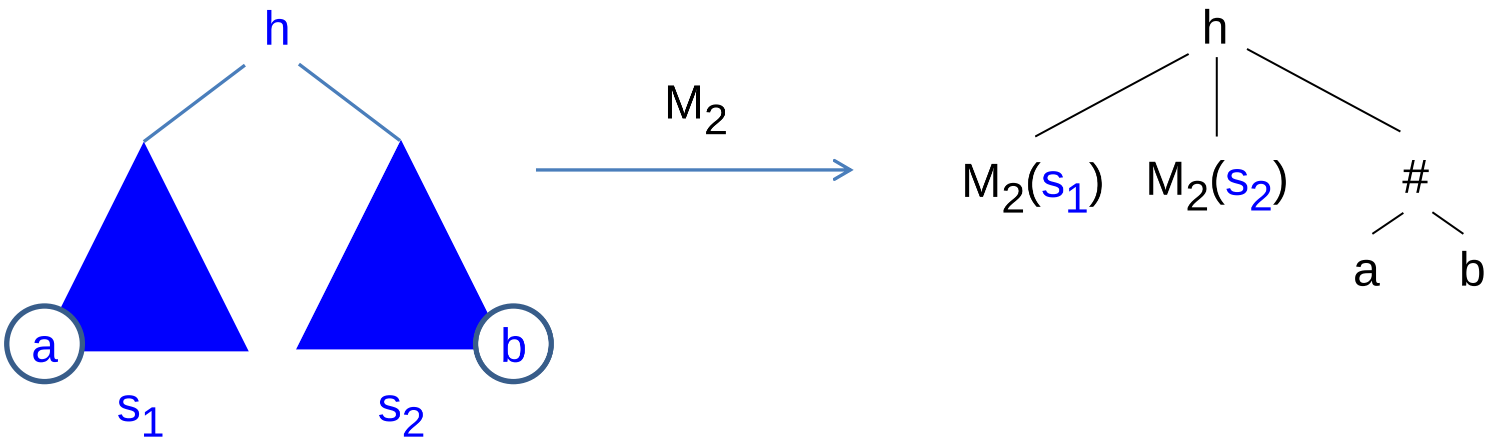
$A(\mathbf{p_a}) = a$
$A(\mathbf{p_b}) = q(x_0)$

$M_1(h^n(a)) = a$
$M_1(h^n(b)) = h^n(b)$

$q(h(x_1{:}\mathbf{p_b})) \rightarrow h(q(x_1))$ $q(b) \rightarrow b$

➔ Can M_1 be realized without look-ahead (i.e., by a DTOP)?

DTLAs



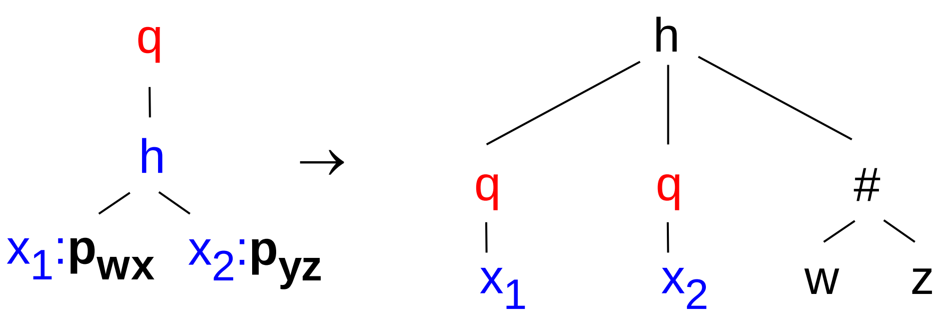
$\mathbf{P} = \{\mathbf{p}_{\mathbf{xy}} \mid x,y \in \{a, b\}\}$

$d(a) = \mathbf{p}_{\mathbf{aa}}$ $d(b) = \mathbf{p}_{\mathbf{bb}}$

$d(h(\mathbf{p}_{\mathbf{wx}}, \mathbf{p}_{\mathbf{yz}})) = \mathbf{p}_{\mathbf{wz}}$ for all w,x,y,z $\in$ {a, b}

