Incremental XPath Evaluation

Wim Martens
Technical University of Dortmund

Joint work with:
Henrik Björklund
Wouter Gelade
Marcel Marquardt
Outline

• Motivation
• Terminology
• Results
• Final Remarks
Outline

• Motivation
• Terminology
• Results
• Final Remarks
Motivation

```
lib
  books
    ...
  papers
    paper
      keywords title author
      "Constraints" "XPath" "..." "Superman"
    paper
      keywords title author
      "Schemas" "XML" "..." "Idiot"
```
I’m interested in: papers about XML or XPath which are not written by Idiot
Motivation

I'm interested in:
papers about XML or XPath which are not written by Idiot

Local Database
Motivation

I’m interested in:
papers about XML or XPath which are not written by Idiot

Local Database
Motivation

I’m interested in:
papers about XML or XPath which are not written by Idiot

Local Database
Motivation

I’m interested in: papers about XML or XPath which are not written by Idiot

Local Database
I'm interested in:
papers about XML or XPath which are not written by Idiot

Local Database
Motivation

I’m interested in:
papers about XML or XPath which are not written by Idiot

Local Database
I’m interested in:
papers about XML or XPath which are not written by Idiot
I’m interested in:
papers about XML or XPath which are not written by Idiot

Local Database
Motivation

So this is:
Efficient XPath View Maintenance for XML Data
Outline

• Motivation
• Terminology
• Results
• Final Remarks
Terminology

Two Versions of the problem:

(1) We want to maintain a view:

Incremental View Maintenance

(2) We want to maintain (non)-satisfaction of a trigger:

Incremental Boolean Maintenance
Terminology

Two Versions of the problem:

1. We want to maintain a view:
   - Incremental View Maintenance

2. We want to maintain (non)-satisfaction of a trigger:
   - Incremental Boolean Maintenance

Incremental XPath Evaluation
Problem Definition

Incremental Boolean Maintenance

Given:
- XPath query \( Q \)
- XML document \( D \)
- Update \( u \), that updates \( D \) to \( D' \)

Question:
Does \( D' \models Q \)?

(Does \( Q \) return a non-empty answer on \( D' \)?)
Problem Definition

Incremental Boolean Maintenance

Question: Does $D' \models Q$?

We can maintain an auxiliary data structure $\text{Aux}(D)$.

Algorithms are evaluated w.r.t.:  
- Size of $\text{Aux}(D)$  
- Time needed to  
  - compute whether $D' \models Q$  
  - update $\text{Aux}(D)$ to $\text{Aux}(D')$
Problem Definition

Incremental Boolean Maintenance

Updates:
- **Relabel**(u,a): overwrite label of u with a
- **InsertNS**(u,a): insert leaf labeled a as next sibling of u
- **InsertFC**(u,a): insert leaf labeled a as first child of u
- **Delete**(u): delete subtree rooted at u
Problem Definition

Incremental View Maintenance

Similar to Boolean maintenance, but:
We want to maintain the set \( Q(D) \) of output nodes

Given update \( u \) that updates \( D \) to \( D' \),
compute update \( v \) that updates \( Q(D) \) to \( Q(D') \)
Prove that XPath Maintenance is possible in
- time polylog(D) . poly(Q)
- auxspace poly(D) . poly(Q)
Outline

- Motivation
- Terminology
- Results
- Final Remarks
### Result Overview

#### Core XPath

<table>
<thead>
<tr>
<th>Operator(s)</th>
<th>Time:</th>
<th>Auxiliary Size:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core XPath</td>
<td>$\polylog(D) \cdot 2^{O(Q)}$</td>
<td>$D \cdot 2^{O(Q)}$</td>
</tr>
<tr>
<td>Core XPath</td>
<td>$\text{depth}(D) \cdot \log(\text{width}(D)) \cdot 2^{O(Q)}$</td>
<td>$D \cdot 2^{O(Q)}$</td>
</tr>
</tbody>
</table>

