

# *Deciding Equivalence of Top-Down XML Transformations in Polynomial Time*

**Sebastian Maneth**

National ICT Australia Ltd. & UNSW, Sydney

Joint work w. [Helmut Seidl](#) (TU Munich)

January 20th, 2007 PLAN-X, Nice

## Prologue *Tree Transducers*

= (finitely described) *models for relations* on (ordered) *trees*

---

E.g. → **finite-state** (generalize FTA to input + output)

**Example** top-down tree transducer (TOP) [Rounds/Thatcher, 70's]

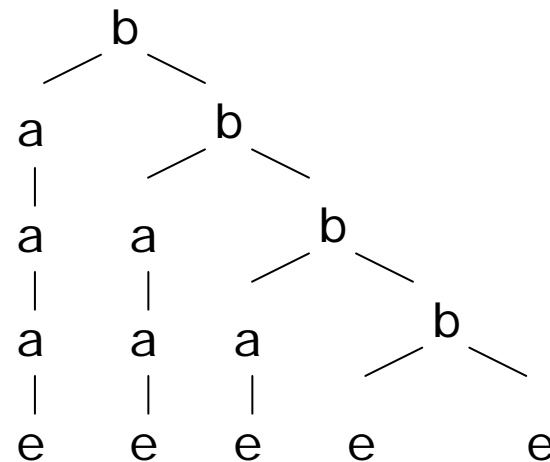
$q_0(a(x)) \rightarrow b(q(x), q_0(x))$

$q_0(e) \rightarrow e$

$q(a(x)) \rightarrow a(q(x))$

$q(e) \rightarrow e$

a  
|  
a  
|  
a  
|  
a  
|  
e



## Prologue *Tree Transducers*

= (finitely described) *models for relations* on (ordered) *trees*

---

E.g. → **finite-state** (generalize FTA to input + output)

**Example** top-down tree transducer (TOP) [Rounds/Thatcher, 70's]

$q_0(a(x)) \rightarrow b(q(x), q_0(x))$

$q_0(e) \rightarrow e$

$q(a(x)) \rightarrow a(q(x))$

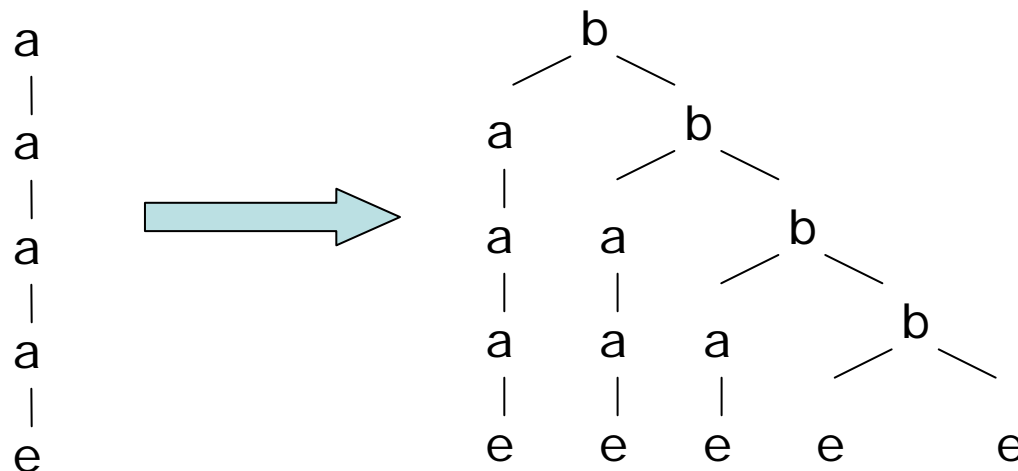
$q(e) \rightarrow e$

$p_0(a(x)) \rightarrow p(x)$

$p_0(e) \rightarrow e$

$p(a(x)) \rightarrow b(a(q(x)), p(x))$

$p(e) \rightarrow b(e, e)$



## Prologue *Tree Transducers*

= (finitely described) *models for relations* on (ordered) *trees*

---

E.g. → **finite-state** (generalize FTA to input + output)

