

INFS 740
DB Programming for the WEB

Professor Alex Brodsky
GMU

XML Pattern Matching with XPath

- XML – Tree structure
- Much like a file directory
- Task:
 - Starting from a node
 - Selection of (other) nodes in the tree
- Use Path Expressions
 - “XPath”
- According to W3C, “XPath is a language for addressing parts of an XML document”

XPath Details

- Location Steps
 - “Axis” specifier
 - Node test
 - Predicates
- Location Paths
 - Optional “/”
 - Followed by zero or more location steps, separated by “/”

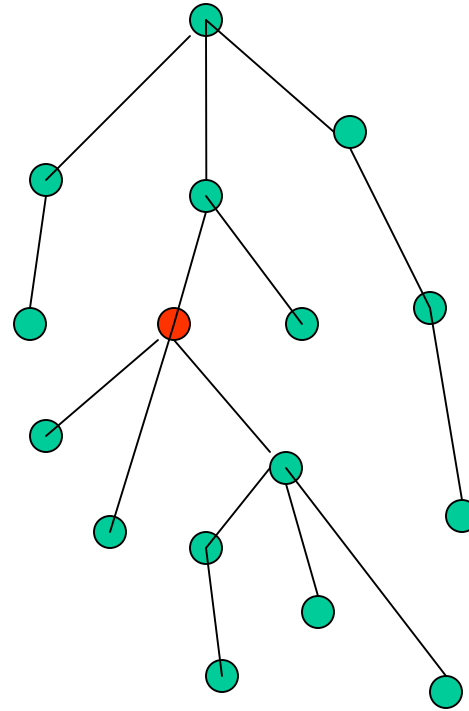
Axis

- Ancestor
- Ancestor-or-self
- Attribute
- Child
- Descendant
- Descendant-or-self
- Following
- Following-sibling
- Namespace
- Parent
- Preceding
- Preceding-sibling
- Self

Three types of nodes in the tree:
Elements, Attributes, Namespace

Example

- Ancestor
- Ancestor-or-self
- Attribute
- Child
- Descendant
- Descendant-or-self
- Following
- Following-sibling
- Namespace
- Parent
- Preceding
- Preceding-sibling
- Self



Axis + Node Test

- **Self::*** select self
- **Self::AAA** select self if it's AAA
- **Child::AAA** select all AAA children
- **Ancestor:: AAA** select all AAA ancestors
- **Self::node()** select self if it's a node
- **Self::comment()** select self if it's comment
- **Attribute::AAA** select Attributes AAA

Axis + Node Test + Predicates

- `Child::AAA[position()=1]`
 - Select first AAA child
- `Child::AAA[position()=last()]`
 - Select the last AAA child
- `Self::*[node()]`
 - Select self if it's a node (same as `self::node()`)
- `Self::*[count(Ancestor::*>2)]`
 - Select self if it's 3 or more levels down from the root.

Union

- Connected with |
- Example:
 - Ancestor::<* | Self::<*

Example Location Paths

- Location Path
 - Optional “/” zero or more location steps separated by “/”
- Examples
 - child::AAA/child::BBB
 - ancestor::*[count(ancestor::*)<1]
 - Find the root!!
 - /child::AAA[position()=1]
 - The first AAA child of the root

Abbreviations for Axes

- (nothing) child::
- / document root
- @ attribute::
- // /descendant-or-self::node()/
- . Self::node()
- .. Parent::node()
- Examples:
 - AAA child::AAA
 - /AAA start from the root: child::AAA
 - //AAA all AAA descendants of root
 - //AAA/./../ all AAA descendants of root

Node-Sets etc

- Node sets:
 - | (union)
 - [expr]
- Functions:
 - Last(), position(), count(), starts-with(), etc etc.
- Boolean:
 - And, or, >=, <=, etc.

One More Example

- Find the AAA element(s) which must be the 2nd child of its parent **and** its ancestors do not include an element with attribute B.

```
//AAA[count(preceding-sibling::*)=1 and  
count(ancestors::*[@B])=0]
```