#### Boolean Maintenance

<table>
<thead>
<tr>
<th>Operator(s)</th>
<th>Time:</th>
<th>Auxiliary Size:</th>
</tr>
</thead>
<tbody>
<tr>
<td>//, [ ] and, or, not</td>
<td>$\log(D) \cdot \text{poly}(Q)$</td>
<td>$D \cdot Q^3$</td>
</tr>
<tr>
<td>nextsib, follow-sib [ ], and</td>
<td>$\text{depth}(D) \cdot \log(\text{width}(D)) \cdot \text{poly}(Q)$</td>
<td>$D \cdot Q^3$</td>
</tr>
<tr>
<td>//, ns, fs [ ], and</td>
<td>$\text{depth}(D) \cdot \log(\text{width}(D)) \cdot \text{poly}(Q)$</td>
<td>$D \cdot Q^3$</td>
</tr>
</tbody>
</table>

#### View Maintenance
## Result Overview

### Core XPath

<table>
<thead>
<tr>
<th>_operation</th>
<th>Time</th>
<th>AuxSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>polylog(D). (2^{O(Q)})</td>
<td>(D \cdot 2^{O(Q)})</td>
<td></td>
</tr>
</tbody>
</table>

### Core XPath

<table>
<thead>
<tr>
<th>operation</th>
<th>Time</th>
<th>AuxSize</th>
</tr>
</thead>
</table>
depth(D). \(\log(width(D)) \cdot 2^{O(Q)}\)| \(D \cdot 2^{O(Q)}\)|

### Boolean Maintenance

<table>
<thead>
<tr>
<th>operation</th>
<th>Time</th>
<th>AuxSize</th>
</tr>
</thead>
</table>
//, [ ] and, or, not| depth(D). \(Q\)| \(D \cdot Q\)|

<table>
<thead>
<tr>
<th>operation</th>
<th>Time</th>
<th>AuxSize</th>
</tr>
</thead>
</table>
nextsib, follow-sib [ ], and| \(\log(D) \cdot \text{poly}(Q)\)| \(D \cdot Q^3\)|

<table>
<thead>
<tr>
<th>operation</th>
<th>Time</th>
<th>AuxSize</th>
</tr>
</thead>
</table>
//, ns, fs [ ], and| depth(D). \(\log(width(D)) \cdot \text{poly}(Q)\)| \(D \cdot Q^3\)|

### View Maintenance
**Theorem (Balmin, Papakonstantinou, Vianu TODS 2005)**

Incremental Maintenance for unranked tree automaton $A$ on document $D$ is in
- time $(\log(D))^2 \cdot \text{poly}(A)$
- auxspace $D \cdot \text{poly}(A)$

---

**Theorem**

A Core XPath query $Q$ can be compiled into an unranked tree automaton of size $2^{O(Q)}$ in time $2^{O(Q)}$

(Standard techniques only seem to give $2^{O(Q^2)}$)
## Full Core XPath

**Corollary**

Incremental Boolean Maintenance for Core XPath is possible in
- time \((\log(D))^2 \cdot 2^{O(Q)}\)
- auxspace \(D \cdot 2^{O(Q)}\)

Similarly, with a different Balmin et al. [TODS 05] result:

**Corollary**

Incremental Boolean Maintenance for Core XPath is possible in
- time \(\text{depth}(D) \cdot \log(\text{width}(D)) \cdot 2^{O(Q)}\)
- auxspace \(D \cdot 2^{O(Q)}\)
## Result Overview

<table>
<thead>
<tr>
<th>Boolean Maintenance</th>
<th>View Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Core XPath</strong></td>
<td></td>
</tr>
<tr>
<td>Time: polylog(D) . 2^{O(Q)}</td>
<td></td>
</tr>
<tr>
<td>AuxSize: D . 2^{O(Q)}</td>
<td></td>
</tr>
<tr>
<td><strong>Core XPath</strong></td>
<td></td>
</tr>
<tr>
<td>Time: depth(D) . log(width(D)) . 2^{O(Q)}</td>
<td></td>
</tr>
<tr>
<td>AuxSize: D . 2^{O(Q)}</td>
<td></td>
</tr>
<tr>
<td><em>/ //, [ ] and, or, not</em></td>
<td>Time: depth(D) . Q</td>
</tr>
<tr>
<td>AuxSize: D . Q</td>
<td></td>
</tr>
<tr>
<td>nextsib, follow-sib [ ], and</td>
<td>Time: log(D) . poly(Q)</td>
</tr>
<tr>
<td>AuxSize: D . Q^3</td>
<td></td>
</tr>
<tr>
<td>*/ //, ns, fs [ ], and</td>
<td>Time: depth(D) . log(width(D)) . poly(Q)</td>
</tr>
<tr>
<td>AuxSize: D . Q^3</td>
<td></td>
</tr>
</tbody>
</table>
Downward XPath

