

Inconsistent Path Detection for XML IDEs

Pierre Genevès Nabil Layaïda

CNRS

INRIA

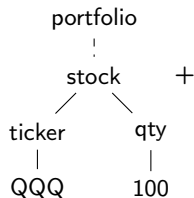
May 25th, 2011

33rd International Conference on Software Engineering
Honolulu, HI, USA

A Simple XQuery Program

- Generate alerts for news related to stocks in portfolio:

```
for $s in doc("portfolio.xml")//stocks/stock
for $line in doc("news.xml")/news/headline
where contains($line, $s/name)
return <alert>{$s/ticker, $line/parent::*//summary}</alert>
```

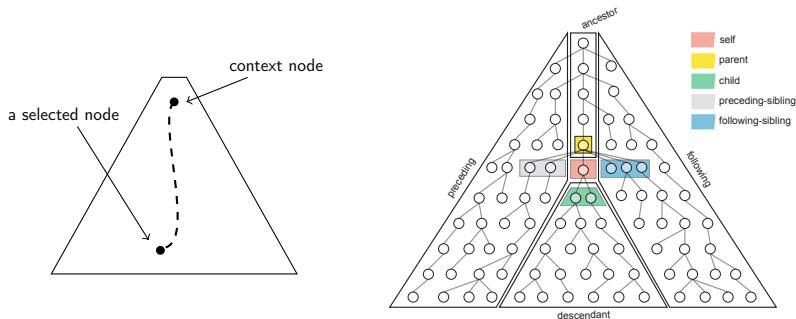


→ <alert>QQQ, Nasdaq falls...</alert>

- Search, selection and information extraction done using XPath expressions

Zoom on XPath Expressions

- General form: $(axis::nodetest[filter]'//')^n$



- Succinct but very powerful
- Describe **binary relations** between context and selected nodes
- Standard recommended by the W3 Consortium
- Central component (XSLT, XQuery, XML Schema, XPointer...)

The Path Consistency Problem

- In real life, XML data are complex and queried using complex paths
 - Paths are error-prone for programmers
 - **Two types of inconsistencies:**
 - self-contradicting paths
 - paths violating schema constraints (more frequent since path and schemas are updated independently)
 - Inconsistencies are hard to detect
- Detect path inconsistencies automatically
- Detect them all (be sound and complete)

```
a/b[following-sibling::c/parent::d]
```

```
self::a/child::e
```

```
a[b*,c,d+]
```

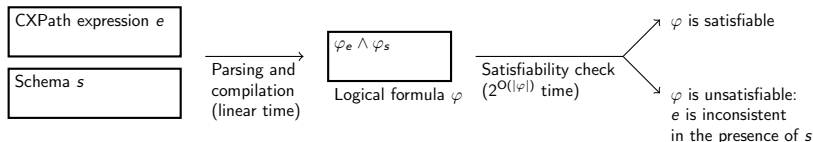
The Path Consistency Problem: Formal Overlook

- An expression e , evaluated from a context node x in a tree t , returns a set of matching nodes $e(t, x)$

XPath expression e is **inconsistent** $\Leftrightarrow \forall t \forall x \in t, e(t, x) = \emptyset$
 e is **inconsistent in the presence of** s $\Leftrightarrow \forall t \vdash s \forall x \in t, e(t, x) = \emptyset$

- The problem of determining whether an XPath expression is inconsistent is:
 - **Undecidable** for XPath in general
 - **EXPTIME** for the navigational core fragment of XPath (XPath \leftrightarrow first-order logic over trees)
 - **EXPTIME** for XPath in the presence of schemas (regular tree grammars \leftrightarrow monadic second-order logic over trees)

Proposed Approach



• Reduction to satisfiability of a unifying tree logic:

- The μ -calculus with converse of finite trees of [Geneves-PLDI07]
- CXPath expressions and schemas are compiled linearly into the logic
- Formula φ is checked for satisfiability in time complexity $2^{O(|\varphi|)}$
- Exact algorithm (sound and complete)
- The core logical solver is available online:

<http://wam.inrialpes.fr/websolver>

Other Applications and Demo

- Dead code elimination (e.g. loops over inconsistent paths)

```
for $x in //news[article]/headline  
  return { ... }
```

← inconsistent w.r.t schema

← dead code

- Code optimization (e.g. removing redundancies in path expressions)
(the decision procedure can also check path equivalence)
- First IDE for XML augmented with static detection of inconsistent paths
(demo)