Bachelor's Thesis

XPath 2.0 Full-Text Query Rewriting

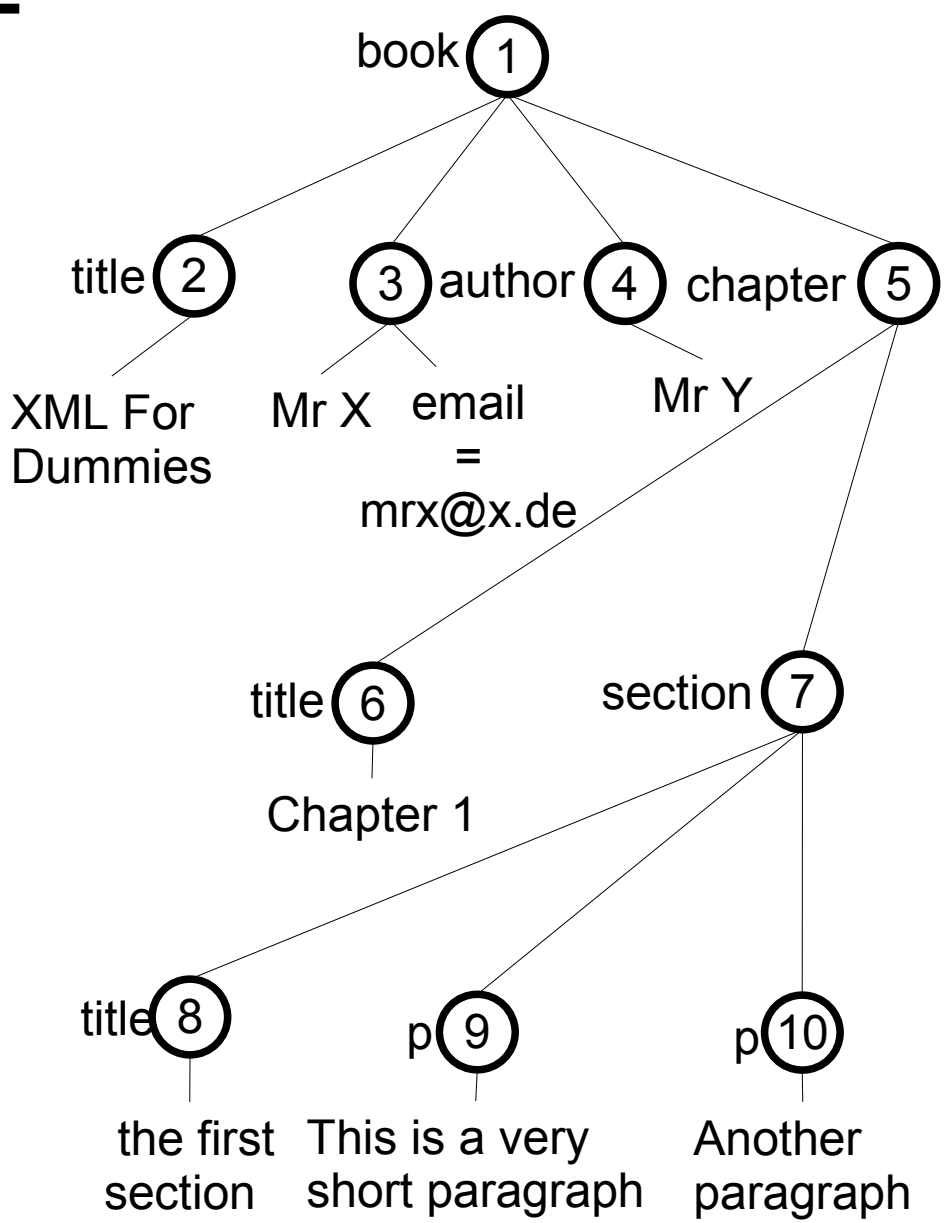
Marco Stadtmüller

February 27, 2007

Contents

XML

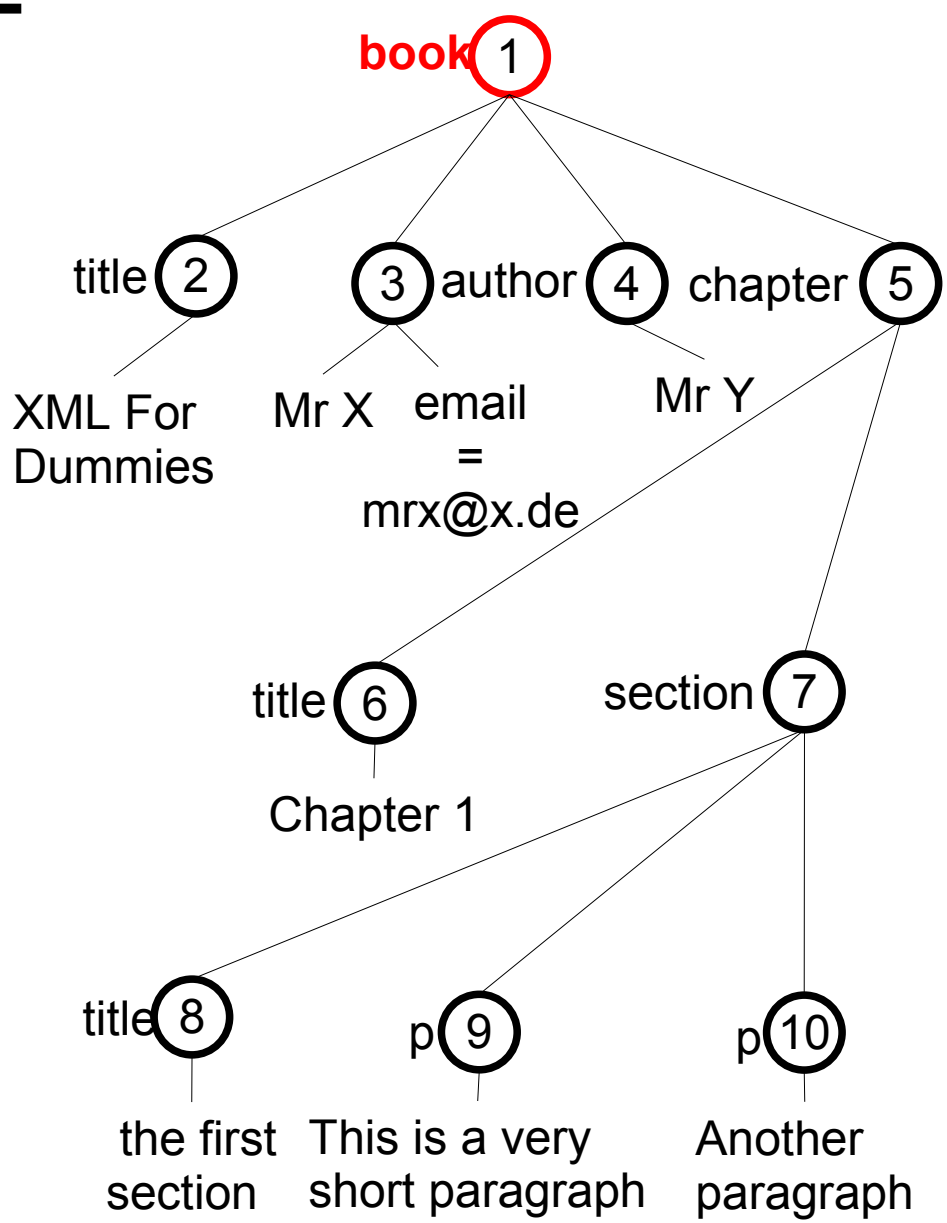
```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author email="mrx@x.de">Mr X</author>
   <author>Mr Y</author>
   <chapter>
      <title>Chapter 1</title>
      <section>
         <title>the first section</title>
         <p>this is a very short paragraph</p>
         <p>another paragraph</p>
      </section>
   </chapter>
   ...
</book>
```