child (/), descendant (//), predicate [], and, or, not
Downward XPath

child (/), descendant (//), predicate [], and, or, not

I’m interested in:
papers about XML or XPath which are not written by Idiot

paper[.(./XPath or .//XML) and not ./author/Idiot]
Downward XPath

child (/), descendant (//), predicate [], and, or, not

I’m interested in:
papers about XML or XPath which are not written by Idiot

paper[(./XPath or ./XML) and not ./author/Idiot]
Downward XPath
Downward XPath
Downward XPath
Downward XPath

```xml
<html>
<head>
<title>Downward XPath</title>
</head>
<body>

 paper
           1
          ...
            2
                and
                 3 or
                   4 "XPath"
                   5 "XML"
        not 8
            7 author
               "Idiot"
            6

 papers
       paper
          2
            1
         paper
            3
               keywords
               1 {2,3,8}
               title
               4 {8}
               author
               5 {8}
        "Constraints" {8} "XPath" {4,8} "..." "Superman" {8}
       paper
          4
            5
         paper
            6
               keywords
               2 {2,3,8}
               title
               7 {8}
               author
               8 {8}
        "XPath" {4,8} "XML" {5,8} "..." "Superman" {8}

</body>
</html>
```
Downward XPath

```
paper 1
   
   2
   and
   or
   
   3
   “XPath” 4
   “XML” 5

not 8
   
   author 7
   “Idiot” 6

papers

paper 1,2,3,8
   
   keywords 2,3,8
   title 8
   author 8

“Constraints” 8 “XPath” 4,8 “...” “Idiot” 8

paper 1,2,3,8
   
   keywords 2,3,8
   title 8
   author 8

“XPath” 4,8 “XML” 5,8 “...” “Superman” 8
```
Downward XPath

```
paper 1
  ... 
  3 or
  4 "XPath"
  5 "XML"
  2 and
  6 not
  7 author
    8 "Idiot"

papers

paper 1, 2, 3, 4, 5, 6, 7, 8
  keywords 2, 3, 8, 4, 8
  title 8
  author 8

paper 1, 2, 3, 4, 5, 6, 7, 8
  keywords 2, 3, 8, 4, 8
  title 8
  author 8
```
Downward XPath

```
  paper 1
   
   2        and
     |        
    or
   
  “XPath”  “XML”
  4        5
   
  not 8
   
  “Idiot”
  6

papers

  paper 1
    
    2
     
     and
      
      or
       
       “XPath”  “XML”
       4        5
        
        not 8
        
        “Idiot”
        6

papers

  paper
    
    keywords
      
      “Constraints”  “XPath”  “…”
      {8}  {4,8}  {2,3,8}

  title
    
    {8}

  author
    
    “Idiot”
    {8,6}

  paper
    
    keywords
      
      “XPath”  “XML”  “…”
      {4,8}  {5,8}  {8}

  title
    
    {8}

  author
    
    “Superman”
    {8}
```
Downward XPath
Downward XPath

```
   paper
     \-- paper
       \-- keywords {2,3,8}  title {8}  author {8,7}  keywords {2,3,8}  title {8}  author {8}
         |  |  1
         |  |  |  or
         |  |  |  \-- author {8}
         |  |  |  |  "XPath" 4
         |  |  |  |  "XML" 5
         |  |  |  \-- not 8
         |  |  \-- or
         |  \-- author 7
         \-- "Idiot" 6

   papers
     \-- paper {1,2,3,8}
```

"Constraints" {8}  "XPath" {4,8}  "...
"Idiot" {8,6}  "XPath" {4,8}  "XML" {5,8}  "...
"Superman" {8}
**Theorem**

Incremental Boolean Maintenance for Downward XPath is possible in
- time $\text{depth}(D) \cdot Q$
- auxspace $D \cdot Q$
Downward XPath

Theorem

Incremental View Maintenance for Downward XPath is possible in
- time $\text{depth}(D) \cdot Q$
- auxspace $D \cdot Q$
(in restricted cases)
Downward XPATH

**Theorem**

Incremental View Maintenance for Downward XPATH is possible in
- time $\text{depth}(D) \cdot Q$
- auxspace $D \cdot Q$
(in restricted cases)

