Web Data Models

XPath: Syntax, Semantics

Silviu Maniu

UNIVERSITÉ PARIS SUD

Comprendre le monde,
construire l'avenir

université PARIS-SACLAY

XPath

- once we have an XML document, we want to query it and retrieve data

- **XPath**: query language to select a sequence of nodes from an XML

- W3C standard — also used in other standards (XQuery, XSLT, XPointer, XLink, …); supported by every modern browser

XPath

- XPath is a navigational language — specifies how the XML documents should be traversed

- **Main issue**: big volume of nodes can be extracted via XPath, so efficient processing is still an ongoing challenge (we'll talk about this in future sessions)

Lecture Outline

- XPath axes
- XPath tests
- XPath nodes

XPath: Quick Example

`Q1 = /bib/book/title`

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

Q1 = /bib/book/title

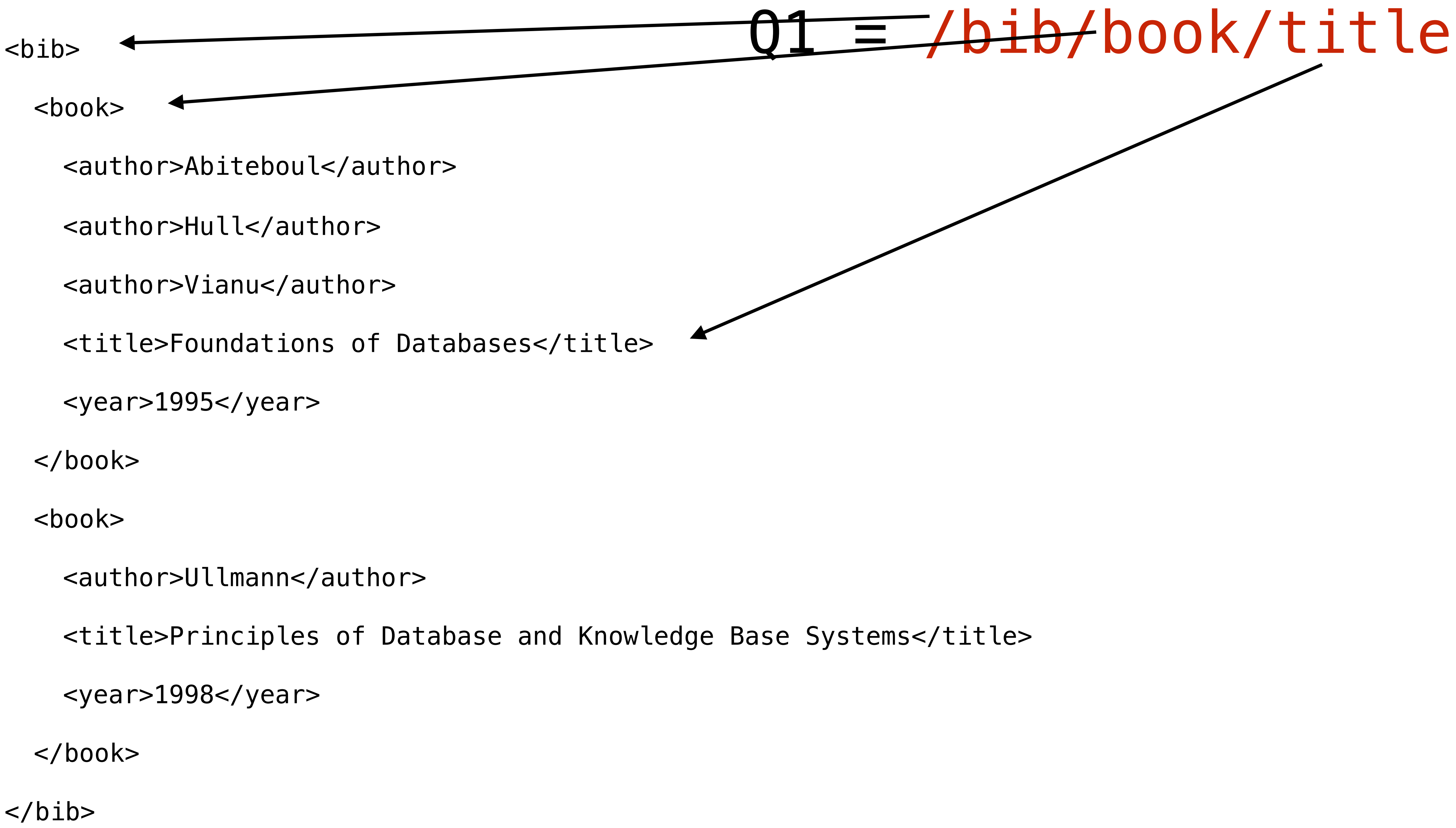
```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

Q1 = /bib/book/title

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example



Q1 = /bib/book/title

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

Q1 = `/bib/book/title`

R1 = **<title>Foundations of Databases</title>**

R2 = **<title>Principles of Database and Knowledge Base Systems</title>**

XPath

XPath is composed of a sequence of context nodes and a step:

$$\mathrm{cs}_0/\mathrm{step}$$

- after the context sequence, take a step in a given direction
- can also have multi-steps:

$$\mathrm{cs}_0/\mathrm{step}_1/\mathrm{step}_2/\ldots$$

$$\mathrm{cs}_1/\mathrm{step}_2/\ldots$$

XPath: Quick Example

`Q1 = /bib/book/title`

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

Q1 = **/bib**/book/title

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

Q1 = /bib/book/title

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Quick Example

