Frontiers of Tractability for Typechecking Simple XML Transformations

Wim Martens    Frank Neven

Limburgs Universitair Centrum
Belgium
Overview

- Introduction
- Tree Languages
- Tree Transformations : XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
Data Integration on the Web

WEB

Relational

Spatial

XML
Data Integration on the Web

- Relational
- Spatial
- XML

WEB

Query
Data Integration on the Web

- Relational
- Spatial
- XML

Query

WEB
Data Integration on the Web

Relational

XML

WEB

Spatial

Query

XML

XML

XML
Data Integration on the Web

Frontiers of Tractability for Typechecking Simple XML Transformations – p.3/2
What is Typechecking?

Database

Schema

Web Server

Schema

T

??
Our Focus

XML to XML transformations

Transformation/query languages:
  - XQuery
  - XSLT

We focus on structural top-down XSLT transformations (i.e. simple restructuring/filtering transformations)
XML Documents are Trees

<pizzas>
  <pizza type="margherita">
    <sauce> tomato </sauce>
    <cheese> mozzarella </cheese>
  </pizza>
  <pizza type="pepperoni">
    <sauce> tomato </sauce>
    <cheese> mozzarella </cheese>
    <topping> pepperoni </topping>
  </pizza>
</pizzas>
XML Documents are Trees

```xml
<pizzas>
  <pizza type="margherita">
    <sauce> tomato </sauce>
    <cheese> mozzarella </cheese>
  </pizza>
  <pizza type="pepperoni">
    <sauce> tomato </sauce>
    <cheese> mozzarella </cheese>
    <topping> pepperoni </topping>
  </pizza>
</pizzas>
```
Previous Results on XML Typechecking

  - Typechecking quickly turns undecidable when data or attribute values are incorporated.

- Milo, Suciu, Vianu (2000):
  - When only looking at structural properties of trees, typechecking is decidable for a large fragment of tree transformations (formalized by $k$-pebble transducers).
  - Complexity is high (non-elementary).
Overview

- Introduction
- Tree Languages
- Tree Transformations: XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
Tree Languages

DTDs:

books → book*
book → title, author+, chapter+
chapter → title, intro, sec+
sec → title, par+, sec*
Tree Languages

DTDs:

books  →  book*
book   →  title, author\(^+\), chapter\(^+\)
chapter →  title, intro, sec\(^+\)
sec    →  title, par\(^+\), sec*
Overview

- Introduction
- Tree Languages
- Tree Transformations: XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[
\begin{align*}
(q_0, \text{books}) & \rightarrow \text{books} \\
\text{books} & \rightarrow q_0 \text{books} \\
(p, \text{chapter}) & \rightarrow \text{chapter } p \\
(p, \text{title}) & \rightarrow \text{title} \\
(p, \text{sec}) & \rightarrow p \\
(q_0, \text{book}) & \rightarrow \text{book} \\
\text{book} & \rightarrow p \text{book} q \\
(q, \text{chapter}) & \rightarrow \text{chapter } q' \\
(q', \text{title}) & \rightarrow \text{title} \\
(q', \text{intro}) & \rightarrow \text{intro}
\end{align*}
\]
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books} \quad (q_0, \text{book}) \rightarrow \text{book} \]
\[(p, \text{chapter}) \rightarrow \text{chapter } p \quad (q, \text{chapter}) \rightarrow \text{chapter } q' \]
\[(p, \text{title}) \rightarrow \text{title} \quad (q', \text{title}) \rightarrow \text{title} \]
\[(p, \text{sec}) \rightarrow p \quad (q', \text{intro}) \rightarrow \text{intro} \]

Diagram:

- books
  - book
    - title
      - author
    - chapter
      - title
        - intro
      - sec
        - par
      - sec
        - par
Generate for each book the list of titles for each chapter, followed by a summary of the book.
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books} \]
\[\downarrow \]
\[p \]
\[\text{books} \]
\[(p, \text{chapter}) \rightarrow \text{chapter } p \]
\[(p, \text{title}) \rightarrow \text{title} \]
\[(p, \text{sec}) \rightarrow p \]

\[(q_0, \text{book}) \rightarrow \text{book} \]
\[\downarrow \]
\[p \]
\[q \]

\[(q, \text{chapter}) \rightarrow \text{chapter } q' \]
\[(q', \text{title}) \rightarrow \text{title} \]
\[(q', \text{intro}) \rightarrow \text{intro} \]
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books}\]
\[(p, \text{chapter}) \rightarrow \text{chapter } p\]
\[(p, \text{title}) \rightarrow \text{title}\]
\[(p, \text{sec}) \rightarrow p\]
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books} \]
\[(p, \text{chapter}) \rightarrow \text{chapter } p \]
\[(p, \text{title}) \rightarrow \text{title} \]
\[(p, \text{sec}) \rightarrow p \]

\[(q_0, \text{book}) \rightarrow \text{book} \]
\[(q, \text{chapter}) \rightarrow \text{chapter } q' \]
\[(q', \text{title}) \rightarrow \text{title} \]
\[(q', \text{intro}) \rightarrow \text{intro} \]
Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books}\]
\[(p, \text{chapter}) \rightarrow \text{chapter } p\]
\[(p, \text{title}) \rightarrow \text{title}\]
\[(p, \text{sec}) \rightarrow p\]
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.
Generate for each book the list of titles for each chapter, followed by a summary of the book.
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\((q_0, books) \rightarrow \text{books} \)