**Example** top-down tree transducer (TOP) [Rounds/Thatcher, 70's]

$$q_0(a(x)) \rightarrow b(q(x), q_0(x))$$

$$q_0(e) \rightarrow e$$

$$q(a(x)) \rightarrow a(q(x))$$

$$q(e) \rightarrow e$$

$$p_0(a(x)) \rightarrow p(x)$$

$$p_0(e) \rightarrow e$$

$$p(a(x)) \rightarrow b(a(q(x)), p(x))$$

$$p(e) \rightarrow b(e, e)$$

**M1**

*is equivalent to*

**M2**

---

Transducers T1, T2 are **equivalent** iff  $\forall$  input  $s$ :  $T1(s) = T2(s)$ .

## Theorem [Esik80]

For two deterministic TOPs it is **decidable** if they are **equivalent**.

Proof idea

Build tree automaton that keeps track of “difference trees”.

CAVE Those trees can be very large! Complexity?!

---

Theorem [Esik80]

For two deterministic TOPs it is **decidable** if they are **equivalent**.

Proof idea

Build tree automaton that keeps track of “difference trees”.  
CAVE Those trees can be very large! Complexity?!

---

## Our Contribution

Canonical normal form for TOPs: “*uniform and earliest*”

### Theorem

$\text{Uniform\&Earliest}(T_1)$  is **isomorphic** to  $\text{Uniform\&Earliest}(T_2)$   
if and only if  $M_1$  is **equivalent** to  $M_2$ .

If  $M$  is total, then  $\text{Uniform\&Earliest}(M)$  obtained in PTIME.

# Outline

- Equivalence Problems & Tree Transducers
- Uniform & Earliest
- Regular Look-Ahead
- Applications
  - Inclusion of XML Queries

## Equivalence Problems of String / Tree Transducers

- *nondeterministic (one-way) finite state transducers*      **undecidable**  
[Griffiths68]  
(→ reduction from PCP, use complement and union)
  - *deterministic (one-way) finite state transducers*      **decidable** [Gurari82]  
(→ use Parikh property)
- 
- *deterministic top-down tree transducers*      **decidable** [Esik80]
  - *nonnested, seperated attributed/marco tree transducers*      **decidable**  
[Courcelle/Franchi-Zannetacci82]  
→ seperated = can be evaluated in two phases,  
(1) only inherited, over  $\Delta_{inh}$       (2) only synthsized, over  $\Delta_{syn}$
  - *MSO definable tree transducers*      **decidable**  
(→ use Parikh property) [Engelfriet/Maneth05]

Tree Transducers

MTT<sup>k+1</sup>

k-pebble Tr Tr

Milo/Suciu/Vianu

Macro Tree Transducers

Courcelle; Engelfriet/Vogler

2Exp size increase  
(linear, if tree-to-SGRAPH)

Attributed Tree Transducers

Knuth; Fülöp

Exp size increase  
(linear, if tree-to-DAG)