XML

element node

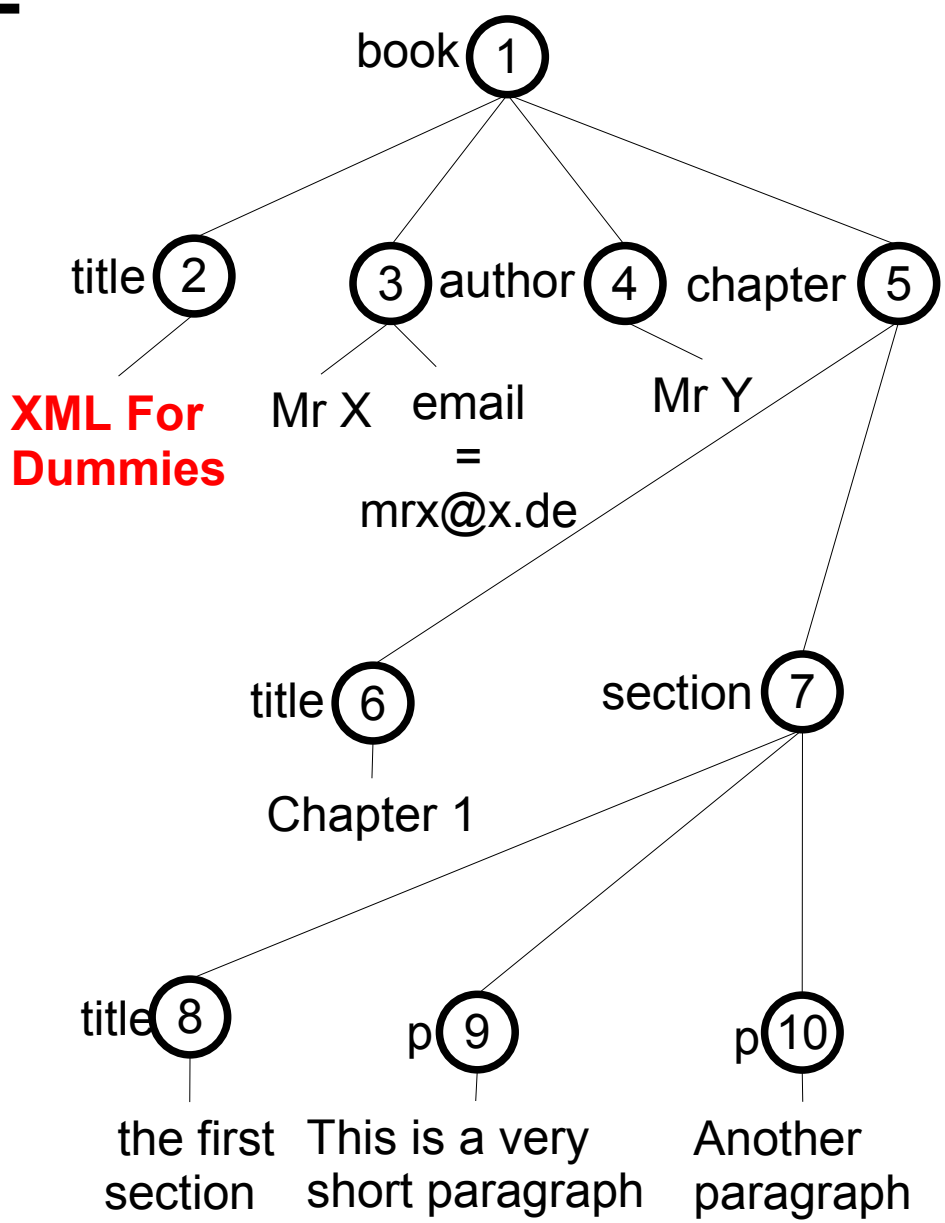
```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author email="mrx@x.de">Mr X</author>
   <author>Mr Y</author>
   <chapter>
      <title>Chapter 1</title>
      <section>
         <title>the first section</title>
         <p>this is a very short paragraph</p>
         <p>another paragraph</p>
      </section>
   </chapter>
   ...
</book>
```



XML

text node

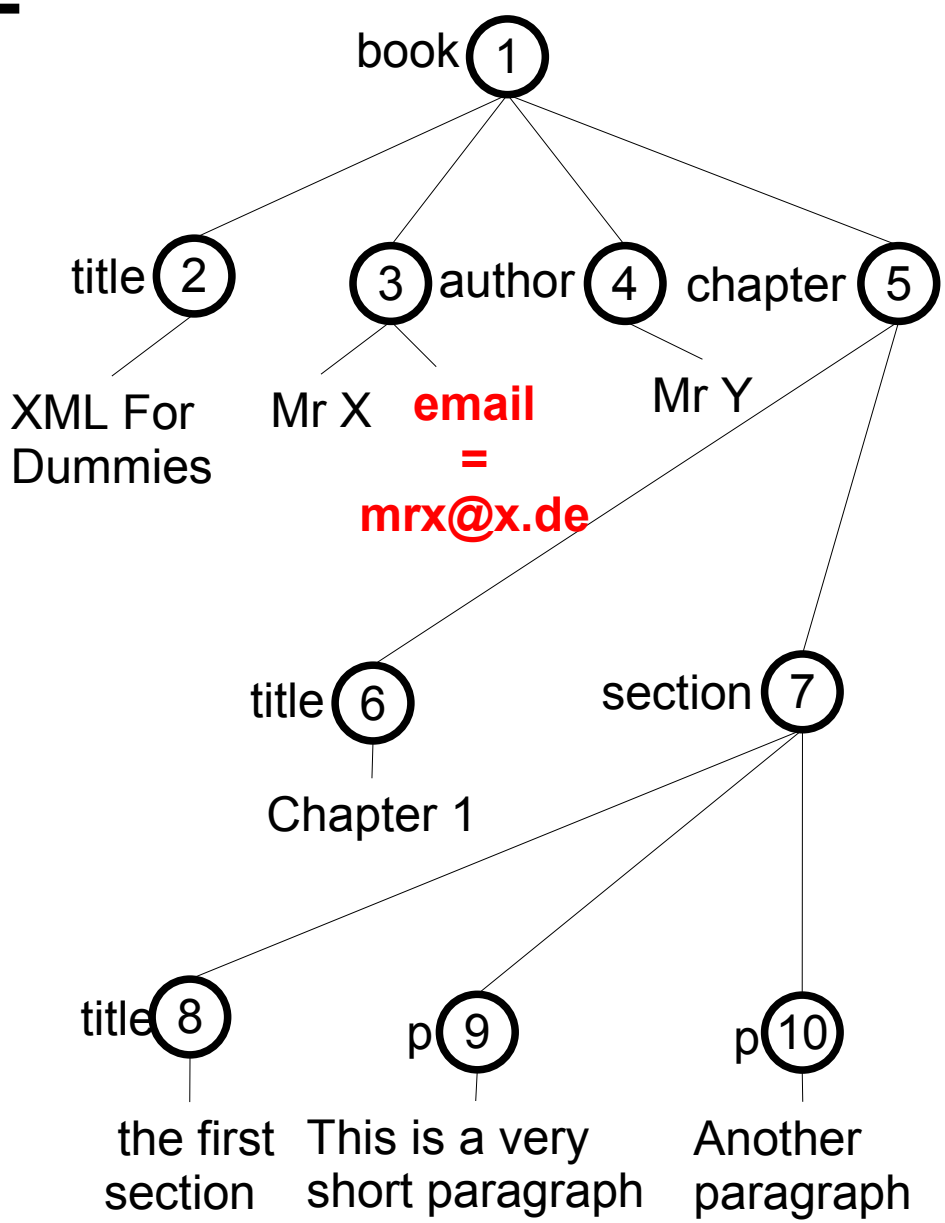
```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author email="mrx@x.de">Mr X</author>
   <author>Mr Y</author>
   <chapter>
      <title>Chapter 1</title>
      <section>
         <title>the first section</title>
         <p>this is a very short paragraph</p>
         <p>another paragraph</p>
      </section>
   </chapter>
   ...
</book>
```



XML

attribute node

```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author email="mrx@x.de">Mr X</author>
   <author>Mr Y</author>
   <chapter>
      <title>Chapter 1</title>
      <section>
         <title>the first section</title>
         <p>this is a very short paragraph</p>
         <p>another paragraph</p>
      </section>
   </chapter>
   ...
</book>
```



