XML databases

Jan Chomicki
University at Buffalo

Outline

XML documents (simplified)

XML tree

- finite, ordered, unranked tree
- element, attribute and text nodes
- element and attribute node labels from a finite label alphabet Σ
- attribute and text string values
- only element nodes have children
- document order (left-to-right prefix order)

XML trees represent well-formed documents:

- matching, properly nested opening and closing tags
- single root element

Regular expressions over Σ

$$E \ := \ \varepsilon \mid a \mid E \cup E \mid E \ E \mid E^* \quad \text{where } a \in \Sigma.$$

Defining valid XML documents

XML schema definitions

- Document Type Definitions (DTDs)
- XML Schema
- automata-based approaches

DTD (over Σ)

- element-only content: a function mapping node labels from Σ to a regular expression to which the concatenated children of the node must conform
- element only content should be deterministic: during pattern matching only one symbol can be chosen
- also text-only (#PCDATA), mixed, empty, and unrestricted (ANY) content
- attributes: text-valued (CDATA), enumerations, ID, IDREF
- attributes can be required (#REQUIRED) or optional (#IMPLIED)

XPath

Data model

- tree-based
- nodes: document root, element, attribute, text,...
- root element is a child of document root
- 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: (current) node, position, size
- absolute (starting at document root) or relative
- consists of steps separated by /
- wildcards
- union (|), intersection, difference

XPath axes

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]`

XQuery

Features

- functional
- compositional: expressions can be nested arbitrarily
- recursion
- declarative: influenced by SQL

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 as text

- hard to query and manipulate

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