seperated

↑ *non-regular  
output paths  
"with context"*

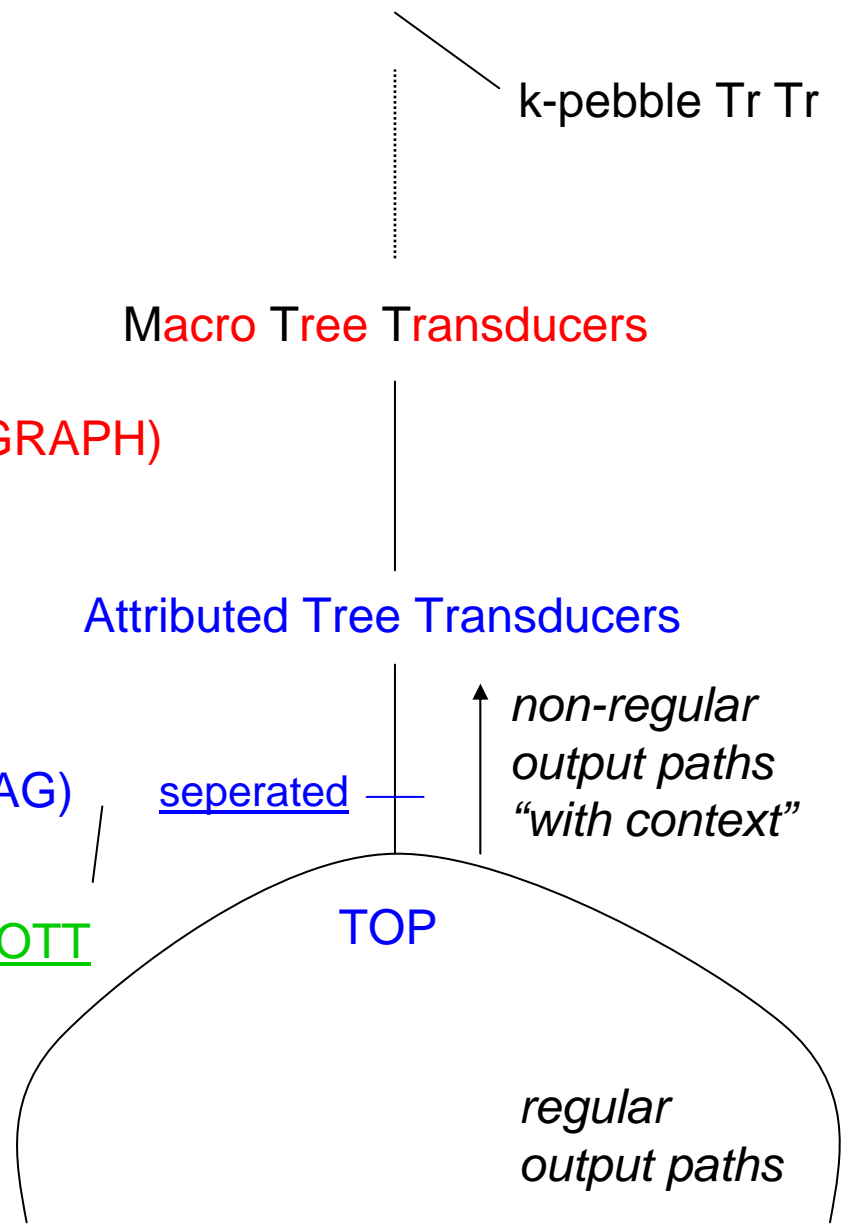
MSOTT

TOP

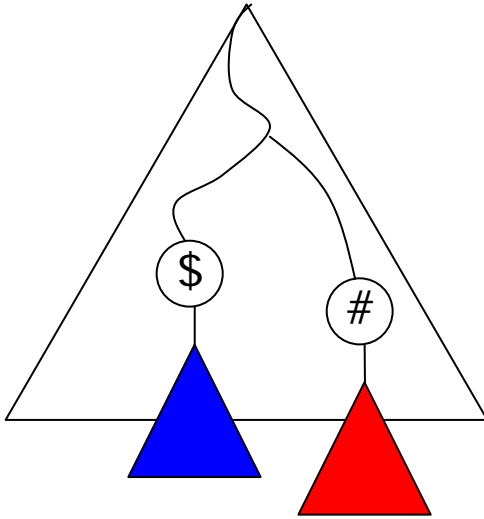
Rounds; Thatcher

Lin size increas

*regular  
output paths*



## *Tree Transducers with Context*

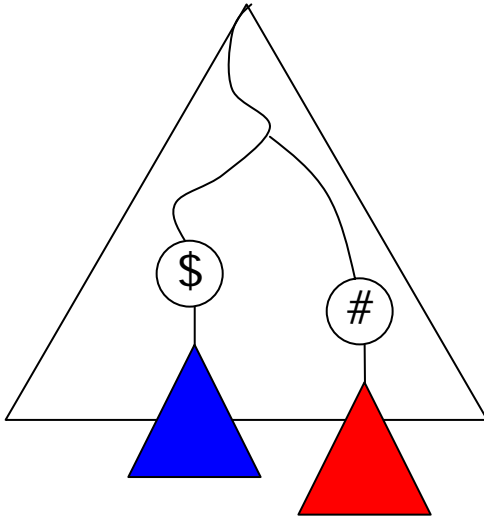


Input: exactly one \$-node and  
one #-node (independent)

Output: remove blue;  
replace red by blue;

→ can NOT be done by a TOP!