\((p, \text{chapter}) \rightarrow \text{chapter } p\)

\((p, \text{title}) \rightarrow \text{title}\)

\((p, \text{sec}) \rightarrow p\)

\((q_0, \text{book}) \rightarrow \text{book}\)

\((q, \text{chapter}) \rightarrow \text{chapter } q'\)

\((q', \text{title}) \rightarrow \text{title}\)

\((q', \text{intro}) \rightarrow \text{intro}\)
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books} \]
\[\text{books} \quad \| \quad q_0 \]
\[(p, \text{chapter}) \rightarrow \text{chapter } p \]
\[(p, \text{title}) \rightarrow \text{title} \]
\[(p, \text{sec}) \rightarrow p \]

\[(q_0, \text{book}) \rightarrow \text{book} \]
\[\text{book} \quad \| \quad p \quad q \]
\[(q, \text{chapter}) \rightarrow \text{chapter } q' \]
\[(q', \text{title}) \rightarrow \text{title} \]
\[(q', \text{intro}) \rightarrow \text{intro} \]
Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[(q_0, \text{books}) \rightarrow \text{books} \]  \[\text{books} \leftarrow q_0\]

\[(p, \text{chapter}) \rightarrow \text{chapter } p\]

\[(p, \text{title}) \rightarrow \text{title}\]

\[(p, \text{sec}) \rightarrow p\]

\[(q_0, \text{book}) \rightarrow \text{book}\]

\[\text{book} \leftarrow p \quad q\]

\[(q, \text{chapter}) \rightarrow \text{chapter } q'\]

\[(q', \text{title}) \rightarrow \text{title}\]

\[(q', \text{intro}) \rightarrow \text{intro}\]
XSLT: Example

Generate for each book the list of titles for each chapter, followed by a summary of the book.

\[
(q_0, \text{books}) \rightarrow \text{books} \\
(q_0, \text{book}) \rightarrow \text{book} \\
(p, \text{chapter}) \rightarrow \text{chapter } p \\
(p, \text{title}) \rightarrow \text{title} \\
(p, \text{sec}) \rightarrow p \\
(q, \text{chapter}) \rightarrow \text{chapter } q' \\
(q', \text{title}) \rightarrow \text{title} \\
(q', \text{intro}) \rightarrow \text{intro}
\]
Generate for each book the list of titles for each chapter, followed by a summary of the book.

Original tree:
Overview

- Introduction
- Tree Languages
- Tree Transformations: XSLT
  - The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
The Typechecking Problem

Given:

- input tree language $L_{in}$
- output tree language $L_{out}$
- XML-transformation $T$

Is it true that,

$$\forall t \in L_{in} : T(t) \in L_{out}$$
Known Results

The complexity of the typechecking problem [M., Neven 03]:

<table>
<thead>
<tr>
<th></th>
<th>Tree Automata</th>
<th>DTD(NFA)</th>
<th>DTD(DFA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>deletion, unbounded copy</td>
<td>EXPTIME</td>
<td>EXPTIME</td>
<td>EXPTIME</td>
</tr>
<tr>
<td>deletion, bounded copy</td>
<td>EXPTIME</td>
<td>EXPTIME</td>
<td>EXPTIME</td>
</tr>
<tr>
<td>no deletion, unbounded copy</td>
<td>EXPTIME</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>no deletion, bounded copy</td>
<td>EXPTIME</td>
<td>PSPACE</td>
<td>PTIME</td>
</tr>
</tbody>
</table>

The **PTIME** fragment is *very restricted*

Goal: *enlarge PTIME* fragment
Overview

- Introduction
- Tree Languages
- Tree Transformations : XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
What can we Delete?