```
Q1 = /bib/book/title
```

```
<bib>
  <book>
    <author>Abiteboul</author>
    <author>Hull</author>
    <author>Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
  </book>
  <book>
    <author>Ullmann</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
</bib>
```

XPath: Syntax

Each step has the syntax:

$$\text{axis} :: \text{ntest}[\text{pred}_1] \cdots [\text{pred}_n]$$

- **axis** specifies the direction of navigation to be followed,
- **ntest** specifies the node test (used to navigate only nodes of a certain kind),
- the optional **pred**, which allows filters on the sequence of nodes we navigate to.

XPath: Axes

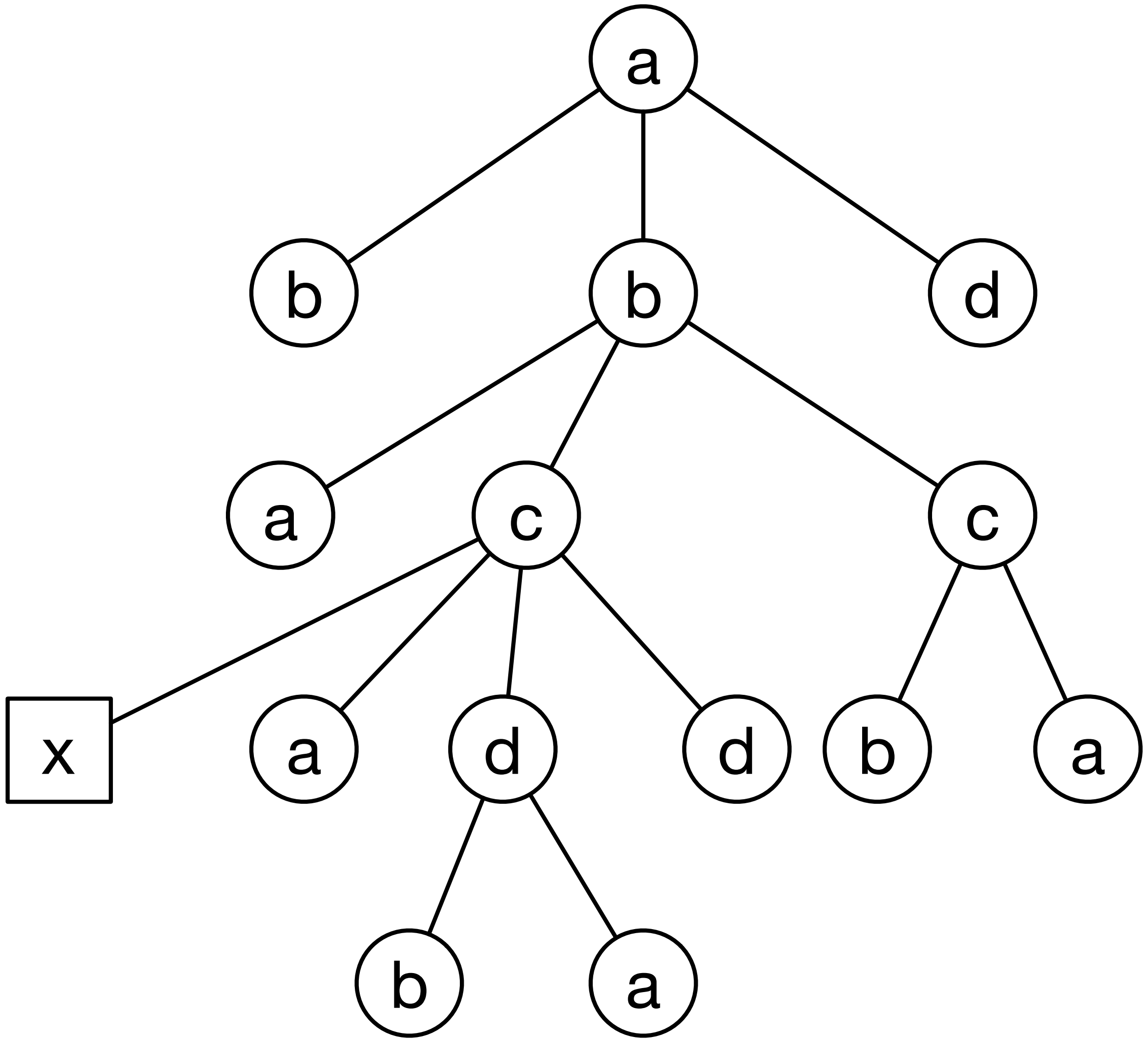
$$\boxed{\mathrm{axis}} :: \mathrm{ntest}[\mathrm{pred}_1] \cdots [\mathrm{pred}_n]$$

XPath has a family of **12 axes** allowing for flexible navigation within the node hierarchy

- forward axes (in document order)
- backward axes (in reverse document order)

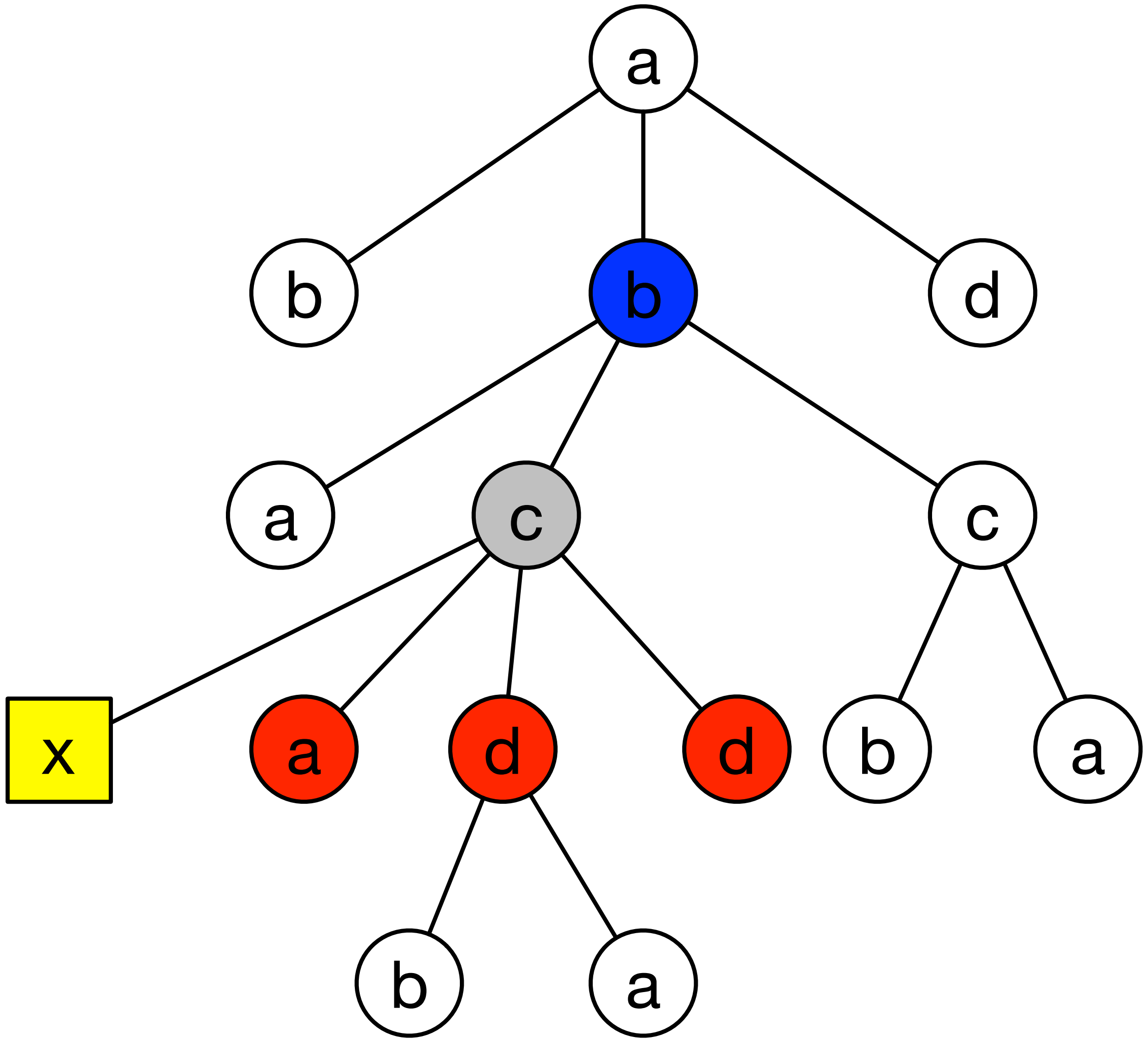
XPath: Axes