## Tree Transducers with Context



Input: exactly one \$-node and one #-node (independent)

Output: remove blue;  
replace red by blue;

→ can NOT be done by a TOP!

Macro Tree Transducer (MTT) = TOP + **Context Parameters**

$q_0(a(x_1, x_2)) \rightarrow$

```

      a
     / \
  q, x1 q, x1
   |   |   |
  r, x1 r, x1
   |   |   |
  r, x2 r, x2
   |   |   |
   e   e   e
  
```

$q(a(x_1, x_2), y) \rightarrow$   
 $a(q(x_1, y), q(x_2, y))$   
 $q(\#(x), y) \rightarrow \#(y)$   
 $q(\$(x), y) \rightarrow \$$

r: find \$ and copy subtree..

## Macro Tree Transducer (MTT) [Engelfriet/Vogler1985]

→ FPs on trees, with **parameters**, pattern matching as ONLY operation

```
function q(s: itree, y: otree): otree
{
  case s=a(x) → return q(x, q(x, y))
  case s=e    → return b(y, y)      }
```

→ can simulate attribute grammars (seen as tree translations)

→ always terminate (no circularities, strictly descent input tree)

→ even compositions (!) can be computed in [Maneth03]  
time  $O(\text{size}(\text{input tree}) + \text{size}(\text{output tree}))$   
“static garbage collection” ☺

→ Linear size incr. *decidable* = MSO transducers (→ *decidable equivalence*)

## Macro Tree Transducer (MTT) [Engelfriet/Vogler1985]

→ FPs on trees, with **parameters**, pattern matching as ONLY operation

```
function q(s: itree, y: otree): otree
{
  case s=a(x) → return q(x, q(x, y))
  case s=e    → return b(y, y)      }

```

→ can simulate attribute grammars (seen as tree translations)

→ always terminate (no circularities, strictly descent input tree)

→ even compositions (!) can be computed in [Maneth02]  
time  $O(\text{size}(\text{input tree}) + \text{size}(\text{output tree}))$   
“static garbage collection” ☺

→ Linear size incr. *decidable* = MSO transducers (→ *decidable equivalence*)

---

But, unfortunately

**OPEN** whether **deterministic MTTs** have **decidable equivalence**.

# Macro Tree Transducer (MTT) [Engelfriet/Vogler1985]

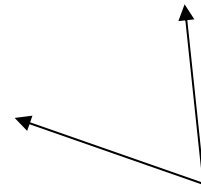
## OPEN

Even for *monadic output* MTTs (= top-down tree-to-string transducers )

$$\begin{aligned} q(a(x), y) &\rightarrow a(q(x, q(x, a(y)))) \\ q(e, y) &\rightarrow y \end{aligned}$$

$$\begin{aligned} q(a(x)) &\rightarrow a \ q(x) \ q(x) \ a \\ q(e) &\rightarrow \lambda \end{aligned}$$

$$\begin{aligned} p(a(x)) &\rightarrow p(x) \ a \ a \ p(x) \\ p(e) &\rightarrow \lambda \end{aligned}$$



Equivalent?!  
OPEN for 30 years... ☹

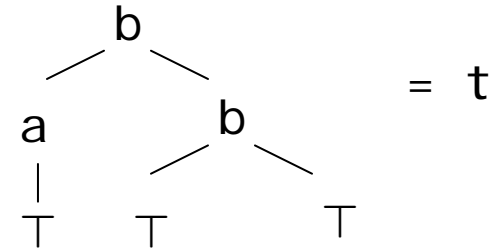
---

But, unfortunately

**OPEN** whether **deterministic MTTs** have **decidable equivalence**.

## Uniform and Earliest TOPs

→ patterns (trees w. holes)  $P_\Sigma = T_{\Sigma \cup \{\top\}}$



→ substitution  $t[t_1, t_2, t_3] = b(a(t_1), b(t_2, t_3))$

→ pattern order  $\perp \sqsubseteq p$  for all  $p \in P_\Sigma$   
 $p \sqsubseteq q$  if  $\exists p_1, \dots, p_k: p = q[p_1, \dots, p_k]$