Parametrize how much a certain transformation deletes:

Formally:

\[(q_1, a) \rightarrow q_2 a q_2 a\]
\[(q_2, a) \rightarrow a q_3 a\]
\[(q_3, a) \rightarrow q_4 q_4\]
\[(q_4, a) \rightarrow a\]
\[(q_5, a) \rightarrow q_6 a a\]
\[(q_5, a) \rightarrow q_6 a a\]
\[(q_6, a) \rightarrow q_7\]
\[(q_7, a) \rightarrow a q_8 a\]
\[(q_7, a) \rightarrow a q_8 a\]

<table>
<thead>
<tr>
<th>state</th>
<th>q_1</th>
<th>q_2</th>
<th>q_3</th>
<th>q_4</th>
<th>q_5</th>
<th>q_6</th>
<th>q_7</th>
<th>q_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>deletion depth</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>(\infty)</td>
<td>(\infty)</td>
<td>(\infty)</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>

\(\infty \equiv \text{“unbounded”}\)
What can we Delete?

\[(q_1, a) \rightarrow q_2 a q_2 a \quad (q_5, a) \rightarrow q_6 a a\]
\[(q_2, a) \rightarrow a q_3 a \quad (q_6, a) \rightarrow q_7\]
\[(q_3, a) \rightarrow q_4 q_4 \quad (q_7, a) \rightarrow a q_8 a\]
\[(q_4, a) \rightarrow a \quad (q_8, a) \rightarrow a a q_7\]

- del depth \(D\): maximally skip branches of length \(D\)
- del width \(W\): maximal no of states on top of rhs
- copy width \(C\): maximal no of states in rhs

\(\mathcal{T}_{\text{trac}}\): \(\max(C)\) and \(\max(W^D)\) are constant
What can we Delete?

Theorem: $TC[\mathcal{T}_{\text{trac}}, DTD(DFA)]$ is in time $O(n^{C \cdot W^D})$

Why is this relevant?

$C \cdot W^D$ is usually very small in practice, i.e.

- 1 for filtering transformations
- 2 or 3 for most other restructuring transformations
What can we Delete?

Theorem: $TC[\mathcal{T}_{\text{trac}}, DTD(DFA)]$ is in time $\mathcal{O}(n^{C \cdot W^D})$

What does $\mathcal{T}_{\text{trac}}$ allow?

Generate the list of titles for each chapter, followed by a summary of the book.

$$(q_0, \text{book}) \rightarrow \text{book}(p q)$$

$$(q, \text{chapter}) \rightarrow \text{chapter } q'$$  
$$(p, \text{chapter}) \rightarrow \text{chapter } p$$

$$(q', \text{title}) \rightarrow \text{title}$$  
$$(p, \text{title}) \rightarrow \text{title}$$

$$(q', \text{intro}) \rightarrow \text{intro}$$  
$$(p, \text{sec}) \rightarrow p$$

$q_0$: $C = 2$

$p$: $C = 1$  
$W = 1$  
$D = \infty$

$q$: $C = 1$  
$W = 1$  
$D = 1$

$q'$: $C = 0$

So, $C = 2$ and $W^D = 1$

Recursive deletion only when no simultaneous copy!

$$(q, a) \rightarrow qq$$
What can we Delete?

Theorem: $TC[T_{trac}, DTD(DFA)]$ is in PTIME

Can we extend this?

1. If $C$ is unbounded, then TC is PSPACE-hard.
2. $W^D$ is unbounded if
   - $W$ is unbounded and $D \geq 1$;
   - $W \geq 2$ and $D$ is unbounded

TC is also PSPACE-hard in either of these cases

So, “TC is tractable iff $C$ and $W^D$ are bounded”
Overview

- Introduction
- Tree Languages
- Tree Transformations : XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
When can we Copy?

DFAs in DTDs are too strong, consider

RE$^+$ expressions: concatenations of $a$ and $a^+$

Example: $a^+bbac^+bb^+$

Theorem: $TC[\mathcal{T}_{d,uc}, DTD(RE^+)]$ is in PTIME

... but $RE^+$ is very limited
When can we Copy?

Consider the following extensions of $RE^+$:

- allow $a$ and $a^*$;
- allow $a$ and $a$?;
- allow $a$ and $(a_1^+ + \cdots + a_n^+)$;
- allow $a$ and $(a_1 \cdots a_n)^+$;
- allow $a$ and $(a_1 + \cdots + a_n)^+$; and
- allow $(a_1 + \cdots + a_n)^+$ and $a^+$.

The inclusion problem is $\text{CONP}$-hard for all these extensions [M., Neven, Schwentick 04]
Overview

- Introduction
- Tree Languages
- Tree Transformations: XSLT
- The Typechecking Problem
- Tractable Deleting Transformations
- Tractable Copying Transformations
- Conclusion
Conclusion

We identified several interesting $\text{PTIME}$-fragments for TC:

- $\mathcal{T}_{\text{trac}}$, extended with $\text{XPath}\{\ell, \, /, \, \ast\}$, w.r.t. DTD(DFA)
- $\mathcal{T}_{\text{nd,bc}}$, extended with DFA, w.r.t. DTD(DFA)
- $\mathcal{T}_{\text{d,uc}}$, w.r.t. DTD(RE$^+$)

Slightly extending these fragments results in $\text{CONP}$-hardness

The typeckecking algorithms can be adapted to generate counterexamples in case of negative output

Our results show when it is needed to search for incomplete typechecking algorithms