An Introduction to XPath Language

Lecturer: Pavle Mogin

COMP 442
Issues in Databases and Information Systems

Plan For XPath Language

- What is XPath?
- What is XPath used for
- Basic XPath syntax
- Abbreviated syntax

- Readings:
  - XML Path Language (XPath) Version 1.0
    http://www.w3.org/TR/1999/REC-xpath-19991116
**What is XPath**

- XPath is a language for specifying navigation within an XML document
- It also provides basic facilities for manipulating strings, numbers, and booleans
- XPath gets its name from its use of path notation as in URLs for navigating through the hierarchical structure of an XML document
- XPath models an XML document as a tree of nodes
- Most common nodes are: elements, attributes, text
- XPath defines a way to compute a string value for each type of node

**What is XPath used for**

- XPath is used to define paths in an XML document in:
  - XSL Transformations (XSLT),
  - XPointer,
  - XQuery Language,
  - Within XML Schema for specifying special constraints like: key, referential integrity, and unique
- A restricted subset of XPath is used in XQuery and for specifying context of XML Schema constraints
- We are almost exclusively interested in using XPath within XQuery and for declaring constraints
**XPath Location Paths**

- Navigation through an XML document is declared using XPath expressions
- Although Location Paths are not the most general grammatical construct in the language, they are most important construct and we shall pay our exclusive attention to them
- Location paths can be expressed using either an unabbreviated or an abbreviated syntax
- There are two kinds of location paths:
  - Relative location paths and
  - Absolute location paths

**Unabbreviated Syntax of Location Paths**

- A relative location path has the following syntax:
  
  \[
  \text{Path} ::= \text{Step}_1/\text{Step}_2/\ldots/\text{Step}_n
  \]

  where each \text{Step} is a triple (\text{Axis}, \text{Node-test}, \text{Predicate}) and is defined as follows:

  \[
  \text{Step} ::= \text{Axis}:: \text{Node-test} \text{Predicate}^* 
  \]
  - The axis specifies the direction to move in the document tree
  - The node test selects nodes along the specified axis, and
  - The predicates (if any) filter the nodes selected

- Separators “/” between two subsequent steps indicate a direct parent-child relationship between nodes involved in the steps
Evaluation of a Relative Location Path

- A relative location path is evaluated step by step, from left to right
- A step is applied to a single node, so called context node
- The application of a step on a context node selects a set of result nodes
- Each node of a result set is then used as a context node in the following step
- The result of an expression is the union of node-sets selected by the last step
- Suppose a location path is declared (in an XML schema) within an element \( E_c \) declaration, then the Step_1 will be applied on the node-set of the type \( E_c \) (each node of the type \( E_c \) will be used as the first context node)

Result of a Relative Location Path

- The result of an expression is the union of node-sets selected by the last step
- Elements of a node-set are nodes of different types:
  - Elements, Attributes, Text, Comments, Processing Instructions,…
- According to the Standard:
  - A string value of a node is the concatenation of the string values of all text descendants of the node
- We shall obey to that definition for all node types except for complex element nodes, where we shall use the node identifier instead
Absolute Location Path

• An absolute location path consists of a / optionally followed by a relative location path
• A leading “/” itself selects the root node of the document as the context node
• Formally:
  – LocationPath ::= RelativeLocationPath | AbsoluteLocationPath
  – AbsoluteLocationPath ::= ‘/’RelativeLocationPath
  – RelativeLocationPath ::= Step | RelativeLocationPath’/’Step
• Both relative and absolute location path may be expressed either using an unabbreviated or abbreviated syntax

A Class XML Document

```xml
<?xml version="1.0" standalone="yes"?>
<CLASS name="COMP442">
  <LECTURER>Pavle</LECTURER>
  <STUDENT no_of_pts="20.0" origin="International">
    <NAME>Ahmed</NAME>
    <ASSIGNMENT no="1">10.0</ASSIGNMENT>
    <ASSIGNMENT no="2">10.0</ASSIGNMENT>
  </STUDENT>
  <STUDENT no_of_pts="0.0" origin="Domestic">
    <NAME>Bad Student</NAME>
  </STUDENT>
</CLASS>
```
**A Class Document Tree**

![Diagram of a class document tree]

**Axis of an XPath Step**

- Within an XPath step, Axis specifies “direction” in which to navigate through a document
  - For example, the step:
    ```
    child::STUDENT
    where Axis = child:: and Node-test = STUDENT
    would select all child nodes (of a context node) that have the name STUDENT
    ```
- The XPath supports 12 different axes for navigation
- The **child::** axis is most commonly used
- Some of the others are:
  - **attribute::** (selects attributes of a context node),
  - **descendant::** (selects descendant nodes of a context node),
  - **descendant-or-self::**
  - **parent::** (selects the parent node of a context node),
  - **self::** (selects the context node itself),
The Node-test of an XPath Step

- A Node-test specifies a simple test on XML nodes found along the steps’ axis
- The nodes that pass that test are candidates for the next step
- The most commonly used Node-test is element name
  - In the step child::STUDENT, STUDENT is an element name
- Other often used node tests are ‘*’, which evaluates to true for all element nodes, and attribute test
  - A step child::* will choose all subelements of a context node
  - A step attribute:: name will select the name attribute of a context node
- There are also Node-tests for:
  - Namespaces,
  - Text nodes,
  - Comments, and
  - Processing instructions