For example,  $t \sqsubseteq b(\top, \top)$   
 $\sqsubseteq b(a(\top), \top)$   
 $\sqsubseteq b(\top, b(\top, \top))$

$\sqcup \{ p_1, \dots, p_k \}$  unique least upper bound (=nodes appearing in all  $p_i$ )

## Uniform and Earliest TOPs

“axiom” (start) tree

Deterministic TOP  $T = (Q, \Sigma, \Delta, \delta, A)$

**Uniform** = “if a state *blocks* at a certain input node, then ALL states translating that node *block*.”

$B \subseteq Q$  is *consistent*, if  $\bigcap_{q \in B} \text{dom}(q) \neq \emptyset$

Change  $q \in Q$  to states  $\langle q, B \rangle$  such that  
 $\langle q, B \rangle, a$ -rule is defined iff for all  $q' \in B$ , the  $\langle q', B \rangle, a$ -rule is defined.

All states that translate the current input node

---

### Lemma

Uniform  $T'$  equivalent to  $T$  can be constructed in exponential time.

## Uniform and Earliest TOPs

Deterministic TOP  $T = (Q, \Sigma, \Delta, \delta, A)$

$[[q]](s) :=$  outputs of  $T$ , starting in state  $q$

### Common Prefix Pattern

$$\text{pref}(q) := \sqcup \{ [[q]](s) \mid s \in T_\Sigma \text{ and } [[q]](s) \text{ defined} \}$$

$$p0(a(x)) \rightarrow p(x)$$

$$p0(e) \rightarrow e$$

$$p(a(x)) \rightarrow b(a(q(x)), p(x))$$

$$p(e) \rightarrow b(e, e)$$

$$q(a(x)) \rightarrow a(q(x))$$

$$q(e) \rightarrow e$$

$$\text{pref}(p0) = T$$

$$\text{pref}(p) = b(T, T)$$

## Uniform and Earliest TOPs

Deterministic TOP  $T = (Q, \Sigma, \Delta, \delta, A)$

$[[q]](s) :=$  outputs of  $T$ , starting in state  $q$

### Common Prefix Pattern

$$\text{pref}(q) := \sqcup \{ [[q]](s) \mid s \in T_\Sigma \text{ and } [[q]](s) \text{ defined} \}$$

$$p0(a(x)) \rightarrow p(x)$$

$$p0(e) \rightarrow e$$

$$p(a(x)) \rightarrow b(a(q(x)), p(x))$$

$$p(e) \rightarrow b(e, e)$$

$$q(a(x)) \rightarrow a(q(x))$$

$$q(e) \rightarrow e$$

$$\text{pref}(p0) = T$$

$$\text{pref}(p) = b(T, T)$$

### Definition

A uniform det. TOP is **earliest** if  $\text{pref}(q) = T$  for all states  $q$ .

## Uniform and Earliest TOPs

Deterministic TOP  $T = (Q, \Sigma, \Delta, \delta, A)$

### Theorem

$\text{pref}(q)$  can be computed in time  $O(|T| \cdot \eta(T)^2)$

$\eta(T)$  = maximal size of a minimal output tree produced by any state.

Proof fixpoint iteration.

At most  $\eta(T)$  iterations needed.

In each iteration, at most  $|T|$  variables are updated,  
and each update takes at most time  $O(\eta(T))$ .

### Theorem

$T_1, T_2$  earliest TOPs.

$T_1$  is **equivalent** to  $T_2$  iff  
their rules are **equal up to state renaming**.

## Uniform and Earliest TOPs

$$\begin{aligned} p_0(a(x)) &\rightarrow p(x) \\ p_0(e) &\rightarrow e \\ p(a(x)) &\rightarrow b(a(q(x)), p(x)) \\ p(e) &\rightarrow b(e, e) \\ q(a(x)) &\rightarrow a(q(x)) \\ q(e) &\rightarrow e \end{aligned}$$

M is uniform (because it is total).