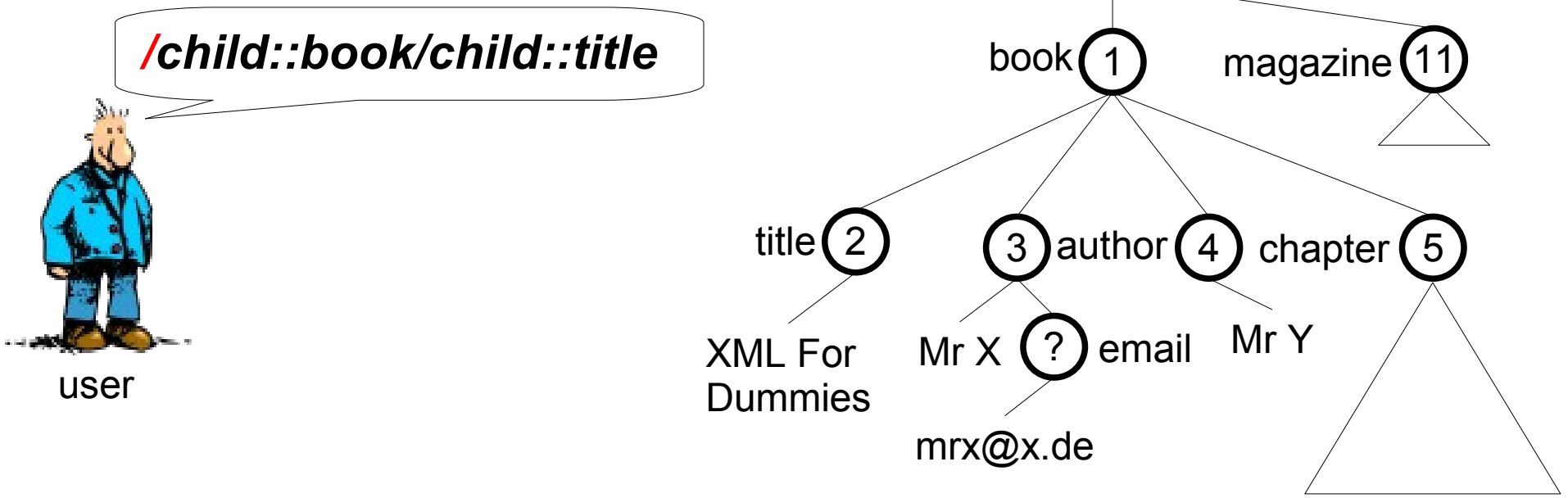
XPath

- addresses nodes of a XML-document
- returns a set of nodes

- **Basics**:
- Axes
- Filter
- Arithmetic Expressions

XPath - Axes

- child-axis

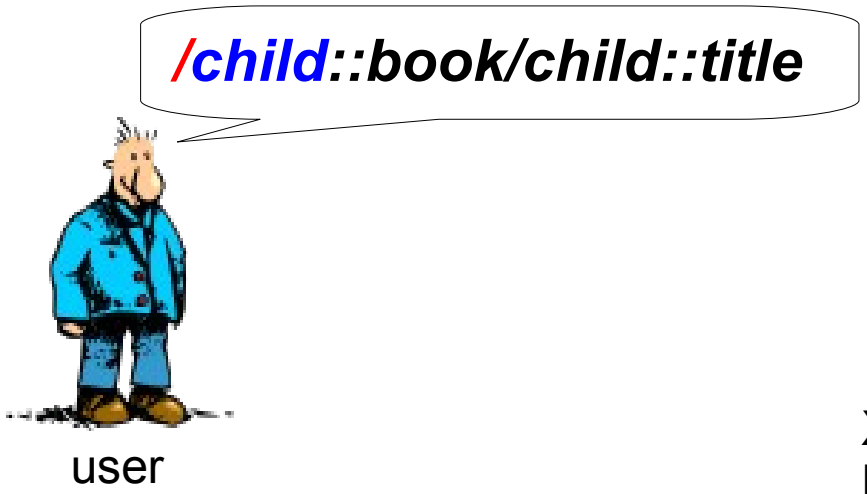


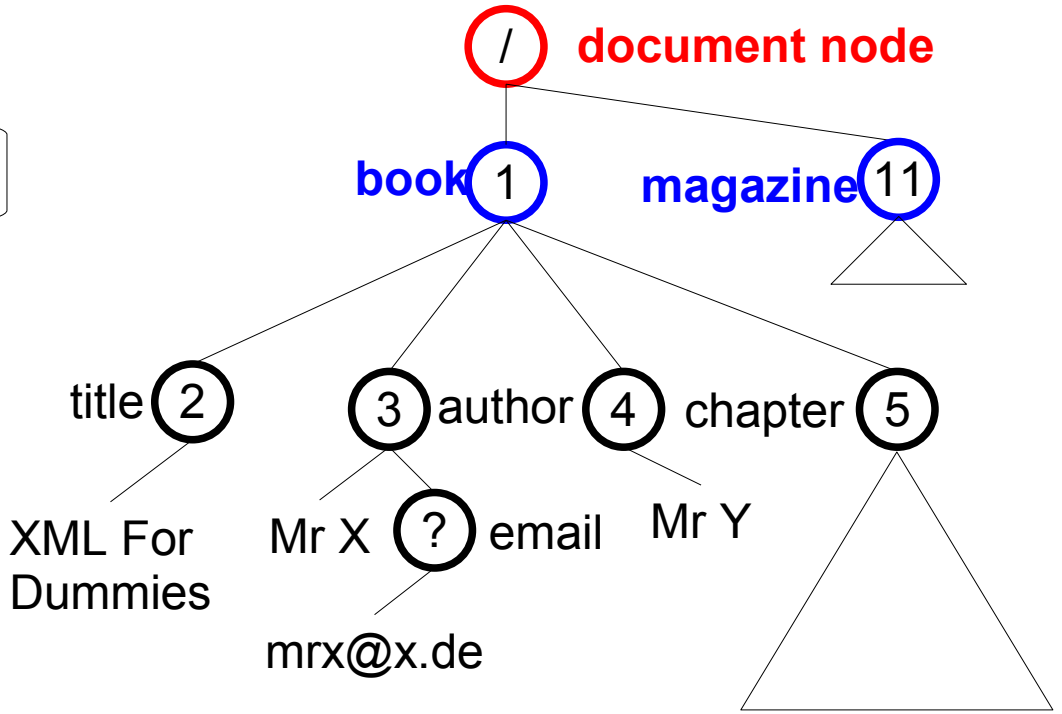
short form

/book/title

XPath - Axes

- child-axis



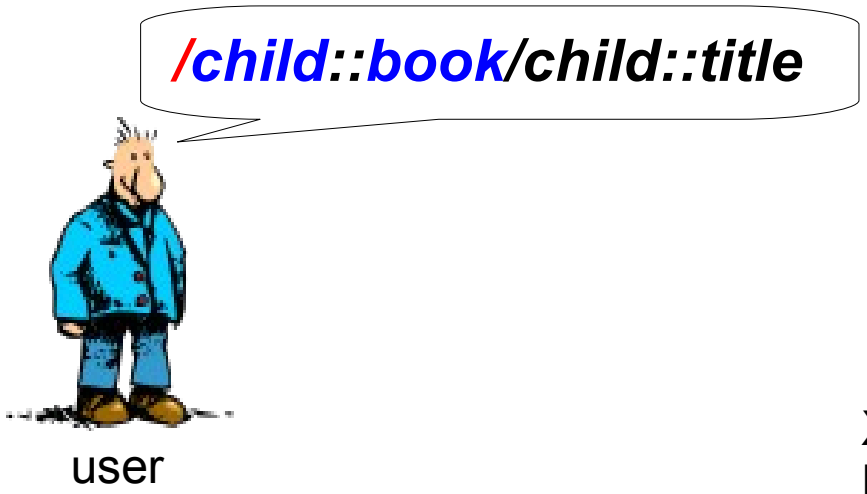


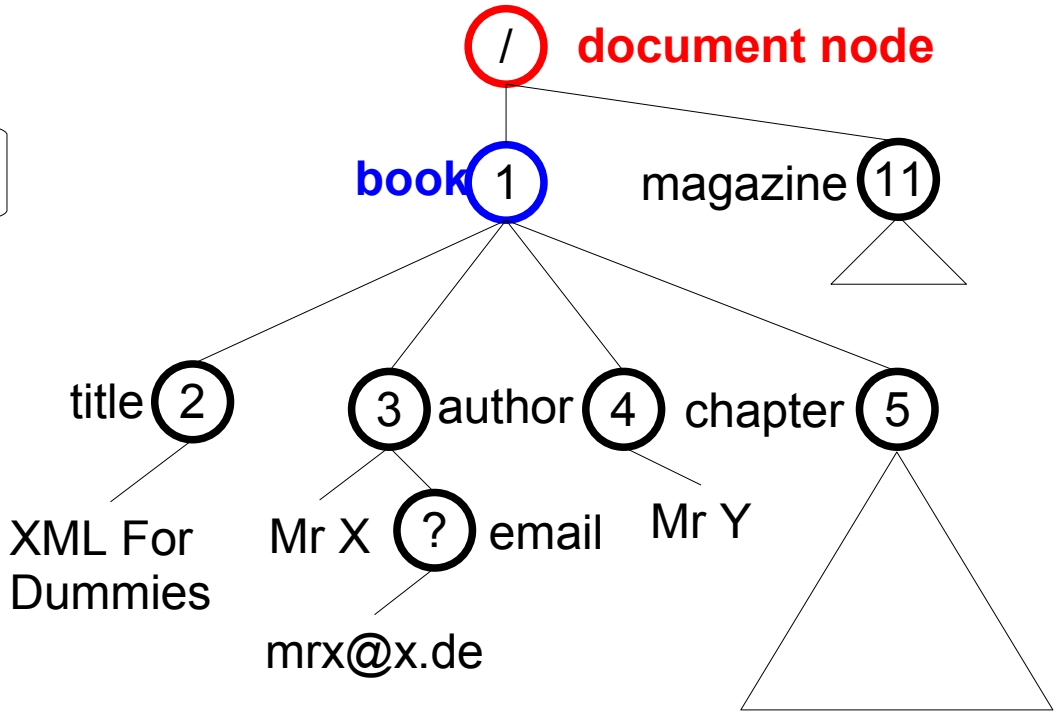
short form

/book/title

XPath - Axes

- child-axis



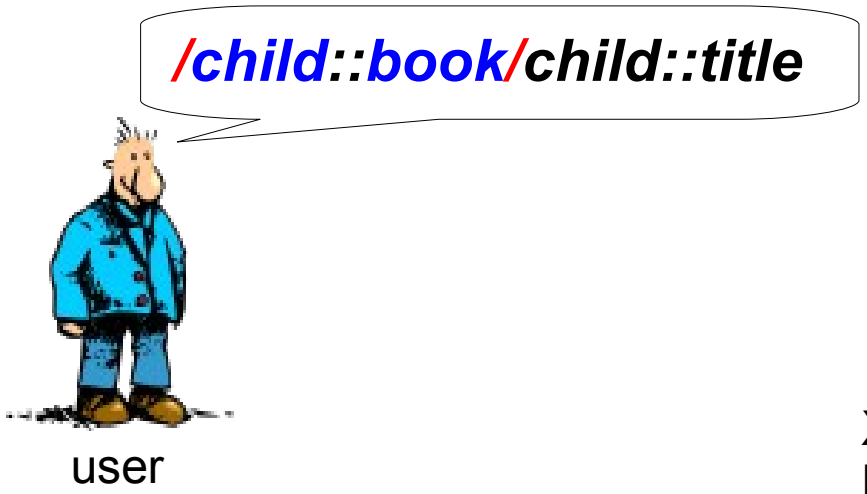


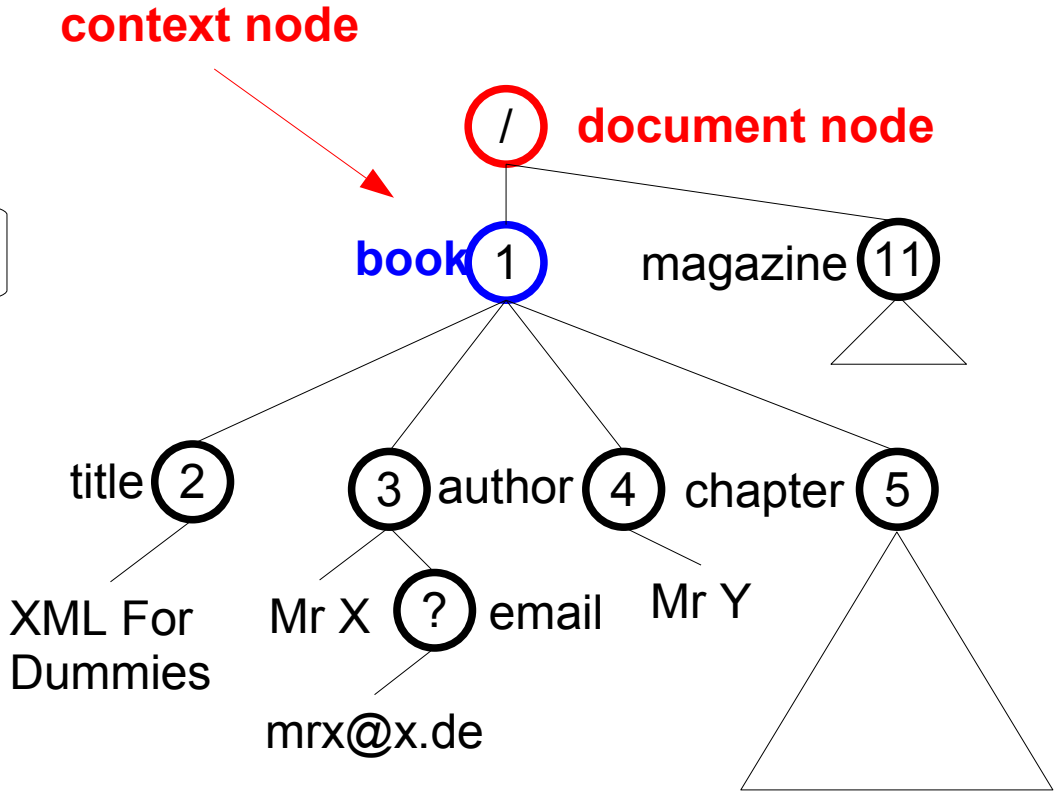
short form