```
<a>
  <b />
  <b>
    <a />
    <c x=“1.0”>
      <a>text1</a>
      <d>
        <b />
        <a />
      </d>
    </c>
    <c>
      <b />
      <a>text2</a>
    </c>
  </b>
  <d />
</a>
```



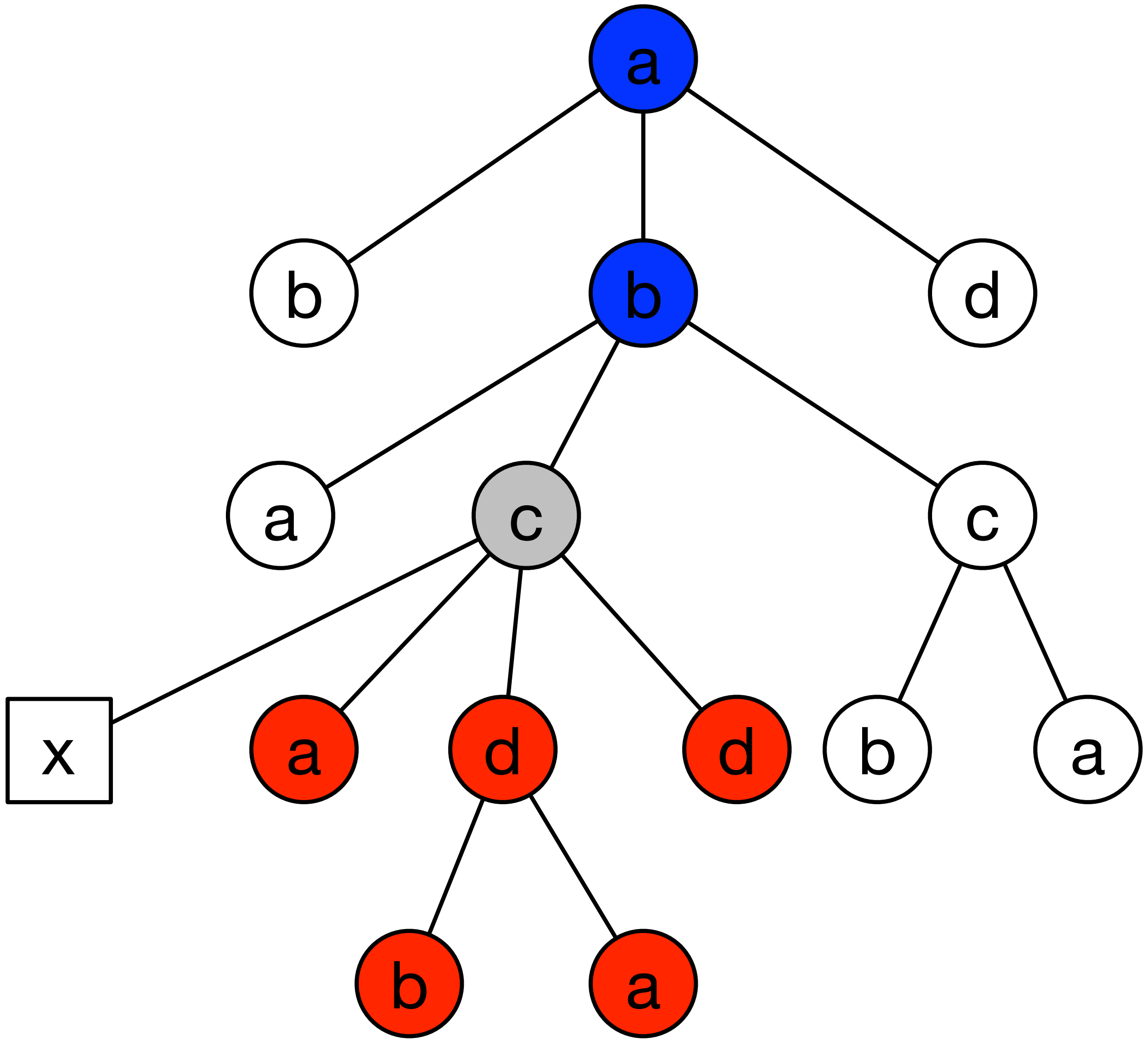
XPath: Axes

- `self` (fwd)
- `child` (fwd)
- `parent` (back)
- `attribute` (fwd)



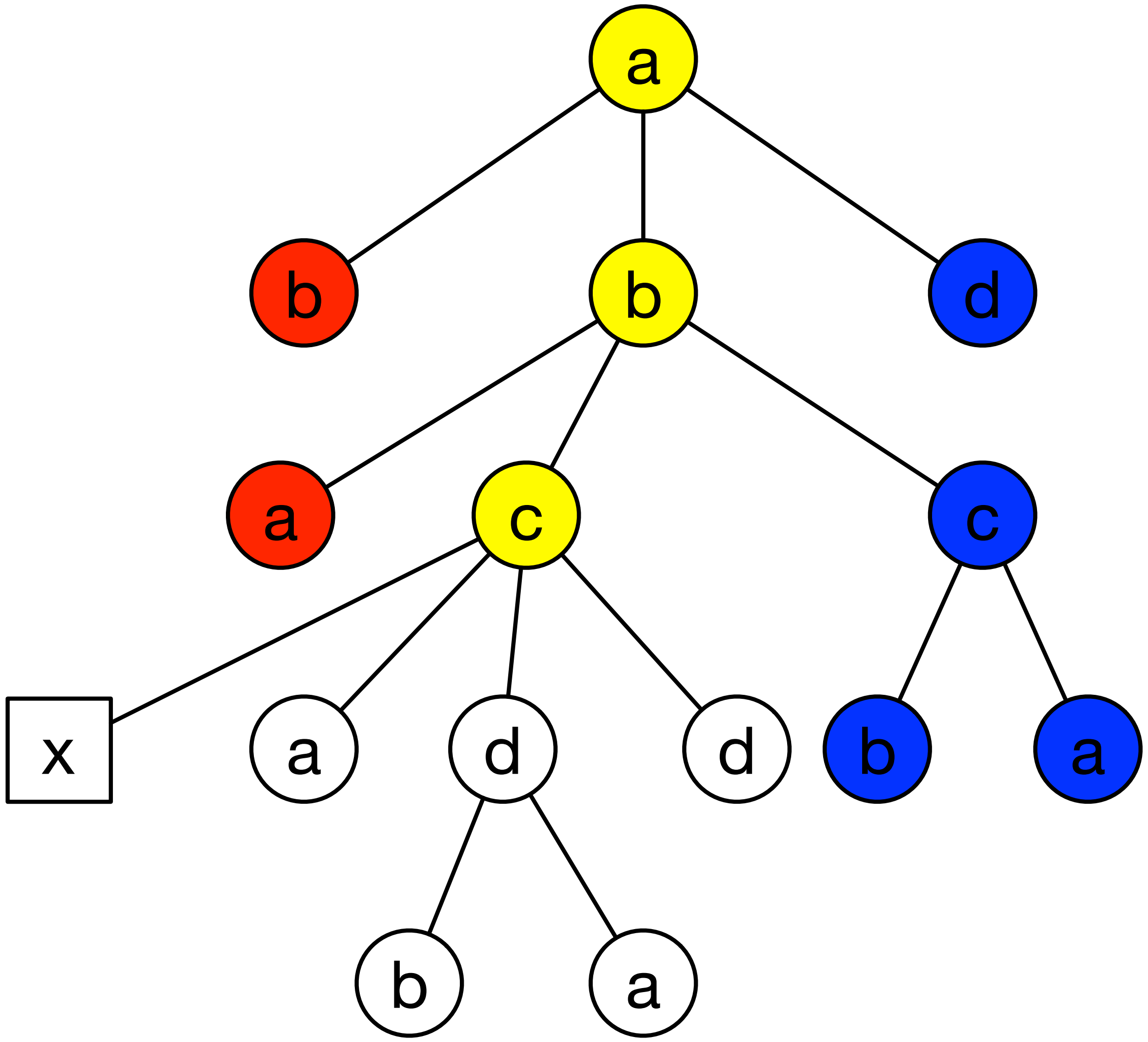
XPath: Axes

- `self` (fwd)
- `descendant` (fwd)
- `ancestor` (back)



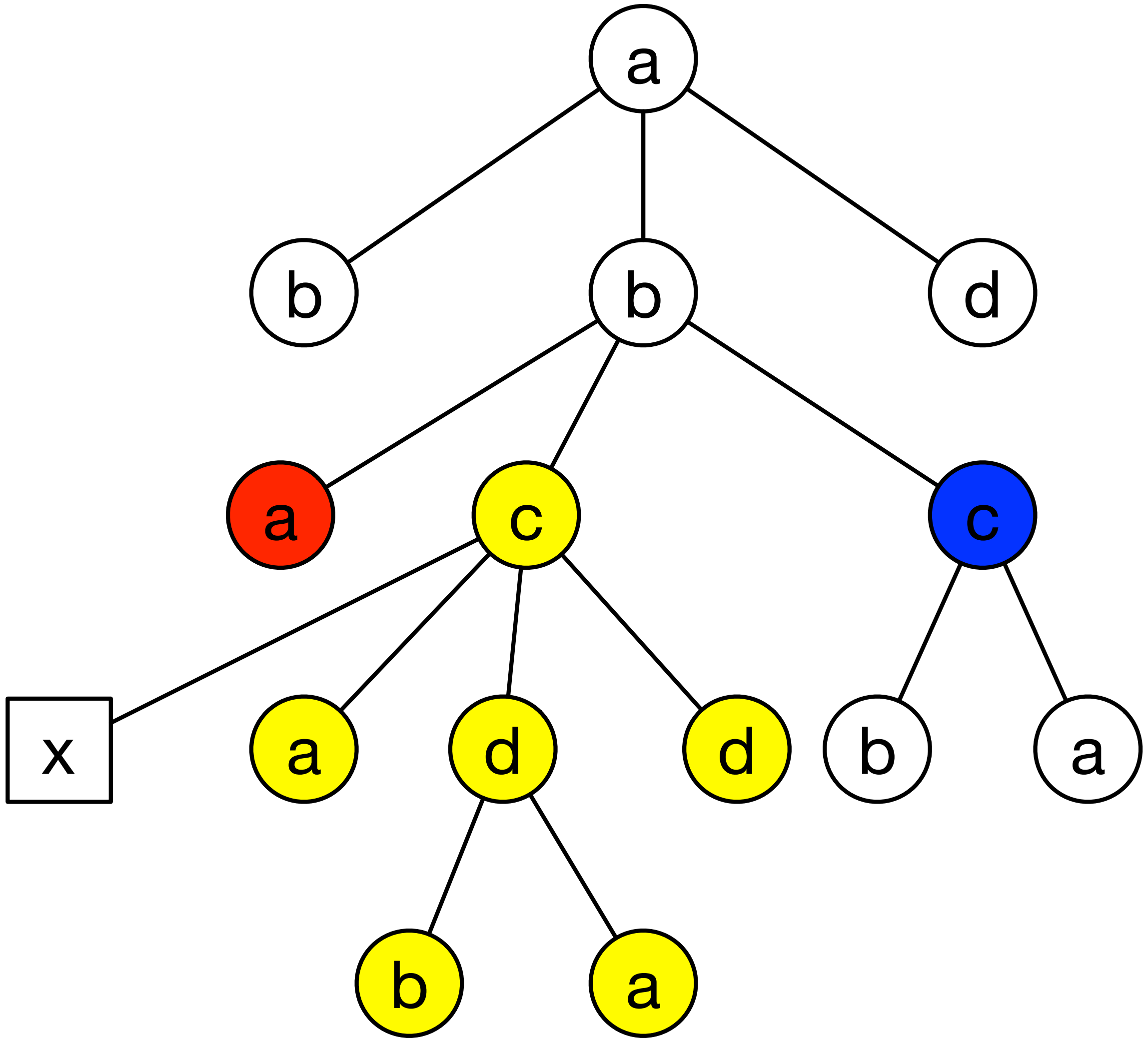