→ Make it earliest

$$\text{pref}(p) = b(T, T)$$

In all right-hand sides, change  $p(x)$  into  $b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x))$

## Uniform and Earliest TOPs

$$\begin{array}{l} p_0(a(x)) \rightarrow \cancel{p(x)} \quad b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ p_0(e) \rightarrow e \\ \cancel{p(a(x)) \rightarrow b(a(q(x)), p(x))} \\ \cancel{p(e) \rightarrow b(e, e)} \\ q(a(x)) \rightarrow a(q(x)) \\ q(e) \rightarrow e \end{array}$$

M is uniform (because it is total).

→ Make it earliest

$$\text{pref}(p) = b(T, T)$$

In all right-hand sides, change  $p(x)$  into  $b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x))$

## Uniform and Earliest TOPs

$$\begin{array}{l} p_0(a(x)) \rightarrow \cancel{p(x)} \quad b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ p_0(e) \rightarrow e \\ \cancel{p(a(x)) \rightarrow b(a(q(x)), p(x))} \\ \cancel{p(e) \rightarrow b(e, e)} \\ q(a(x)) \rightarrow a(q(x)) \\ q(e) \rightarrow e \end{array}$$

M is uniform (because it is total).

→ Make it earliest

$$\text{pref}(p) = b(T, T)$$

In all right-hand sides, change  $p(x)$  into  $b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x))$

$$\begin{array}{l} \langle p, 1 \rangle(a(x)) \rightarrow a(q(x)) \\ \langle p, 1 \rangle(e) \rightarrow e \\ \langle p, 2 \rangle(a(x)) \rightarrow b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ \langle p, 2 \rangle(e) \rightarrow e \end{array}$$

## Uniform and Earliest TOPs

$$\begin{aligned} p_0(a(x)) &\rightarrow \cancel{p(x)} \quad b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ p_0(e) &\rightarrow e \\ \cancel{p(a(x))} &\rightarrow \cancel{b(a(q(x)), p(x))} \\ \cancel{p(e)} &\rightarrow \cancel{b(e, e)} \\ q(a(x)) &\rightarrow a(q(x)) \\ q(e) &\rightarrow e \end{aligned}$$

$$\begin{aligned} \langle p, 1 \rangle(a(x)) &\rightarrow a(q(x)) \\ \langle p, 1 \rangle(e) &\rightarrow e \\ \langle p, 2 \rangle(a(x)) &\rightarrow b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ \langle p, 2 \rangle(e) &\rightarrow e \end{aligned}$$

## Uniform and Earliest TOPs

$$\begin{array}{l} p_0(a(x)) \rightarrow \cancel{p(x)} \quad b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ p_0(e) \rightarrow e \\ \cancel{p(a(x)) \rightarrow b(a(q(x)), p(x))} \\ \cancel{p(e) \rightarrow b(e, e)} \\ q(a(x)) \rightarrow a(q(x)) \\ q(e) \rightarrow e \end{array}$$

$$\begin{array}{l} \langle p, 1 \rangle(a(x)) \rightarrow a(q(x)) \\ \langle p, 1 \rangle(e) \rightarrow e \\ \langle p, 2 \rangle(a(x)) \rightarrow b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\ \langle p, 2 \rangle(e) \rightarrow e \end{array}$$

$\langle p, 1 \rangle$  state-equivalent to  $q$   
 $\langle p, 2 \rangle$  state-equivalent to  $p_0$

## Uniform and Earliest TOPs

$$\begin{array}{l} p0(a(x)) \rightarrow \cancel{p(x)} \quad b(\overset{q}{\cancel{\langle p, 1 \rangle}}(x), \overset{p0}{\cancel{\langle p, 2 \rangle}}(x)) \\ p0(e) \rightarrow e \\ \cancel{p(a(x)) \rightarrow b(a(\overset{q}{q}(x)), \overset{p}{p}(x))} \\ \cancel{p(e) \rightarrow b(e, e)} \\ q(a(x)) \rightarrow a(q(x)) \\ q(e) \rightarrow e \end{array}$$

$$\begin{array}{l} \cancel{\langle p, 1 \rangle(a(x)) \rightarrow a(q(x))} \\ \cancel{\langle p, 1 \rangle(e) \rightarrow e} \\ \cancel{\langle p, 2 \rangle(a(x)) \rightarrow b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x))} \\ \cancel{\langle p, 2 \rangle(e) \rightarrow e} \end{array}$$