/book/title

XPath - Axes

- child-axis





short form

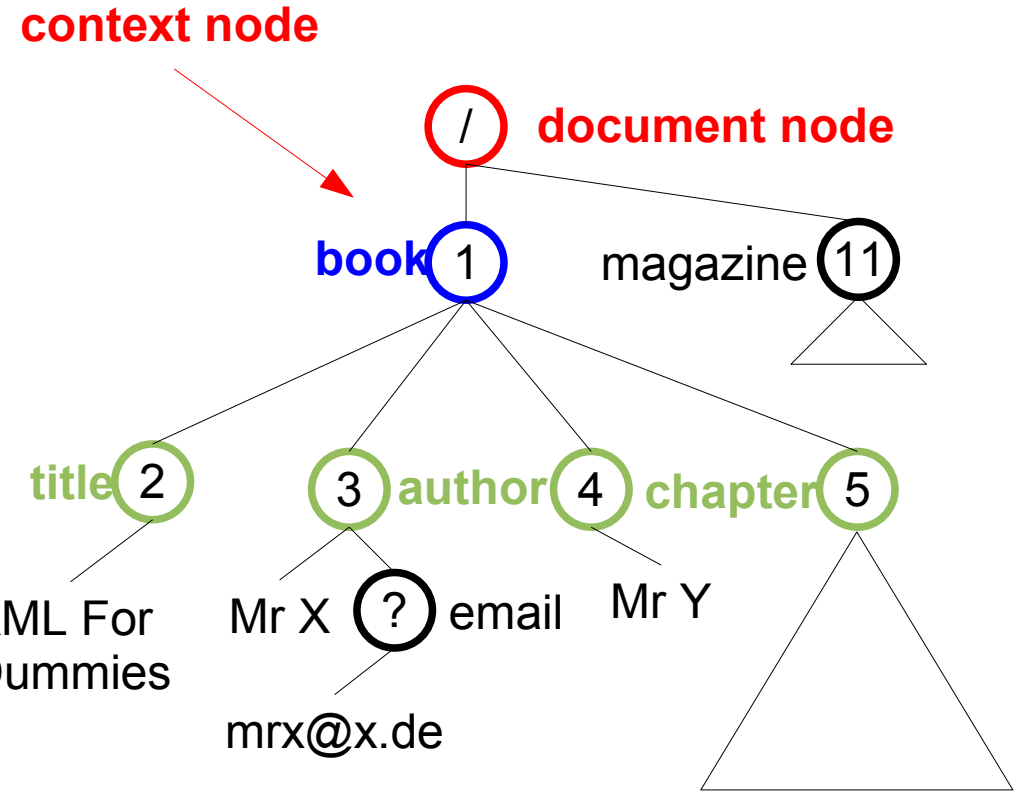
/book/title

XPath - Axes

- child-axis

/*child*::*book*/*child*::*title*

user

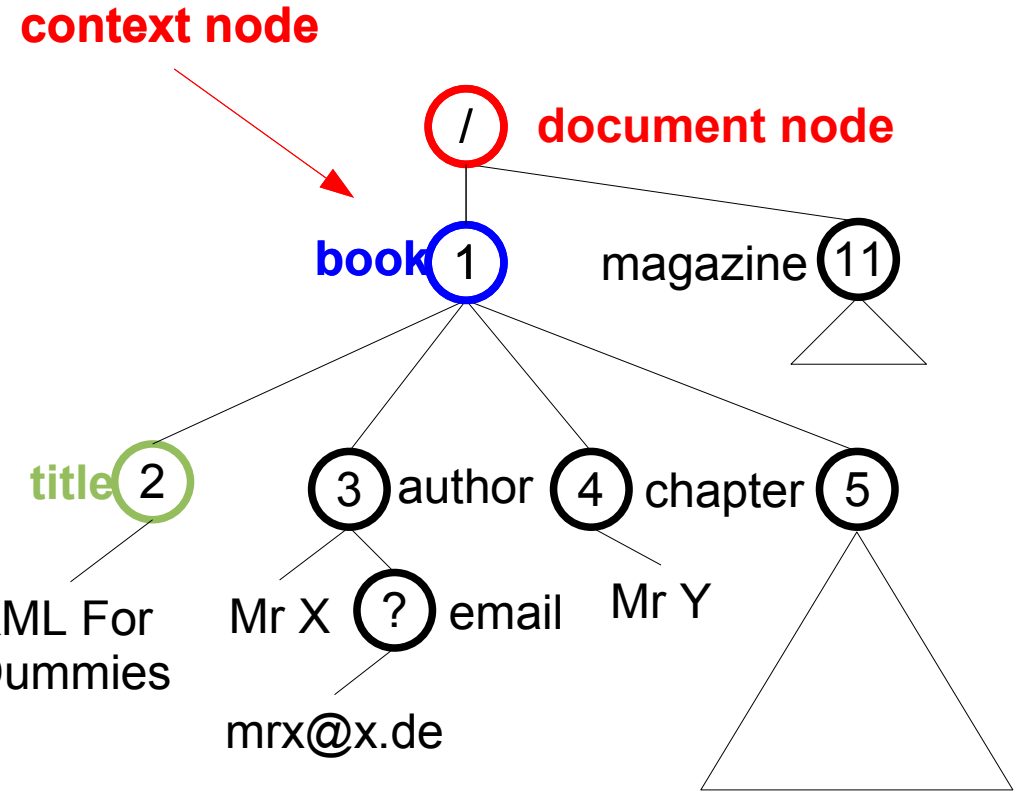


short form

/book/title

XPath - Axes

- child-axis



short form

/book/title

XPath - Axes

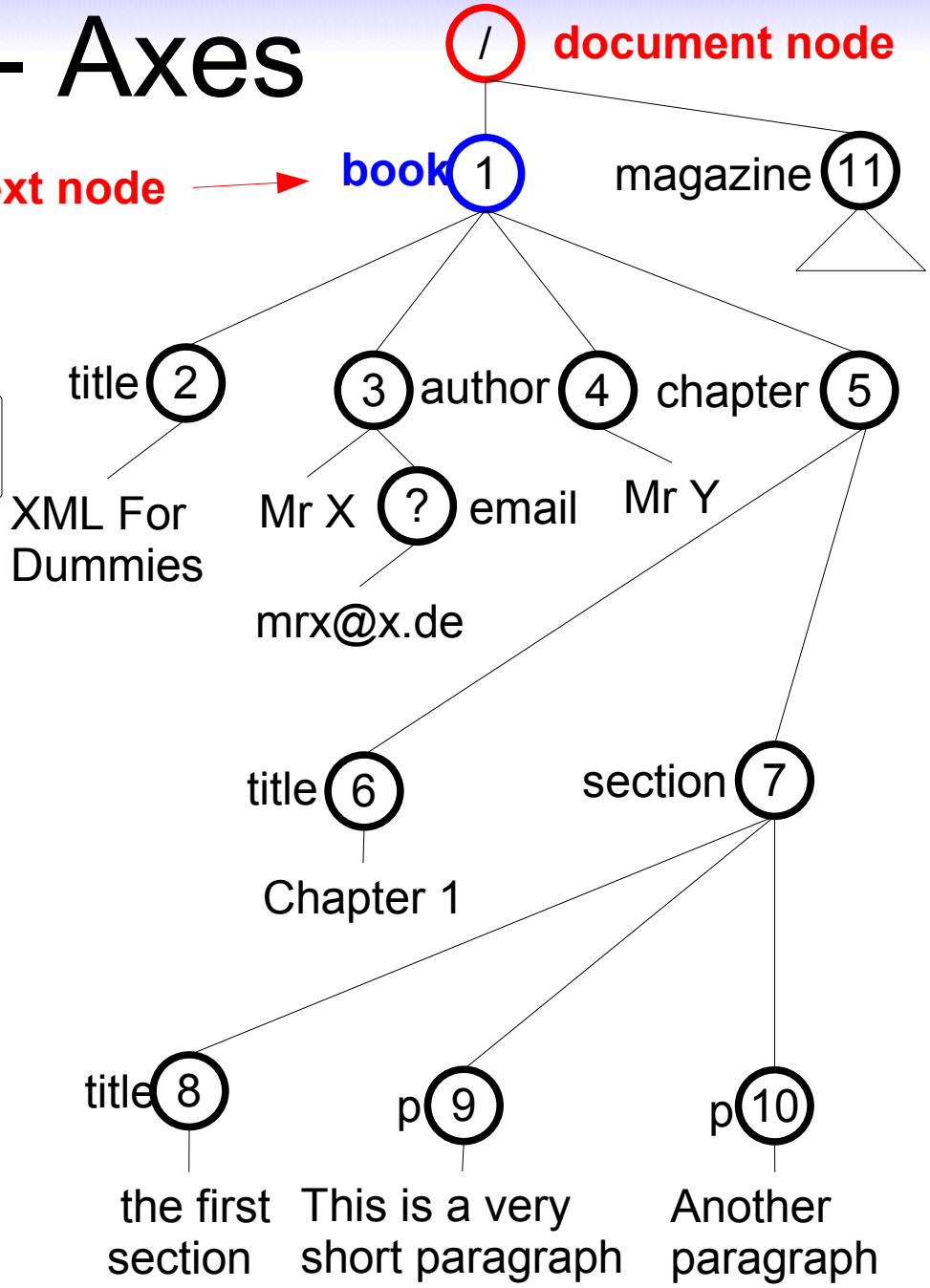
- descendant-axis

/child::book/descendant-or-self::title

user

short form

/book//title



XPath - Axes

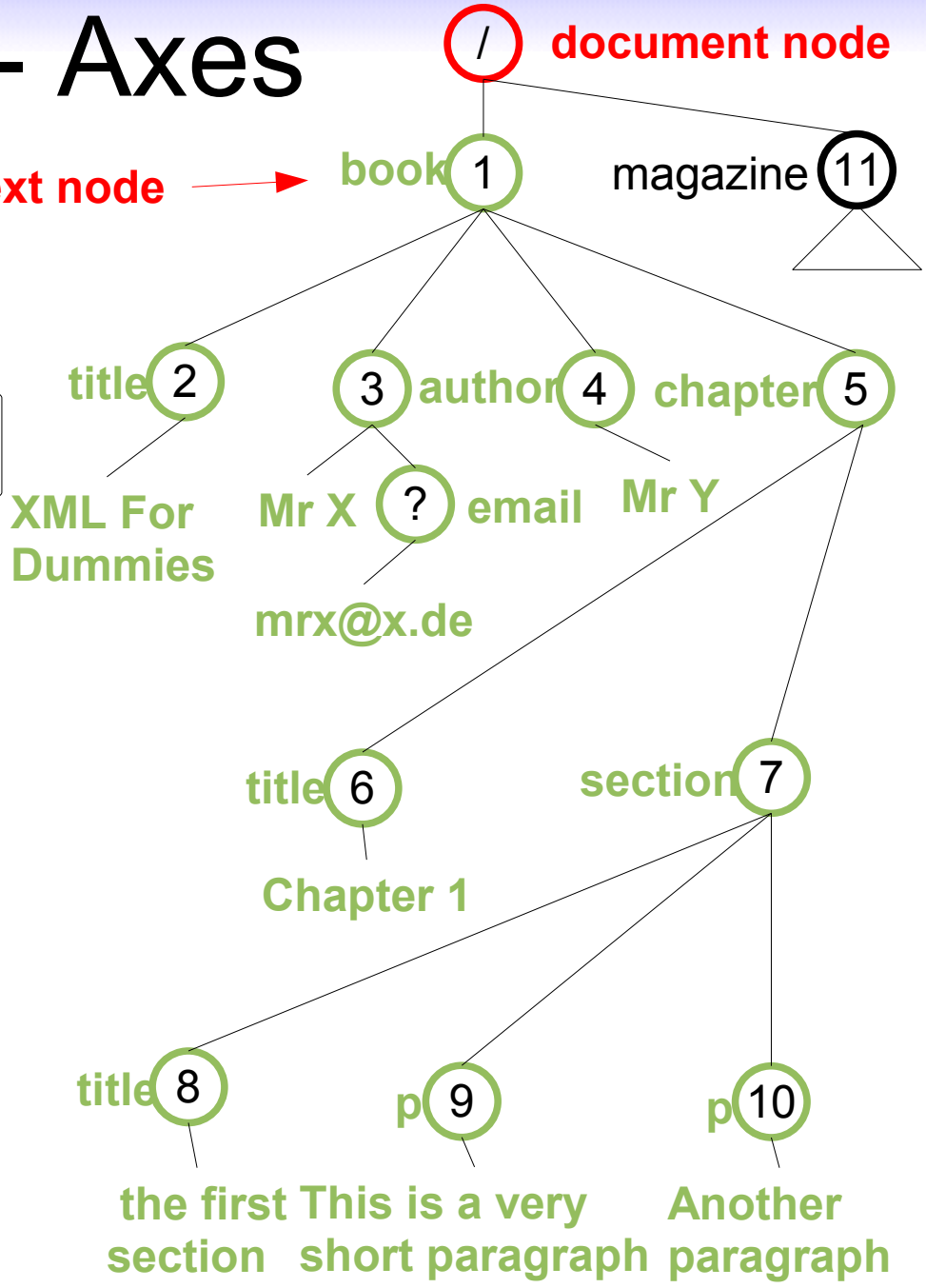
- descendant-axis

/child::book/descendant-or-self::title

