

Data Integration: XPath/XQuery

Jan Chomicki

University at Buffalo and Warsaw University

April 12, 2007

XPath

Data model

- tree-based
- nodes: root, element, attribute, text,...
- document order: left-to-right prefix traversal

Path expression

- describes a set of paths in a document
- returns a sequence of nodes in document order
- evaluated in a *context*:
 - node
 - position
 - size
- absolute (starting at root) or relative
- consists of steps separated by /
- wildcards
- union (|), intersection, difference

axis::nodeTest stepQualifiers

- *axis*:
 - *forward*: child, descendant, following-sibling, following, self, descendant-or-self
 - *backward*: parent, ancestor, preceding-sibling, preceding, ancestor-or-self
 - attribute
- *node test*: name test (name or wildcard), kind test
- *step qualifiers*: predicate expressions (in square brackets)

Abbreviated syntax

- ① child is the default axis, can be omitted
- ② the attribute axis can be abbreviated to @
- ③ // is short for /descendant-or-self::node()/
- ④ . is short for self::node()
- ⑤ .. is short for parent::node()
- ⑥ a positive integer K is short for [position()=K]

Integrity constraints in XML Schema

Keys

```
<( key | unique) name=KeyName>  
<selector xpath=Path/>  
<field xpath=Path1/>  
...  
<field xpath=PathN/>  
</key>
```

Foreign keys

```
<keyref name=RefName refer=KeyName>  
<selector xpath=Path/>  
<field xpath=Path1/>  
...  
<field xpath=PathN/>  
</keyref>
```

Features

- functional
- compositional: expressions can be nested arbitrarily
- recursion
- declarative: influenced by SQL (and OQL)

XQuery expressions

- Constants: numbers, strings,...
- Variables
- XPath expressions
- Element/attribute constructors
- Operators and functions: arithmetic,...
- FLWOR expressions
- Quantifiers
- Aggregation
- User-defined functions

FLWOR expressions

```
for variableRangeSpecifications
let variableDefinitions
where condition
order by orderExpression
return resultExpression
```

User-defined functions

```
declare function Name(Arguments)
as Type
{Expression}
```

Storing XML documents in relational databases

Storing nodes and edges of the document tree

- a binary edge relation
- implementing XPath requires recursion (SQL3)

Encoding the tree structure using ranges

- range of child \subset range of parent
- queries w/o recursive functions can be translated to SQL2