$\langle p, 1 \rangle$  state-equivalent to  $q$   
 $\langle p, 2 \rangle$  state-equivalent to  $p0$

## Uniform and Earliest TOPs

$$\begin{array}{l}
 p0(a(x)) \rightarrow \cancel{p(x)} \quad b(\overset{q}{\cancel{\langle p, 1 \rangle}}(x), \overset{p0}{\cancel{\langle p, 2 \rangle}}(x)) \\
 p0(e) \rightarrow e \\
 \cancel{p(a(x)) \rightarrow b(a(\overset{q}{q}(x)), \overset{p0}{p}(x))} \\
 \cancel{p(e) \rightarrow b(e, e)} \\
 q(a(x)) \rightarrow a(q(x)) \\
 q(e) \rightarrow e
 \end{array}$$

~~$$\begin{array}{l}
 \langle p, 1 \rangle(a(x)) \rightarrow a(q(x)) \\
 \langle p, 1 \rangle(e) \rightarrow e \\
 \langle p, 2 \rangle(a(x)) \rightarrow b(\langle p, 1 \rangle(x), \langle p, 2 \rangle(x)) \\
 \langle p, 2 \rangle(e) \rightarrow e
 \end{array}$$~~

Uniform&Earliest( T1 )  
w.  $p0$  renamed to  $q0$

*equals*

T2

$\langle p, 1 \rangle$  state-equivalent to  $q$   
 $\langle p, 2 \rangle$  state-equivalent to  $p0$

$$\begin{array}{l}
 q0(a(x)) \rightarrow b(q(x), q0(x)) \\
 q0(e) \rightarrow e \\
 q(a(x)) \rightarrow a(q(x)) \\
 q(e) \rightarrow e
 \end{array}$$

## Equivalence of TOPs with regular look-ahead

$M = (Q, \Sigma, \Delta, \delta, A, B)$

$B$  det. bottom-up tree automaton

With  $L(p1) \cap L(p2) = \emptyset$  for all states  $p1, p2$

$q(a(x1, x2)) \rightarrow t \langle p1, p2 \rangle$

Given TOPs w la  $M1, M2$ :

Change input symbol  $a$  into  $\langle a, (p1, p2), (u1, u2) \rangle$

Then  $M1, M2$  become ordinary TOPs (without lookahead)

Now, change  $M1, M2$  so that they

check if input tree is a correct relabeling wrt the automata  $B1, B2$ .

Finally,

make them uniform and earliest and check isomorphism as before.

## Applications

→ *XML query optimization*

Assume result to query Q1 is already materialized.

Given a new query Q2, check if Q2 equivalent to Q1.  
If so, return materialized result, instead of recomputing.

---

Possible extension

Q1 “subsumes” Q2, if for all inputs  $s$ ,  
Q2( $s$ ) can be obtained by deleting subtrees from Q1( $s$ ).

Conjecture

Given uniform and earliest Q1, Q2, it is decidable whether  
or not Q1 subsumes Q2.

Given a new query Q2, check if Q1 subsumes Q2.  
If so, return materialized result, with appropriate subtrees removed.

## Applications

→ *XML query optimization*

Assume result to query Q1 is already materialized.

Given a new query Q2, check if Q2 equivalent to Q1.  
If so, return materialized result, instead of recomputing.

---

Nicer....

Q1 “subsumes” Q2, if there exists a TOP Q3 such that  
for all inputs s,  
 $Q2(s) = Q3(Q1(s))$

Decidable?

Seems difficult...

## Applications

→ *XML query optimization*

Assume result to query Q1 is already materialized.

Given a new query Q2, check if Q2 equivalent to Q1.  
If so, return materialized result, instead of recomputing.

---

Nicer....

Q1 “subsumes” Q2, if there exists a TOP Q3 such that  
for all inputs s,  
 $Q2(s) = Q3(Q1(s))$

Decidable?

Seems difficult...

**THE END**