user

short form

/book//title



XPath - Axes

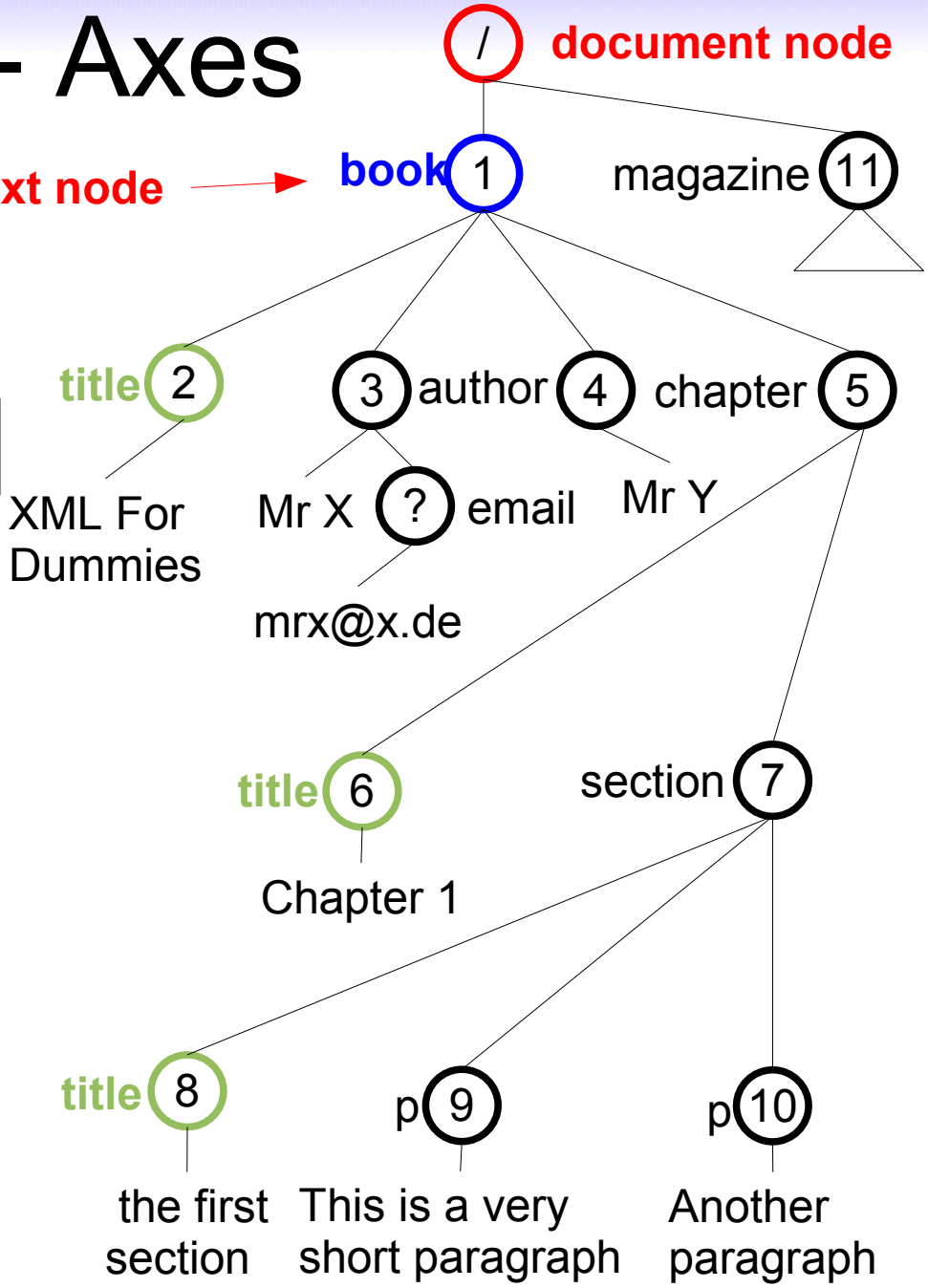
- descendant-axis

/child::book/descendant-or-self::title

user

short form

/book//title



XPath - Axes

further axes:

Ancestor	Parent, parent of parent and so on
Ancestor-or-self	Ancestor + context node
Parent	Parent node of the context node
Descendant	Children and children of children and so on
Following	Following nodes in document order, without descendants
Following-sibling	Following siblings in document order
Preceding	Preceding nodes in document order, without descendants
Preceding-sibling	Preceding siblings in document order
Attribute	Attributes of the context node
Namespace	Namespace nodes of the context node
Self	Context node

XPath - Predicates

- **axis::nodetest[predicate]**
- **contains:**

  <string> x <string> → *<bool>*

user

/book[./title[contains(text(), "XML")]]

```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author>Mr X</author>
   <author>Mr Y</author>
   <chapter>
      ...
   </chapter>
   ...
</book>
<book>
   <title>Easy Cooking</title>
   <chapter>
     ...
   </chapter>
  ...
</book>
...
```