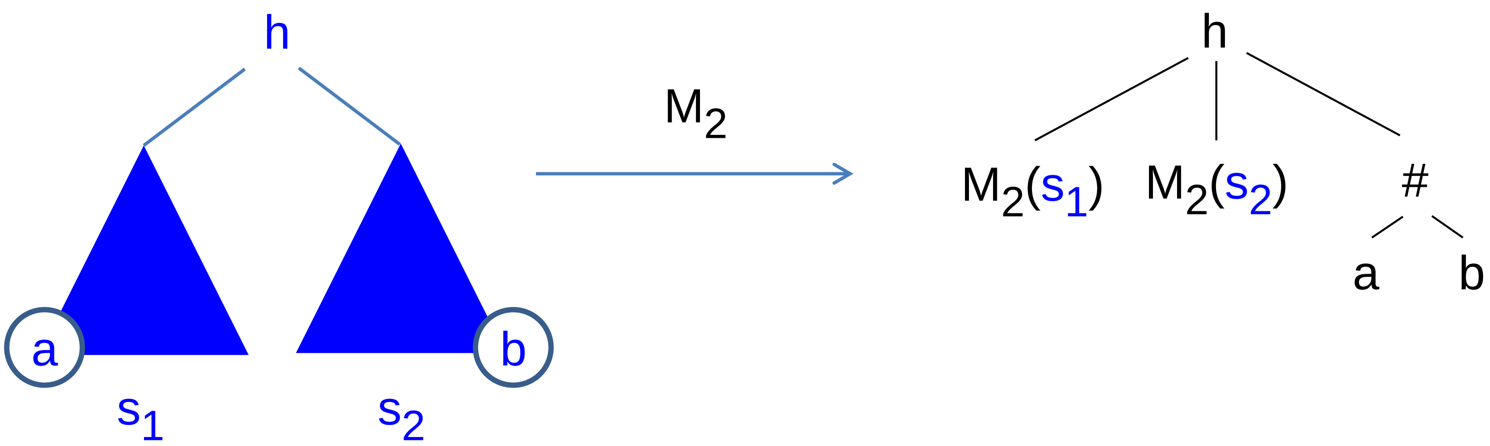
$A(\mathbf{p}_{\mathbf{yz}}) = q(x_0)$

$q(a/b) \rightarrow a/b$



DTLAs

➔ Can M_2 be realized wo look-ahead (i.e., by a DTOP)?



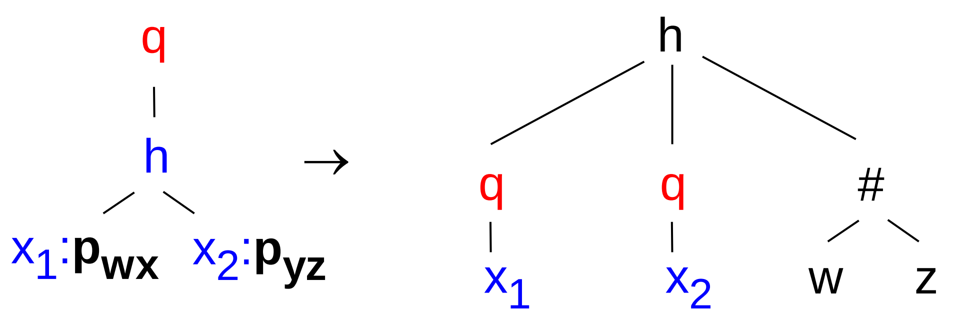
$\mathbf{P} = \{\mathbf{p}_{\mathbf{xy}} \mid x,y \in \{a, b\}\}$

$d(a) = \mathbf{p}_{\mathbf{aa}}$ $\quad d(b) = \mathbf{p}_{\mathbf{bb}}$

$d(h(\mathbf{p}_{\mathbf{wx}}, \mathbf{p}_{\mathbf{yz}})) = \mathbf{p}_{\mathbf{wz}}$ $\quad$ for all $w,x,y,z \in \{a, b\}$