XPath: Axes

- `preceding` (back)
- `following` (fwd)
- `ancestor-or-self` (back)



XPath: Axes

- `preceding-sibling` (back)
- `following-sibling` (fwd)
- `descendant-or-self` (fwd)



XPath: Result Order

The result node sequence of any XPath navigation is returned in document order with no duplicate nodes.

XPath: Node Tests

$$\text{axis} :: \text{ntest}[\text{pred}_1] \cdots [\text{pred}_n]$$

Can also apply a **node test** to filter nodes based on kind and name.

Test	Semantics
`node()`	let any node pass
`text()`	preserve only text nodes
`comment()`	preserve only comment nodes
`processing-instruction()`	preserve processing instructions
`processing-instruction(p)`	preserve processing instructions of the form `<?p …p>`
`document-node()`	preserve the document root node

XPath: Node Tests

$$\text{axis} :: \boxed{\text{ntest}}[\text{pred}_1] \cdots [\text{pred}_n]$$

Can also apply a **node test** to filter nodes based on kind and name.

Test	Semantics
`name`	preserve element node with tag *name* only
`*`	preserve element nodes with arbitrary tag names

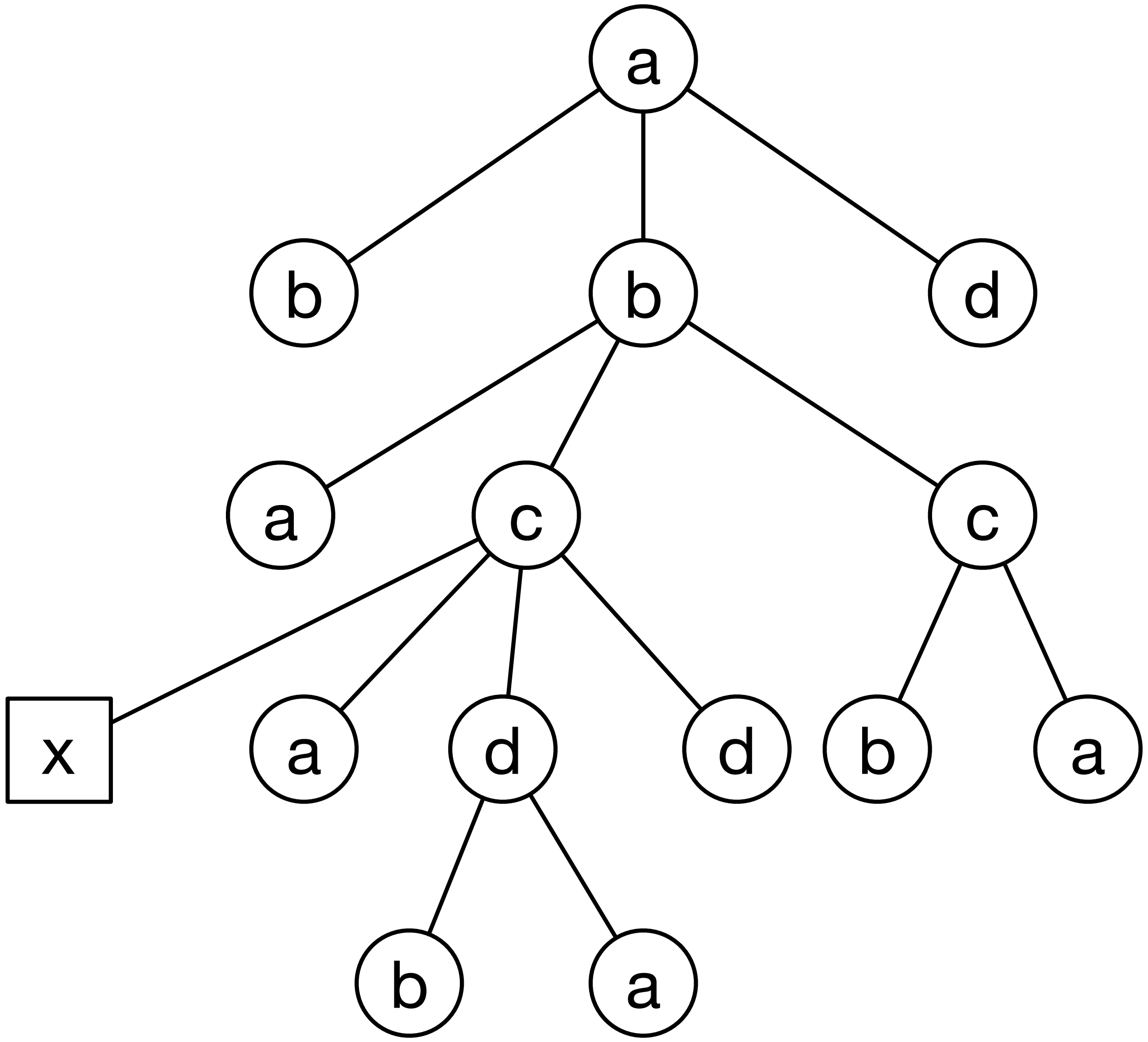
XPath: Abbreviations

There are a few abbreviations in XPath:

`/a`	`/child::a`
`//a`	`/descedant-or-self::node()/child::a`
`//`	`/descendant-or-self::node()`
`.`	`/self::node()`
`..`	`/parent::node()`
`@`	`attribute::`

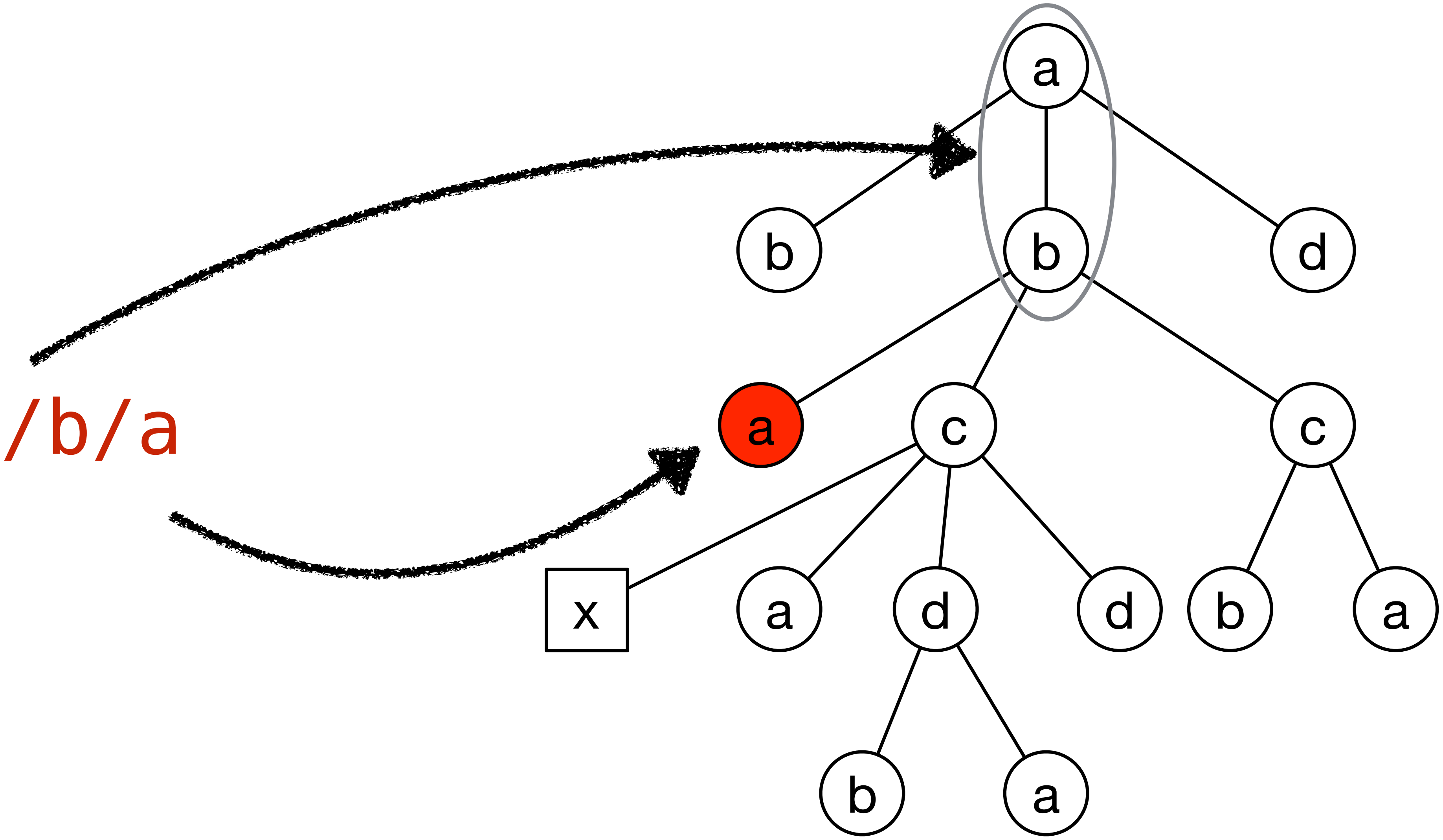
XPath: Examples