XPath - Predicates

- **axis::nodetest[predicate]**
- **contains:**

  <string> x <string> → *<bool>*

user

```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author>Mr X</author>
   <author>Mr Y</author>
   <chapter>
      ...
   </chapter>
   ...
</book>
<book>
   <title>Easy Cooking</title>
   <chapter>
      ...
   </chapter>
  ...
</book>
...
```

/book[./title[contains(text(), "XML")]]

target

XPath - Predicates

- **axis::nodetest[predicate]**
- **contains:**

  <string> x <string> → <bool>

user

```
/book[./title[contains(text(), "XML")]]
```

target

```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author>Mr X</author>
   <author>Mr Y</author>
   <chapter>
      ...
   </chapter>
   ...
</book>
<book>
   <title>Easy Cooking</title>
   <chapter>
     ...
   </chapter>
  ...
</book>
...
```

XPath - Predicates

- **axis::nodetest[predicate]**
- **contains:**

  <string> x <string> → <bool>

user

```
/book[./title[contains(text(), "XML")]]
```

target

```
<?xml version="1.0" encoding=...>
<book>
   <titel>XML For Dummies</titel>
   <author>Mr X</author>
   <author>Mr Y</author>
   <chapter>
      ...
   </chapter>
   ...
</book>
<book>
   <title>Easy Cooking</title>
   <chapter>
      ...
   </chapter>
  ...
</book>
...
```

XPath - Predicates

- **axis::nodetest[predicate]**
- **ftcontains:**

<string> x <option>→<score>

```
<?xml version="1.0" encoding=...>
<book>
  <titel>Of Mice and Man</titel>
  <author>John Steinbeck</author>
  <chapter>
    <title>Chapter 1</title>
    <section>
        ...
</book>
<book>
  <title>Town Mouse, Country Mouse</title>
  <author>Jan Brett</author>
...
</book>
```

user

/book[./title ftcontains("mouse" with thesaurus)]

target

XPath - Predicates

- **axis::nodetest[predicate]**
- **ftcontains:**

<string> x <option>→<score>

```
<?xml version="1.0" encoding=...>
<book>
  <titel>Of Mice and Man</titel>
  <author>John Steinbeck</author>
  <chapter>
    <title>Chapter 1</title>
    <section>
        ...
</book>
<book>
  <title>Town Mouse, Country Mouse</title>
  <author>Jan Brett</author>
...
</book>
```

user

/book[./title ftcontains("mouse" with thesaurus)]

target

XPath - Predicates

- **axis::nodetest[predicate]**
- **ftcontains:**

<string> x <option> → <score>

```
<?xml version="1.0" encoding=...>
<book>                                   2
  <titel>Of Mice and Man</titel>
  <author>John Steinbeck</author>
  <chapter>
    <title>Chapter 1</title>
    <section>
      ...
</book>
<book>                                   1
  <title>Town Mouse, Country Mouse</title>
  <author>Jan Brett</author>
...
</book>
```

rank

user

/book[./title ftcontains("mouse" with thesaurus)]

target

TopX

- Top-k retrieval engine
  - XML retrieval
  - Full text search
  - **N**arrowed **E**xtended **X**path **I** (NEXI)
- Based on:
  - Threshhold algorithm
  - Optimized index-acces
  - Incremental query expansion

NEXI

Narrowed **E**xtended **X**Path **I**

- „subset“ of Xpath
- descendant-or-self axis only
- small syntactic differences
- designed for „content only“ and „content and structure“ queries

NEXI - Examples

co:

- Hello (term)
- „Hello World“ (phrase)
- Hello - World (negative term)

cas:

- //book[about(.//title, mouse)]

Goal of this thesis

- XPath 2.0 Parser for TopX
- Rewriting of Xpath 2.0 Queries
  - transformation to TopX datastructures

Architecture

Xpath 2.0 Query

Architecture

user

Xpath 2.0 Query

Parser

Architecture

user

Xpath 2.0 Query

Parser

syntaxtree

Rewriter

Architecture

user

Xpath 2.0 Query

Parser

syntaxtree

Rewriter

TopX

Architecture

user

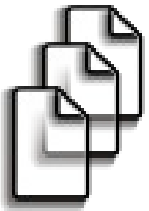
Xpath 2.0 Query

Parser

syntaxtree

Rewriter

TopX



Top k results

Architecture

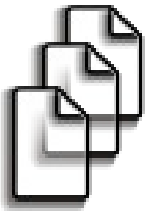
Xpath 2.0 Query

Parser

syntaxtree

Rewriter

TopX



Top k results

Phase1: Parser & Lexer

Jflex

Fast Lexical Analyser Generator

- Generates a Lexer
- specification file based on regular expressions
- Generated Lexer provides the parser with tokens and constant values

Cup

LALR Parser Generator

- Generates a parser
- Specification file based on LR-grammar
- allows to carry out java functions at various points of production rules

LARL := **L**ook**a**head, **l**eft to **r**ight parser

Phase1: Parser & Lexer

Production Rules for the Parser Generator

```
/* Productions */

LocationPath     ::=

   RelativeLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :}
 | AbsoluteLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :};

AbsoluteLocationPath ::=

   SLASH
     {: RESULT = parser.concatAbsoluteLocationPath(null); :}
 | SLASH RelativeLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :}
 | AbbreviatedAbsoluteLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :};
```

Phase1: Parser & Lexer

Production Rules for the Parser Generator

```
/* Productions */

LocationPath     ::=

   RelativeLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :}
 | AbsoluteLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :};

AbsoluteLocationPath ::=

   SLASH
     {: RESULT = parser.concatAbsoluteLocationPath(null); :}
 | SLASH RelativeLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :}
 | AbbreviatedAbsoluteLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :};
```

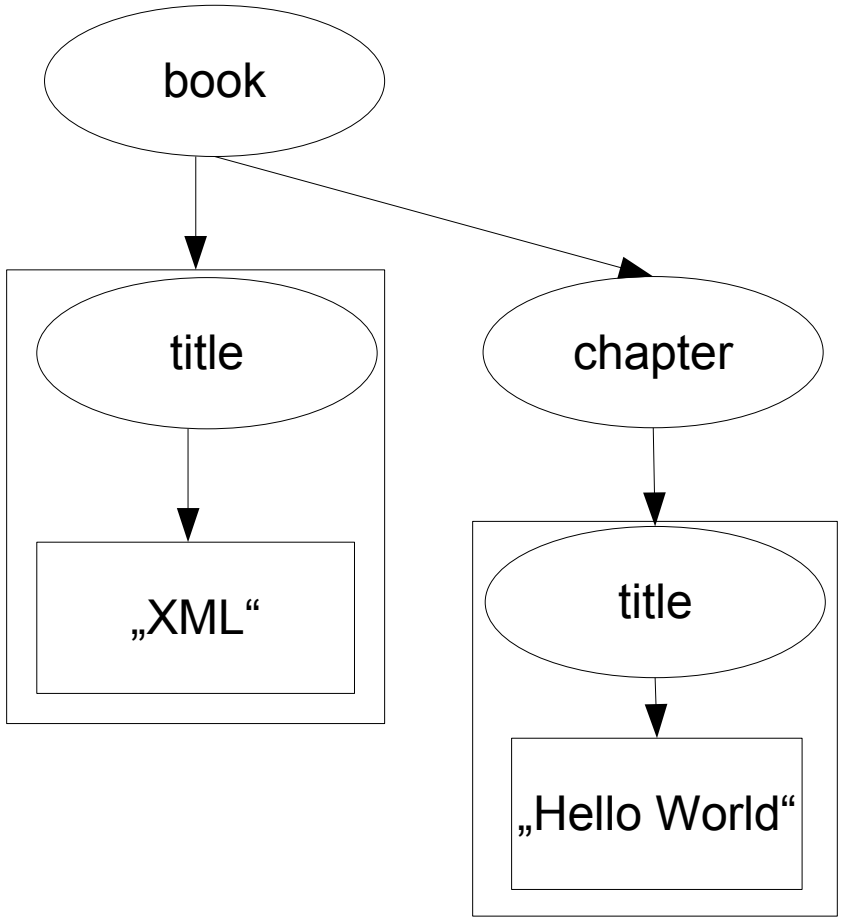
Phase1: Parser & Lexer

Production Rules for the Parser Generator

```
/* Productions */

LocationPath     ::=

   RelativeLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :}
 | AbsoluteLocationPath:xTree
     {: RESULT = parser.concatLocationPath(xTree); :};

AbsoluteLocationPath ::=

   SLASH
     {: RESULT = parser.concatAbsoluteLocationPath(null); :}
 | SLASH RelativeLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :}
 | AbbreviatedAbsoluteLocationPath:xTree
     {: RESULT = parser.concatAbsoluteLocationPath(xTree); :};
```