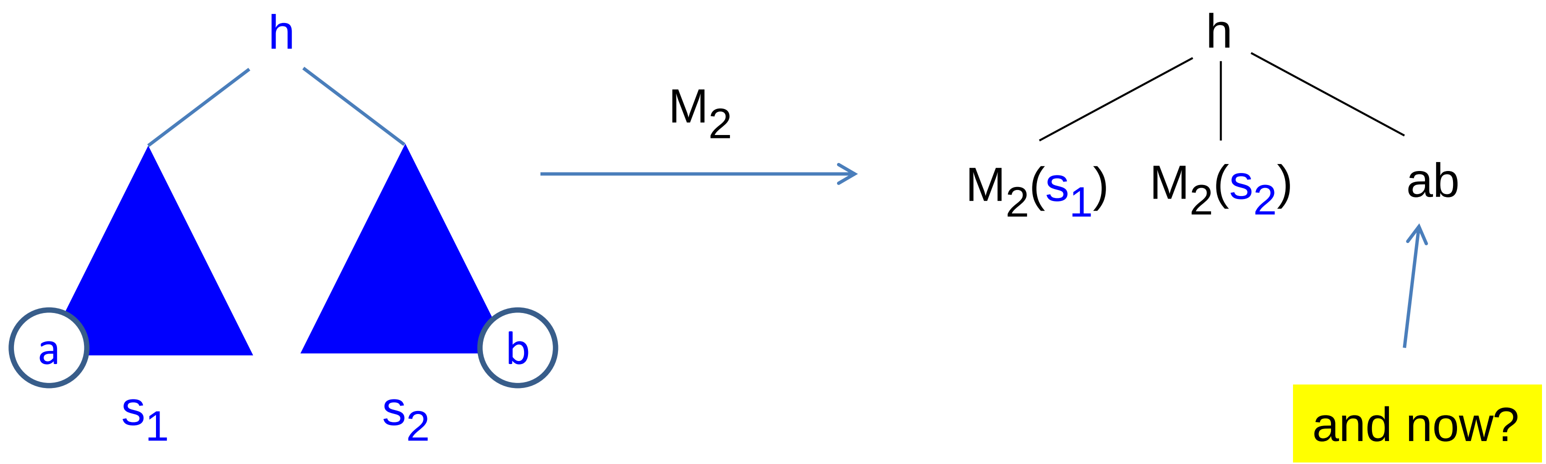
$A(\mathbf{p}_{\mathbf{yz}}) = q(x_0)$

$q(a/b) \rightarrow a/b$



DTLAs

➔ Can M_2 be realized wo look-ahead (i.e., by a DTOP)??



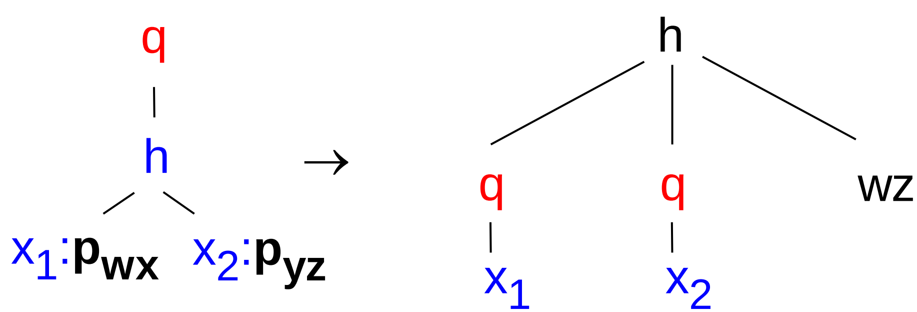
$\mathbf{P} = \{\mathbf{p_{xy}} \mid x,y \in \{a, b\}\}$

$d(a) = \mathbf{p_{aa}}$ $d(b) = \mathbf{p_{bb}}$

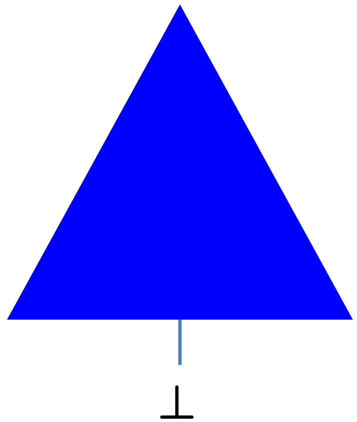
$d(h(\mathbf{p_{wx}}, \mathbf{p_{yz}})) = \mathbf{p_{wz}}$ for all $w,x,y,z \in \{a, b\}$