Restricted cases: root element must be selected

**Example**

```
paper[(.//XPath or .//XML) and not ./author/Idiot]
```
<table>
<thead>
<tr>
<th>Core XPath</th>
<th>Time: ( \text{polylog}(D) \cdot 2^{O(Q)} )</th>
<th>AuxSize: ( D \cdot 2^{O(Q)} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core XPath</td>
<td>Time: ( \text{depth}(D) \cdot \log(\text{width}(D)) \cdot 2^{O(Q)} )</td>
<td>AuxSize: ( D \cdot 2^{O(Q)} )</td>
</tr>
<tr>
<td>//, [ ], and, or, not</td>
<td>Time: ( \text{depth}(D) \cdot Q )</td>
<td>AuxSize: ( D \cdot Q )</td>
</tr>
<tr>
<td>nextsib, follow-sib [ ], and</td>
<td>Time: ( \log(D) \cdot \text{poly}(Q) )</td>
<td>AuxSize: ( D \cdot Q^3 )</td>
</tr>
<tr>
<td>//, ns, fs [ ], and</td>
<td>Time: ( \text{depth}(D) \cdot \log(\text{width}(D)) \cdot \text{poly}(Q) )</td>
<td>AuxSize: ( D \cdot Q^3 )</td>
</tr>
</tbody>
</table>
Forward XPath

child (/), descendant (//), nextsib, following-sib, predicate [], and
Forward XPath
child (/), descendant (//), nextsib, following-sib, predicate [], and

First: extremely shallow trees
Forward XPath
child (/), descendant (//), nextsib, following-sib, predicate [], and

First: extremely shallow trees
(well, strings)
Forward XPath

nextsib, following-sib, predicate [], and

First: extremely shallow trees
(well, strings)
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath
nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath
nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and
Forward XPath

nextsib, following-sib, predicate [], and

**Theorem**

Incremental Boolean Maintenance for Forward XPath is possible in
- time $\log(D) \cdot \text{poly}(Q)$
- auxspace $D \cdot Q^3$

on strings
Forward XPath
child (/), descendant (//), nextsib, following-sib, predicate [], and

Combining this idea with the \texttt{depth(D)} algorithm:

\begin{center}
\begin{tabular}{|l|}
\hline
\textbf{Theorem} \\
Incremental Boolean Maintenance for Forward XPath is possible in \\
- time $\text{depth}(D) \log(\text{width}(D)) \cdot \text{poly}(Q)$ \\
- auxspace $D \cdot Q^3$  \\
on trees \\
\hline
\end{tabular}
\end{center}
Outline

• Motivation
• Terminology
• Results
• Final Remarks
Final Remarks

- Incremental XPath Evaluation is interesting
- Boolean version is already non-trivial
Final Remarks

- We like depth(D) maintenance for downward XPath
- Our Algorithm for Forward XPath is quite involved...
- But without NextSibling, it's much simpler
Outlook

The big questions:

For which XPath fragments is Boolean Maintenance possible in
- time \( \text{polylog}(D) \cdot \text{poly}(Q) \)
- auxspace \( \text{poly}(D) \cdot \text{poly}(Q) \)
Outlook

The big questions:

For which XPath fragments is View Maintenance possible in:
- **time** \( \text{polylog}(D) \cdot \text{poly}(Q) \)
- **auxspace** \( \text{poly}(D) \cdot \text{poly}(Q) \)
### Result Overview

#### Boolean Maintenance

<table>
<thead>
<tr>
<th>Core XPath</th>
<th>Time:</th>
<th>AuxSize:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>polylog(D) . $2^{O(Q)}$</td>
<td>$D . 2^{O(Q)}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Core XPath</th>
<th>Time:</th>
<th>AuxSize:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>depth(D) . log(width(D)) . $2^{O(Q)}$</td>
<td>$D . 2^{O(Q)}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>/, //, [ ] and, or, not</th>
<th>Time:</th>
<th>AuxSize:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>depth(D) . Q</td>
<td>$D . Q$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>nextsib, follow-sib [ ], and</th>
<th>Time:</th>
<th>AuxSize:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>log(D) . poly(Q)</td>
<td>$D . Q^3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>/, //, ns, fs [ ], and</th>
<th>Time:</th>
<th>AuxSize:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>depth(D) . log(width(D)) . poly(Q)</td>
<td>$D . Q^3$</td>
</tr>
</tbody>
</table>

#### View Maintenance