Predicates of a Step

- An XPath step can also include a sequence of predicates
- The predicates are applied to the nodes selected by Node-test
- Only nodes that evaluate to true for all predicates will belong to the result nodeset of the step
- A predicate compares a node property with a value using operators from the set {=, ≤, ≥, <, >, !≡,}
- A node property can be:
  - The value of an attribute,
  - The value of PCDATA of an element, or
  - The sibling order value of a node (returned by the function position())
Examples of XPath Predicates

- `child::STUDENT[position()=2]` selects the second `STUDENT` child element of the context node.
- `child::STUDENT/attribute::[origin="International"] [no_of_pts>"35"]` selects all student children of the context node that are international and have more than 35 points.
- `child::STUDENT/child::NAME="Ahmed"` selects the `STUDENT` children of the context node that have a `NAME` child with a string value equal to Ahmed.

Abbreviated Syntax of Location Path (1)

- The most important abbreviation is that `child::` axis can be omitted from a location step.
- In effect, `child::` is the default axis.
- For example,
  - `STUDENT/NAME` is a short for `child::STUDENT/child::NAME`.
- There is also an abbreviation for attributes: `attribute::` can be abbreviated to `@`.
- For example,
  - `STUDENT[@origin="Domestic"]` is short for `child::STUDENT[attribute::origin="Domestic"]` and will select all `STUDENT` children of the context node whose origin is Domestic.
Abbreviated Syntax of Location Path (2)

- If a predicate expression evaluates to an integer value that value is considered to be the position of the node selected
  - For example, \texttt{STUDENT[2]} step would select the second \texttt{STUDENT} child of the context node
- An empty step ‘//’ is also a frequently used abbreviation, it specifies that the element that follows may be nested anywhere within the document
- For example,
  - ‘//STUDENT’ would select all student nodes anywhere within the document

Abbreviated Syntax of Location Path (3)

- A location step of ‘.’ is short for \texttt{self::node()}, where \texttt{self::} refers to the context node and \texttt{node()} returns nodes of any type
- For example,
  - ‘./STUDENT’ is short for \texttt{self::node()//descendant-or-
    self::node()//child::STUDENT}
    and will select all \texttt{STUDENT} elements that are children of the context node itself or any of its descendants
- A location step of ‘.’ is short for \texttt{parent::node()}
- For example,
  - ‘../LECTURER’ is short for \texttt{parent::node()//child::LECTURER}
    and will select all \texttt{LECTURER} children of the parent of the context node
Declaring Location Path in XML Schema

- A location path is declared in an XML schema using the attribute `path` within the definition of an element.
- Let a location path `p` be declared within an element `E`, then the element `E` is the context of the path `p`.
- The selection of the context element has to provide for an unambiguous path evaluation.
- Next we are going to consider a couple of example location path declarations using abbreviated syntax.

Location Path Examples (1)

- Example 1:
  ```xml
  <xsd:element name="CLASS" ...>
    ...
    <!-- ...xpath="STUDENT/NAME"...> -->
    ...
  </xsd:element>
  - Result = {Ahmed, Bad Student}
- Example 2
  ```xml
  <xsd:element name="CLASS" ...>
    ...
    <!-- ......xpath="STUDENT/ASSIGNMENT[@no="1"]"...> -->
    ...
  </xsd:element>
  - Result = {10}
  ```
Location Path Examples (2)

- Example 3:
  
  ```xml
  <xsd:element name="STUDENT" ...>
    ...
    <......xpath="//LECTURER"...>
    ...
  </xsd:element>
  - Result = {Pavle}
  
- Example 4
  
  ```xml
  <xsd:element name="CLASS" ...>
    ...
    <......xpath="STUDENT[ASSIGNMENT="10"] [@no="2"]/NAME"...>
    ...
  </xsd:element>
  - Result = {Ahmed}
  ```

Location Path Examples (3)

- Example 5:
  
  ```xml
  <xsd:element name="STUDENT" ...>
    ...
    <......xpath="."...>
    ...
  </xsd:element>
  - Result = {&1}
  
- Example 6
  
  ```xml
  <xsd:element name="CLASS" ...>
    ...
    <......xpath="STUDENT"...>
    ...
  </xsd:element>
  - Result = {&3, &7}
Summary (1)

- XPath is a language for specifying navigation within an XML document
- XPath models an XML document as a tree of nodes
- A restricted subset of XPath called LocationPath is used in XQuery and for specifying context of XML Schema constraints
- A location path has the following syntax:

  \[ \text{Path} ::= \text{Step}_1/\text{Step}_2/\ldots/\text{Step}_n \]

  where each \text{Step} is a triple (\text{Axis}, \text{Node-test}, \text{Predicate}):
  - The axis specifies the direction to move in the document tree
  - The node test selects nodes along the specified axis, and
  - The predicates (if any) filter the nodes selected
- An abbreviated syntax is used to declare navigation through an XML document in XQuery and when defining XML constraints

Summary (2)

- A location path can be either:
  - Relative, or
  - Absolute
- A relative location path is declared with regard to a context node and its evaluation stars from this node
- A relative location path is declared in an XML schema using the attribute \text{path} within the declaration of an element
- This element is the context of the relative location path declared by \text{path}