$A(\mathbf{p_{yz}}) = q(x_0)$

$q(a/b) \rightarrow a/b$



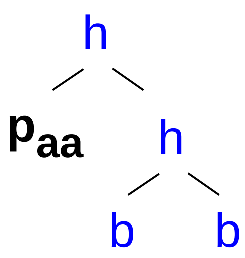
DTLAs

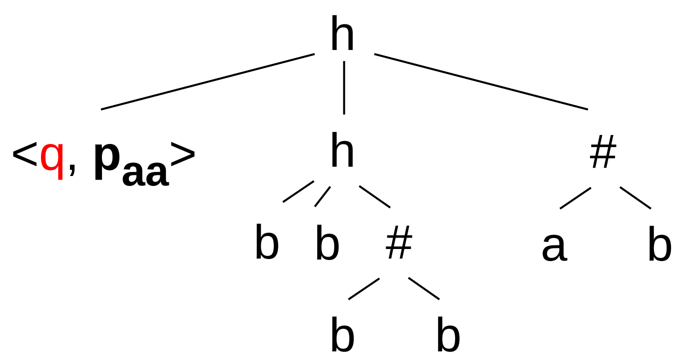


Context C = input tree with exactly one ⊥-leaf

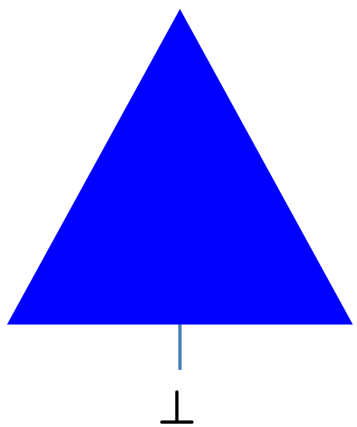
C[**p**] = C[⊥ ← **p**]

M(C[**p**]) = output tree with leaves <q, **p**> where q reaches the **p**-leaf

$M_2($ 

$) =$ 

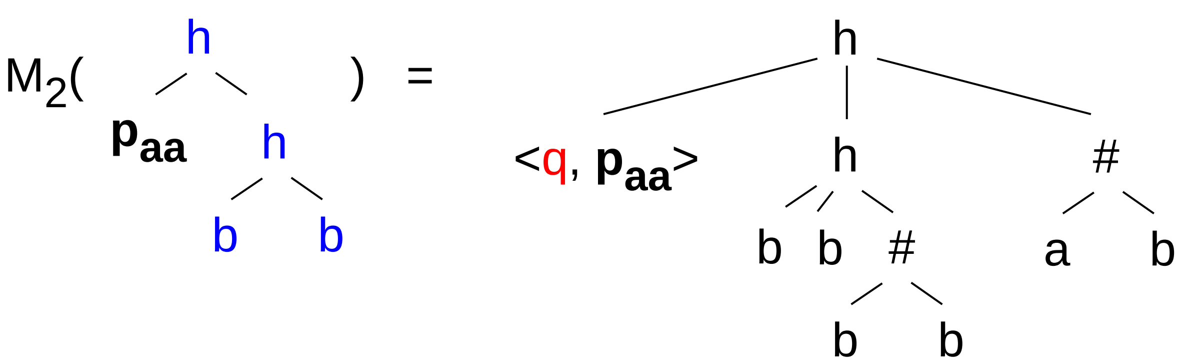
Replacement Lemma



Context C = input tree with exactly one ⊥-leaf

$C[\mathbf{p}] = C[\bot \leftarrow \mathbf{p}]$

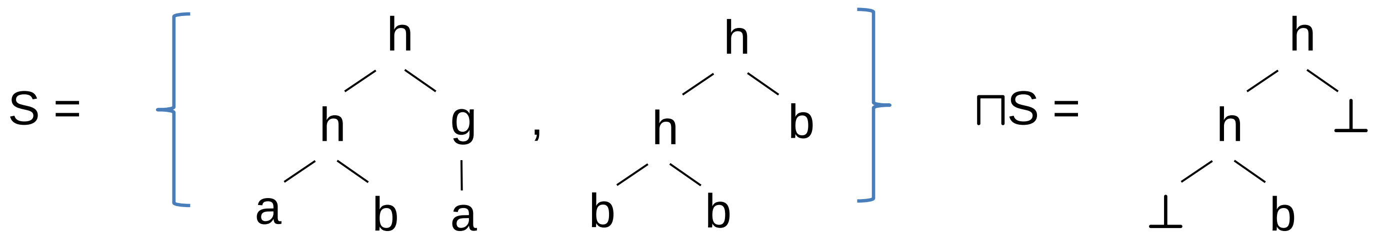
Lemma $\forall s$ with $d(s)=\mathbf{p}$, $M(C[s]) = M(C[\mathbf{p}])[\langle q, \mathbf{p}\rangle \leftarrow M_q(s) \mid q \in Q]$.



