Schema-based independence analysis for XML updates

Michael Benedikt and James Cheney

University of Oxford and University of Edinburgh

VLDB 2009
Motivation

• Suppose we want to maintain multiple (materialized) views or constraints
  • expressed by queries Q1, Q2, ...

• When the database is updated
  • If we can determine (quickly) that query and update are independent
  • then can skip view/constraint maintenance
Query-update independence
Query-update independence
Query-update independence
Query-update independence
Query-update independence

\[ Q(U(DB)) = \ ?? \]
Contributions

• a static analysis
• that safely detects independence
• and takes schema into account
• for XQuery Update with all XPath axes
• fast enough to be useful as an optimization
XML updates

• SQL has update expressions
  • Allow in-place modification
  • Implemented efficiently
• XQuery does not
• W3C developing XQuery Update Facility
  • Goal: SQL-like updates for XML?
Update Example

• First, let’s illustrate the semantics of W3C updates via an example.

  delete /a/b,
  insert <foo>bar</foo>
  before /a/b

• Note: this is **not** equivalent to doing nothing!
Update Example

delete /a/b,
insert <foo>bar</foo>
before /a/b
First **collect** updates

```
delete /a/b,
insert <foo>bar</foo>
before /a/b
```
First **collect** updates

delete /a/b,  
insert `<foo>bar</foo>`  
before /a/b
Then reorder & apply

delete /a/b,
insert <foo>bar</foo>
before /a/b
Then reorder & apply

delete /a/b,
insert <foo>bar</foo>
before /a/b
Then reorder & apply

delete /a/b,
insert <foo>bar</foo>
before /a/b
Independence example

for $x$ in /c/a
return d[$x]$
Independence example

for $x$ in /c/a
return d[$x$]
Independence example

for $x$ in /c/a
return d[$x$]
Independence example

```
for $x$ in /c/a
return d[$x]
```
Independence example

for $x$ in /c/a
  return d[$x]
Independence non-example

for $x$ in /a
return d[$x$]
Independence **non-example**

```
for $x in /a
  return d[$x]
```

```
for $x in /a
  return d[$x]
```

```
for $x in /a
  return d[$x]
```

```
for $x in /a
  return d[$x]
```
Independence **non-example**

```
for $x$ in /a
return d[$x]
```

```
≠
```

```
for $x$ in /a
return d[$x]
```
Our approach

• Exploits a **schema** $S$ that describes the input
• Statically calculate:
  • $c =$ **Copied nodes** of $Q$
  • $a =$ **Accessed Nodes** of $Q$
  • $u =$ **Updated Nodes** of $U$
• $c$, $a$, $u$ are sets of type names in $S$
Our approach

• We show that Q and I are independent modulo S if:

• $a$ is disjoint from $u$
  • that is, the update has no impact on accessed nodes

• and $c/\ast$ is disjoint from $u$
  • that is, the update does not impact any copied nodes or their descendants
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

```
for $x$ in /c/a
return d[$x]
```
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
return d[$x$]
for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
return d[$x$]

Updated node
Analysis example

for $x$ in /c/a
return d[$x$]
Analysis example

for $x$ in /c/a
    return d[$x]

Updated nodes disjoint from accessed or copied nodes
Analysis example

for $x$ in /c/a
return d[$x$]
for $x$ in /a
return d[$x$]
Analysis example II

for $x$ in /a
return d[$x]

Updated node avoids accessed & copied nodes
Analysis example II

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]
Analysis example II

```
for $x$ in /a
  return d[$x]
```
Analysis example II

Updated node is a descendant of a copied node!

for $x$ in /a
return d[$x$]
for $x$ in /a
return d[$x$]
Analysis example II

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]

for $x$ in /a
return d[$x$]
Analysis example II

Query and update are **not** independent

for $x$ in /a
return d[$x$]
Oversimplification Warning

- These preceding examples are drastic oversimplifications
  - Glossing over schema aspects
  - The real story (in paper) is more complicated
  - But this is the basic idea
- We’ve formalized and proved soundness
  - Exact independence checking undecidable for full XQuery/Update; hard even for special cases
Experimental results

- Implemented analysis in OCaml
- Built testbed of queries and updates
  - XMark/XPathMark
- Considered all query-update pairs
  - Tested whether they are independent on fixed input
  - Used static analysis to avoid recomputing queries after updates
- Used Galax 1.1 with 1-2MB XMark documents (but benefits improve as size increases)
Independence hit rate

44% overall
Does it help view maintenance?
Does it help view maintenance?
Related work

- Independence for XPath-based queries/updates
  - Raghavachari and Shmueli [2006]:
- Commutativity analysis
  - Ghelli, Rose & Simeon [2007,2008]
- XML projection
  - Marian & Simeon [2003]
  - Benzaken et al. [2006]
Future work

• Several opportunities for improving analysis
  • more accurate XPath typing (following Benzaken et al. [2006])?
  • combine schema-based techniques with path-based (following Raghavachiari & Shmueli 2006])?
  • combine static techniques with incremental view maintenance?
Conclusions

• We’ve presented a query-update independence analysis that is:
  • static
  • schema-based
  • for XML queries and (W3C-style) updates
• It’s implemented and experimentally validated
  • savings of 25% even for small documents
  • low overhead; independent of data size