XPath
Query Language for XML

csc343, Introduction to Databases
Diane Horton
with material from Ryan Johnson, Manos Papagelis, Jeff Ullman, Ramona Truta,
and Renée Miller
Winter 2017

UNIVERSITY OF
TORONTO

Data Model

- We saw that an XML file has a tree structure.
- White space in the file is represented in the tree.

  So these two files have different document trees:

  ```
  <?xml  version=“1.0” ?><Things><Thing>bucket</Thing>\n
  <Thing>mop</Thing></Things>
  ```

  ```
  <?xml  version=“1.0” ?>\n
  <Things>\n
  \t<Thing>bucket</Thing>\n
  \t<Thing>mop</Thing>\n
  </Things>\n
  ```

- How they look in a file and a document tree . . .
- How xmllint shows the contents to you . . .

XPath Query Language

Path expressions

- Goal of a query is to find items you want from a document.
- It does this by describing a path(s) through the document tree.
- The query takes the form of a path expression.
- Analogy in unix:
  - File system is a tree with files as leaves and directories as internal nodes.
  - `ls /course*/assignments/a1/solution/*.py`

Example

```
<?xml  version=“1.0” ?>
<Students>
  <Student StudId=“111111111” >
     <Name><First>John</First><Last>Doe</Last></Name>
     <Status>U2</Status>
     <CrsTaken CrsCode=“CS308” Semester=“F1997” />
     <CrsTaken CrsCode=“MAT123” Semester=“F1997” />
  </Student>
  <Student StudId=“987654321” >
     <Name><First>Bart</First><Last>Simpson</Last></Name>
     <Status>U4</Status>
     <CrsTaken CrsCode=“CS308” Semester=“F1994” />
  </Student>
</Students>
```

- To find all course codes, we use this path:
  root → Student → CrsTaken → CrsCode attribute

Writing and Running an XPath query

- Create a file containing:
  `fn:doc(“`*«xml file»*`”)`*«path expression»*
  - `fn:doc` is a function that parses the document an evaluates to a document tree.
- Suppose `query.xq` contains:
  `fn:doc(“`*`courses.xml`*`”)/Student/CrsTaken/@CrsCode`
  - Each slash takes us down one level in the tree.
  - `@` takes us to an attribute; otherwise we go to a sub-element.
- To run it on cdf:
  `galax-run query.xq`

Result of a path expression

- The result of a path expression is a sequence of items from the document.
- Each item is either
  - a primitive value, such as a string or integer
  - or a node in the document.

Homogeneous or heterogeneous results

- Often, queries yield homogeneous results.
  Examples:
  ```
  doc("quiz.xml")//questions/mc-question
  doc("quiz.xml")//tf-question/@solution
  ```
- But some queries don't.
  Example:
  ```
  doc("quiz.xml")/quiz/questions/*/*
  ```
  Yields a mix of question elements and option elements.

XPath documentation

- Official Xpath documentation:
  http://www.w3.org/TR/xpath20/
- Functions and operators (very useful!):
  http://www.w3.org/TR/xpath-functions/
- Manual (available on cdf):
  /usr/share/doc/galax-doc/manual/manual.html
  (Relevant if installing galax on your own machine.)

Other axes

Axes

- So far, we've navigated the tree by going from parent to child node.
- There are many more modes of navigation, called axes.
- Here, axes is the plural of axis, not axe!

Syntax for axes

- Notation:
  `/`*«axis»*`::`
  where *axis* is one of
  - `child`
  - `parent`
  - `attribute` (we'll see more axes later)
- If you do not specify an axis, the default is used: `child`
- So the path expression

  ```
  fn:doc("courses.xml")/Students
  ```

  is shorthand for

@ is shorthand for the attribute axis

- So this path expression

  ```
  fn:doc(“courses.xml”)
     /Students
     /Student
     /CrsTaken
     /@CrsCode
  ```

  is short for

  ```
  fn:doc(“courses.xml”)
     /child::Students
     /child::Student
     /child::CrsTaken
     /attribute::CrsCode
  ```

Attribute axis in a condition

- This path expression

```
fn:doc(“courses.xml”)
   /Students
   /Student
   /CrsTaken[@CrsCode=”cs308”]
```

is short for

```
fn:doc(“courses.xml”)
   /child::Students
   /child::Student
   /
child::CrsTaken[attribute::CrsCode=”cs308”]
```

Other shorthand for axes

- // is shorthand for the descendant-or-self axis, so this

```
fn:doc("courses.xml")
   //CrsTaken
```

is short for

```
fn:doc("courses.xml")
   /descendant-or-self::CrsTaken
```

- Dot (.) is shorthand for the self axis, so this

```
fn:doc("courses.xml")
   //CrsTaken/@CrsCode[.="cs308"]
```

is short for

```
fn:doc("courses.xml")
   /descendant-or-self::CrsTaken
```

And there are even more axes

- Other axes include:
  - `parent`
  - `ancestor`
  - `ancestor-or-self`
  - `following-sibling`
  - `preceding-sibling`
- See section 2.2 of the documentation for more: `http://www.w3.org/TR/xpath/#axes`