2. Difference Trees

How can we characterize output that a DTLA produces
due to its look-ahead information?

S = set of trees
⊓S = largest common prefix of the trees in S

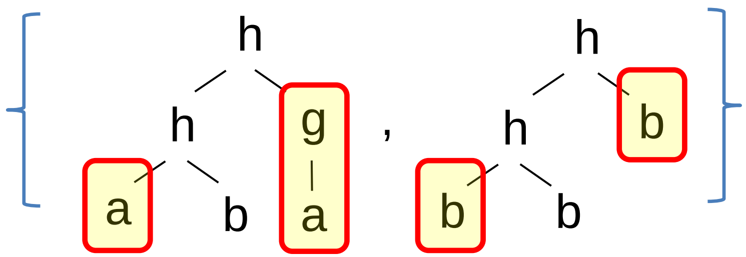


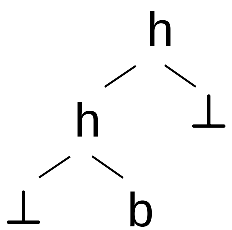
Difference tree = subtree of s at v for s ∈ S
and v a ⊥-node in ⊓S

2. Difference Trees

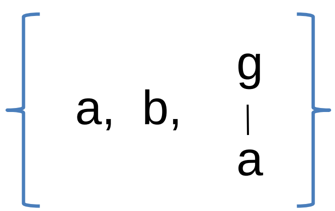
How can we characterize output that a DTLA produces
due to its look-ahead information?

S = set of trees
⊓S = largest common prefix of the trees in S

S = 

⊓S = 

Difference tree = subtree of s at v for s ∈ S
and v a ⊥-node in ⊓S

diff(S) = 

2. Difference Trees

How can we characterize output that a DTLA produce
due to its look-ahead information?

S = set of trees
$\sqcap$S = largest common prefix of the trees in S
Difference tree = subtree of s at v for s $\in$ S
and v a $\perp$-node in $\sqcap$S

p, **p'**: distinct look-ahead states

M(C[**p**]) $\sqcap$ M(C[**p'**]) = output **not** depending on look-ahead **p**/**p'**

diff({ M(C[**p**]), M(C[**p'**]) }) = trees that M outputs due to **p**/**p'**-information

diff(M) := $\bigcup$ { diff({ M(C[**p**]), M(C[**p'**]) }) | C$\in$Con, **p**,**p'** $\in$ P }

Measure of how much M makes use of its look-ahead

2. Difference Trees

$M_1(h^n(\mathbf{p_a})) = a$
$M_1(h^n(\mathbf{p_b})) = h^n(\langle q, \mathbf{p_b} \rangle)$

$\sqcap = \bot$

$\mathrm{diff}(M_1) = \{a\} \cup \{ h^n(\langle q, \mathbf{p_b} \rangle) \mid n \geq 0 \}$

---

p, **p’**: distinct look-ahead states

$M(C[\mathbf{p}]) \sqcap M(C[\mathbf{p'}])$ = output **not** depending on look-ahead **p**/**p’**

$\mathrm{diff}(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \})$ = trees that M outputs due to **p**/**p’**-information

$\mathrm{diff}(M) := \bigcup \{ \mathrm{diff}(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \}) \mid C \in Con, \mathbf{p},\mathbf{p'} \in P \}$

