University of Washington
Department of Computer Science & Engineering

Managing XML and Semistructured Data

Lecture 8: Query Languages - XML-QL

Prof. Dan Suciu

Spring 2001

In this lecture

- XML-QL features
  - patterns, constructors, nested queries
  - skolem functions
- Constructing trees in XML-QL
- XML-QL compared with XQuery

Resources

- XML-QL: A Query Language for XML by Deutsch, Fernandez, Florescu, Levy, Suciu, in WWW8.
- **Data on the Web** Abiteboul, Buneman, Suciu : *section 5.1*

XML-QL

- First declarative language for XML
- How to obtain a query language for XML fast ?
  - Assume OEM as data model
  - Use features from UnQL and StruQL
    - Patterns
    - Templates
    - Skolem functions
  - Design XML-like syntax

Patterns in XML-QL

Find all authors who published in Morgan Kaufmann:

```
WHERE <book language="french">
          <publisher>
              <name> Morgan Kaufmann </>
          </>
          <author> $A </>
      </book> in "www.a.b.c/bib.xml"
CONSTRUCT <author> $A </>
```

Abbreviation: </> closes any tag.

Patterns in XML-QL

Find all languages in which Jones' coauthors have published:

```
where <book language=$X>
          <author> $A </author>
      </book> in "www.a.b.c/bib.xml"
      <book>
          <author> $A </author>
          <author> Jones </author>
      </book> in "www.a.b.c/bib.xml"
construct <result> $X </>
```

There is a join here…

Constructors in XML-QL

Find all authors and the languages in which they published:

```
where <book language = $L>
          <author> $A </>
      </> in "www.a.b.c/bib.xml"
construct <result> <author> $A </> <lang> $L </> </>
```

Result is:

```
<result> <author>Smith</author> <lang>English </lang> </result>
<result> <author>Smith</author> <lang>Mandarin</lang> </result>
<result> <author>Doe </author> <lang>English </lang> </result>
. . . .
```

Nested Queries in XML-QL

Find all authors and the languages in which they published; group by authors:

```
WHERE <book.author> $A </> in "www.a.b.c/bib.xml"
CONSTRUCT <result> <author> $A </>
                  WHERE <book language = $L>
                              <author> $A </>
                        </> in "www.a.b.c/bib.xml"
                  CONSTRUCT <lang> $L </>
              </>
```

Note: book.author is a (regular) path expression

Result is:

```
<result> <author>Smith</author>
         <lang>English</lang>
         <lang>Mandarin</lang>
         <lang>…</lang>
         …
</result>
<result> <author>Doe</author>
         <lang>English</lang>
         …
</result>
```

Skolem Functions in XML-QL

Same query, with Skolem functions

```
WHERE <book language = $L>
            <author> $A </>
      </> in "www.a.b.c/bib.xml"
CONSTRUCT <result id=F($A)> <author> $A</>
                                <lang> $L </>
                  </>
```

Assumptions:

- the ID attribute is always id
- default Skolem function for author is G($A), for lang is H($A, $L) (why ?)

Skolem Functions in XML-QL

Object fusion with Skolem functions and block structure

- Compile a complete list of authors, from two sources

```
{ WHERE <book> <author> $A </>
              <title> $T </>
        </> in "www.a.b.c/bib.xml"
  CONSTRUCT <person id=F($A)>
              <name id=G($A)> $A </>
              <booktitle> $T</>    /* implicit Skolem function H($A, $T) */
            </>
}
{ WHERE <paper> <author> $A </>
              <title> $T </>
              <journal> $J </>
        </> in "www.d.e.f/papers.xml"
  CONSTRUCT <person id=F($A)>
              <name id=G($A)> $A </>
              <papertitle> $T</>    /* implicit Skolem function J($A, $T) */
              <journaltitle> $J</>    /* implicit Skolem function K($A, $T) */
            </>
}
```

Result:

(some have only books,
Others only papers,
Others have both)

```
<person> <name>Smith</name>
         <booktitle>Book1</booktitle >
         <booktitle>Book2</booktitle >
</result>
<person> <name>Jones</name>
         <booktitle>Book3</booktitle >
         <papertitle>paper1</papertitle >
         <journaltitle>journal1</journaltitle >
</result>
<person> <name>Mark</name>
         <papertitle>paper2</papertitle >
         <journaltitle>journal3</journaltitle >
</result>
…
```

Skolem Functions in XML-QL

"Wrong" query number 1:

```
WHERE <book language = $L>
            <author> $A </>
      </> in "www.a.b.c/bib.xml"
CONSTRUCT <result id=F($A)>
                  <author id=G($A)> $A</>
                  <lang id=H($A)> $L </>
              </>
```

What is "wrong" here ?

Skolem Functions in XML-QL

"Wrong" query number 2:

```
WHERE <book language = $L>
            <author> $A </>
          </> in "www.a.b.c/bib.xml"
CONSTRUCT <result id=F($A,$L)>
                    <author id=G($A)> $A</>
                    <lang id=H($A,$L)> $L </>
                  </>
```

What is "wrong" here ?

Skolem Functions in XML-QL

"Wrong" query number 3:

```
{ WHERE <book language = $L> <author> $A </>
          </> in "www.a.b.c/bib.xml"
  CONSTRUCT <author id=F($A)>
                    <lang id=H($A,$L)> $L </>
                  </> }
{ WHERE <person> <city> $C </>
                    <fluent-in> $X </>
          </> in "www.a.b.c/bib.xml"
  CONSTRUCT <location id=G($C)>
                    <lang id=H($C,$L)> $L </>
                  </> }
```

Three Rules to Construct Only Trees

Rule 1: nested elements must have Skolem functions with arguments such that args1 $\subseteq$ args2

```
CONSTRUCT <tag1 id=F([args1])>
                <tag2 id=G([args2])> …</>
              </>
```

Rule 2: an element that has an atomic content must have a Skolem function with $X $\in$ args

```
CONSTRUCT …  <tag id=F([args])> $X </> …
```

Contructing trees with Skolem fcns (con't)

Rule 3: if a Skolem function occurs in two different places then the following must hold:

- G = H
- args1 = args2
- tag1 = tag2

```
{ CONSTRUCT <tag1 id=G([args1])> <tag id=F([args])> …</> </>  }
{ CONSTRUCT <tag2 id=H([args2])>  <tag id=F([args])> …</> </> }
```

XML-QL v.s. XQuery

- Xquery (=Quilt) v.s. XML-QL
  - + faithful XML data model
  - + Xpath sublanguage
  - + aggregate functions (like in SQL)
  - + some features from XQL
  - - Patterns
  - - Skolem functions