Phase2: Rewriting

Tree structure:

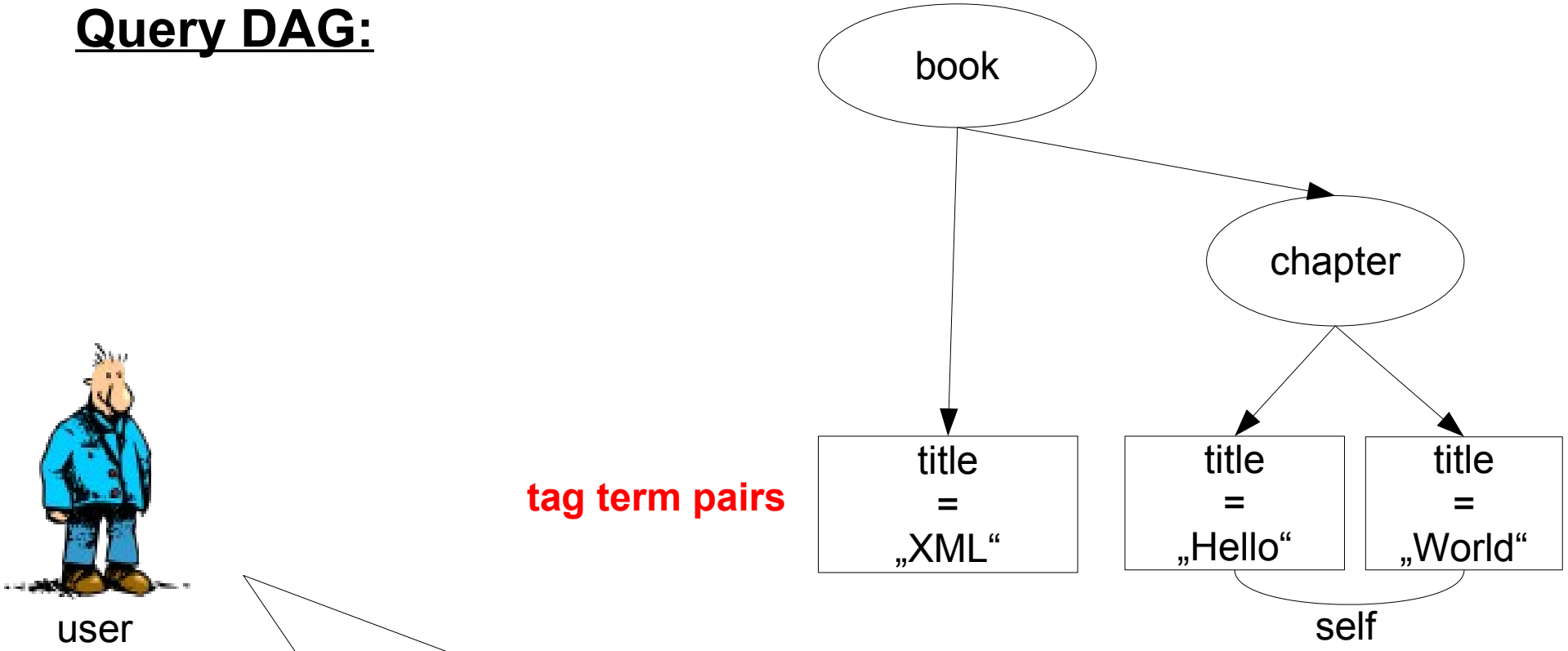


user

/book[./title[contains(text(), “XML“)] AND
./chapter/title[contains(text(), „Hello World“)]]

Phase2: Rewriting

Query DAG:



Phase2: Rewriting

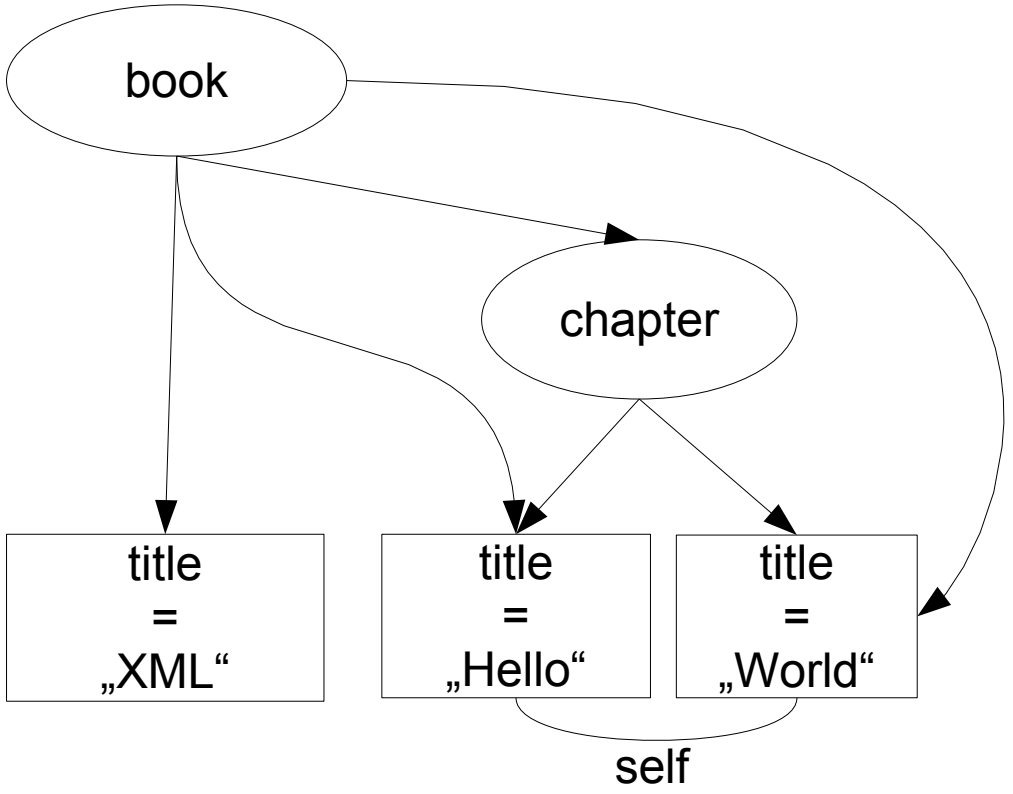
Query DAG:

Transitively expanded
descendant constraints

final TopX format

user

tag term pairs



/book[./title[contains(text(), “XML“)] AND
./chapter/title[contains(text(), „Hello World“)]]

Summary

Xpath 2.0 Query

Parser

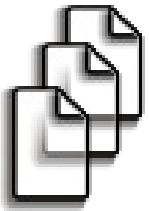
Jflex ⇒ **Cup**

syntaxtree

Rewriter

Query tree representation ⇒ **DAG representation**

TopX



Top k results

Future Work

- Adding of further axis, like child or siblings
- Adding of thesaurus functionality for full text search
- There are much more XPath 2.0 features, that could be included.

Thank you!

Any Questions ?

References:

Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. Mc-Graw Hill, London - New York - Paris, 3. aufl. edition, 2003.

James Clark and Steven DeRose. XML path language (XPath) version 1.0. W3C recommendation, W3C, November 1999. http://www.w3.org/TR/1999/RECxpath-19991116.

Ashok Malhotra and Norman Walsh and Jim Melton, XQuery 1.0 and XPath 2.0 Functions and Operators, W3C} Recommendation, W3C, January 2007, http://www.w3.org/TR/2007/REC-xpath-functions-20070123

Martin-Theobald, Efficient Top-k Query Processing for Text, Semistructured, and Structured Data, Universität des Saarlandes, May 2006