Measure of how much M makes use of its look-ahead

2. Difference Trees

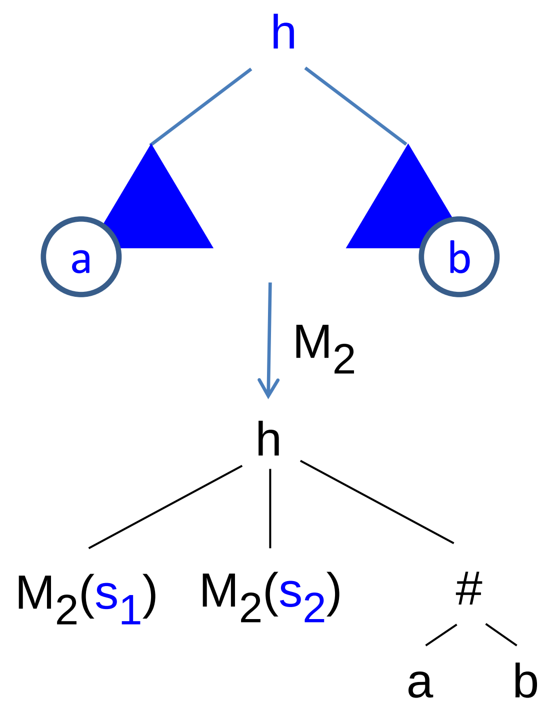
$M_1(h^n(\mathbf{p_a})) = a$

$M_1(h^n(\mathbf{p_b})) = h^n(<q, \mathbf{p_b}>)$

$\sqcap = \bot$

$diff(M_1) = \{a\} \cup \{ h^n(<q, \mathbf{p_b}>) \mid n \geq 0 \}$

$diff(M_2) = ?$



---

p, **p'**: distinct look-ahead states

$M(C[\mathbf{p}]) \sqcap M(C[\mathbf{p'}])$ = output **not** depending on look-ahead **p**/**p'**

$diff(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \})$ = trees that M outputs due to **p**/**p'**-information

$diff(M) := \bigcup \{ diff(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \}) \mid C \in Con, \mathbf{p},\mathbf{p'} \in P \}$

Measure of how much M makes use of its look-ahead

2. Difference Trees

$M_1(h^n(\mathbf{p_a})) = a$

$M_1(h^n(\mathbf{p_b})) = h^n(\langle q, \mathbf{p_b} \rangle)$

$\sqcap = \bot$

$\mathrm{diff}(M_1) = \{a\} \cup \{ h^n(\langle q, \mathbf{p_b} \rangle) \mid n \geq 0 \}$

$\mathrm{diff}(M_2) = \{a, b\} \cup \{ \langle q, \mathbf{p} \rangle \mid \mathbf{p} \in \mathbf{P} \}$

---

p, **p'**: distinct look-ahead states

$M(C[\mathbf{p}]) \sqcap M(C[\mathbf{p'}])$ = output **not** depending on look-ahead **p**/**p'**

$\mathrm{diff}(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \})$ = trees that M outputs due to **p**/**p'**-information

$\mathrm{diff}(M) := \bigcup \{ \mathrm{diff}(\{ M(C[\mathbf{p}]), M(C[\mathbf{p'}]) \}) \mid C \in \mathrm{Con}, \mathbf{p},\mathbf{p'} \in P \}$

Measure of how much M makes use of its look-ahead

2. Difference Trees

$M_1(h^n(\mathbf{p_a})) = a$
$M_1(h^n(\mathbf{p_b})) = h^n(<q, \mathbf{p_b}>)$ → $\sqcap = \bot$

$diff(M_1) = \{a\} \cup \{ h^n(<q, \mathbf{p_b}>) \mid n \geq 0 \}$ ← infinite

$diff(M_2) = \{a, b\} \cup \{ <q, \mathbf{p}> \mid \mathbf{p} \in \mathbf{P} \}$ ← finite

---

Define maxdiff(M) = maximal height of a tree in diff(M)
if is diff(M) finite

difference bound of M = number m such that
height(s) $\leq$ m for every s$\in$ diff(M)

3. Canonical Transducers

[EMS 2008] For every DTOP M an equivalent canonical DTOP can(M) can be constructed.

canonical = (1) each output node is produced as **early** as possible
(2) different states are inequivalent

Two DTOPs are equivalent iff their canonical DTOPs are equal (up to renaming of states).

Theorem 1 For every DTLA M an equivalent canonical DTLA can(M) with same look-ahead automaton as M can be constructed s.t.

$\text{maxdiff}(M) - 8^{|M|^3} \leq \text{maxdiff}(\text{can}(M)) \leq \text{maxdiff}(M) + 8^{|M|^3}$.

3. Canonical Transducers

M: canonical DTLA
N: canonical DTOP
such that M is equivalent to N

→ How are they related??

3. Canonical Transducers

M: canonical DTLA
N: canonical DTOP
such that M is equivalent to N

→ How are they related??

M is always ahead of N, i.e.,

→ N(C) is a prefix of M(C[**p**]) $\forall$C and **p**$\in$**P**

3. Canonical Transducers

M: canonical DTLA
N: canonical DTOP
such that M is equivalent to N

→ How are they related??

M is always ahead of N, i.e.,

→ N(C) is a prefix of M(C[**p**]) $\forall$C and **p**$\in$**P**

"aheadness mapping from N to M" = replaces <q, **p**> by tree in $T_{\Delta}(Q_M)$

Lemma
There is a *unique* aheadness mapping φ from N to M.

4. Difference Tuples

s/v = subtree of s at node v

diftup$(s_1, \ldots, s_n) = \{(s_1/v, \ldots, s_n/v) \mid v \text{ is } \perp\text{-node in } \sqcap\{s_1, \ldots, s_n\}\}$

$\mathbf{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$ (in that order)

diftup$(M) = \bigcup \{ \text{diftup}(M(C([\mathbf{p_1}]), \ldots, M(C[\mathbf{p_n}])) \mid C \in \text{Con} \}$

M,N: canonical and equivalent

φ: aheadness mapping from N to M

Let $\psi(q) = (\varphi(q, \mathbf{p_1}), \ldots, \varphi(q, \mathbf{p_n}))$ for $q \in Q_N$.

Lemma ψ is a bijection between Q_N and diftup(M).

4. Difference Tuples

s/v = subtree of s at node v

diftup($s_1, \ldots, s_n$) = $\{(s_1/v, \ldots, s_n/v) \mid v \text{ is } \bot\text{-node in } \sqcap\{s_1, \ldots, s_n\}\}$

$\mathbf{P} = \{\mathbf{p_1}, \ldots, \mathbf{p_n}\}$ (in that order)

diftup(M) = $\bigcup \{ \text{diftup}(M(C([\mathbf{p_1}]), \ldots, M(C[\mathbf{p_n}])) \mid C \in \text{Con} \}$

M,N: canonical and equivalent

φ: aheadness mapping from N to M

Let $\psi(q) = (\varphi(q, \mathbf{p_1}), \ldots, \varphi(q, \mathbf{p_n}))$ for $q \in Q_N$.

Lemma ψ is a bijection between Q_N and diftup(M).

Corollary diff(M) is finite.

4. Difference Tuples

Corollary diff(M) is finite.

→ so, we “only” need to check finiteness of diff(M) (for M canonical) plus some other properties

For instance:

M(f(a,a)) → 1
M(any other tree) → 0

diff(M) is finite, but translation is not in DTOP

5. Main Result

Theorem It is decidable for given DLTA M and difference bound m for M, whether there exists an equivalent DTOP N, and if so, N can be constructed.

Proof

→ make M canonical (adds $8^{|M|^3}$ to m)
→ successively construct aheadness mapping and corresponding rules
→ finally check if resulting transducer is equivalent to M

Main Crux

How to compute a difference bound for a given DTLA?!

DTLA is

→ linear — if every x_i occurs at most once in every rhs of a rule

→ nonerasing — if no rhs is of the form $q(x_i)$

Theorem It is decidable for a linear nonerasing DTLA M whether there exists an equivalent DTOP N (then N can be constructed)

Proof By (difficult) pumping arguments show that $37*|M|^5$ is a difference bound.

Main Crux

How to compute a difference bound for a given DTLA?!

DTLA is
→ linear if every x_i occurs at most once in every rhs of a rule
→ nonerasing if no rhs is of the form $q(x_i)$

Theorem It is decidable for a linear nonerasing DTLA M whether there exists an equivalent DTOP N (then N can be constructed)

Proof By (difficult) pumping arguments show that $37*|M|^5$ is a difference bound.

→ can be extended to "ultralinear" and "bounded erasing".
See [arXiv]-version (57 pages)

6. Future

Extend to

→ partial ultralinear and bounded erasing DTLAs
(our proofs are for total transducers)

→ all DTLAs ☺

→ Algorithm to construct equivalent DTLA with minimal |**P**|

(would enable Myhill-Nerode and Gold-style learning)

5. Future

Extend to

→ partial ultralinear and bounded erasing DTLAs
(our proofs are for total transducers)

→ all DTLAs ☺

→ Algorithm to construct equivalent DTLA with minimal |**P**|

(would enable Myhill-Nerode and Gold-style learning)

>- THE END -<

Thanks for your Attention!