```
Q1 := /a/b/a
```



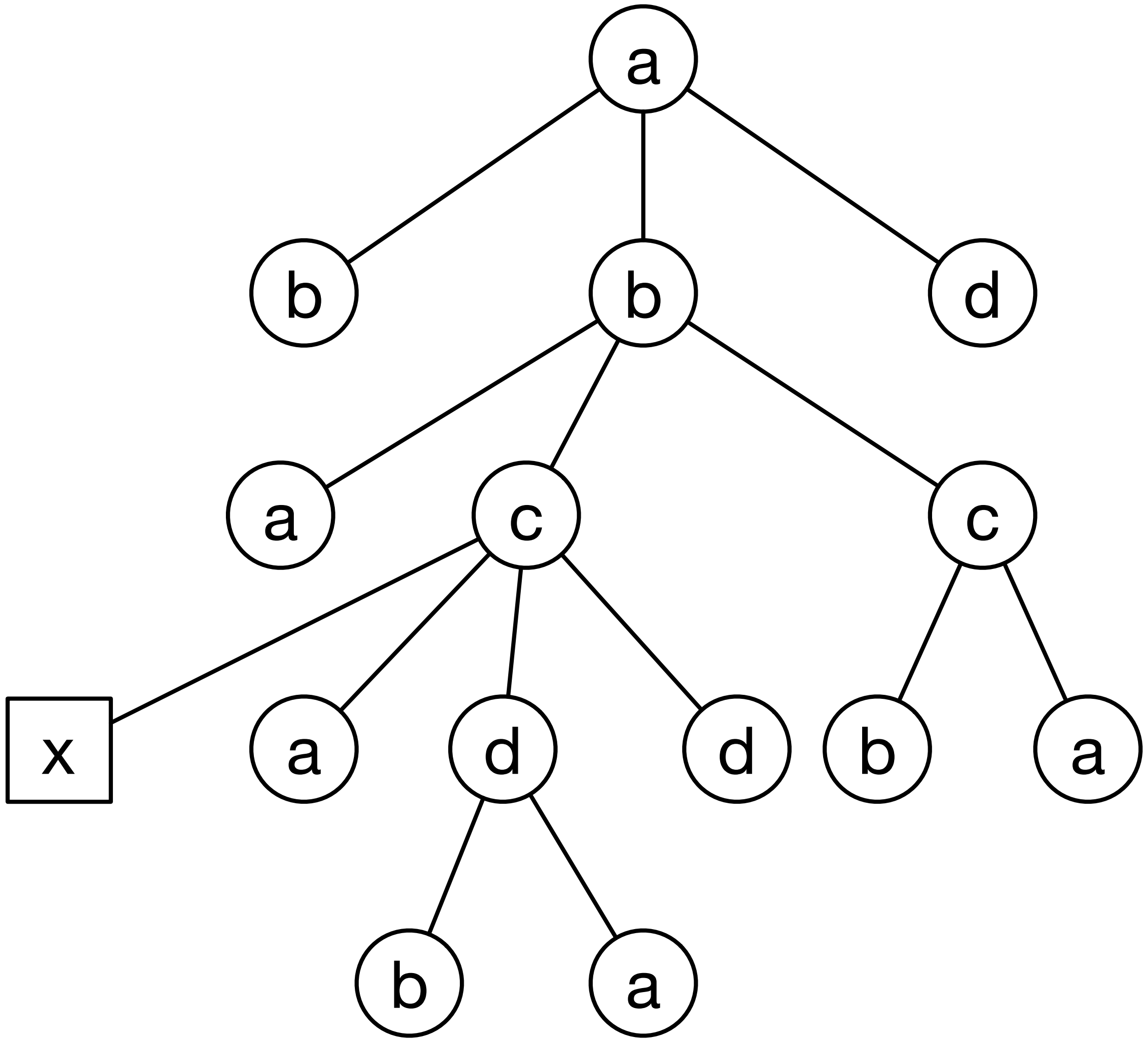
XPath: Examples

Q1 := /a/b/a



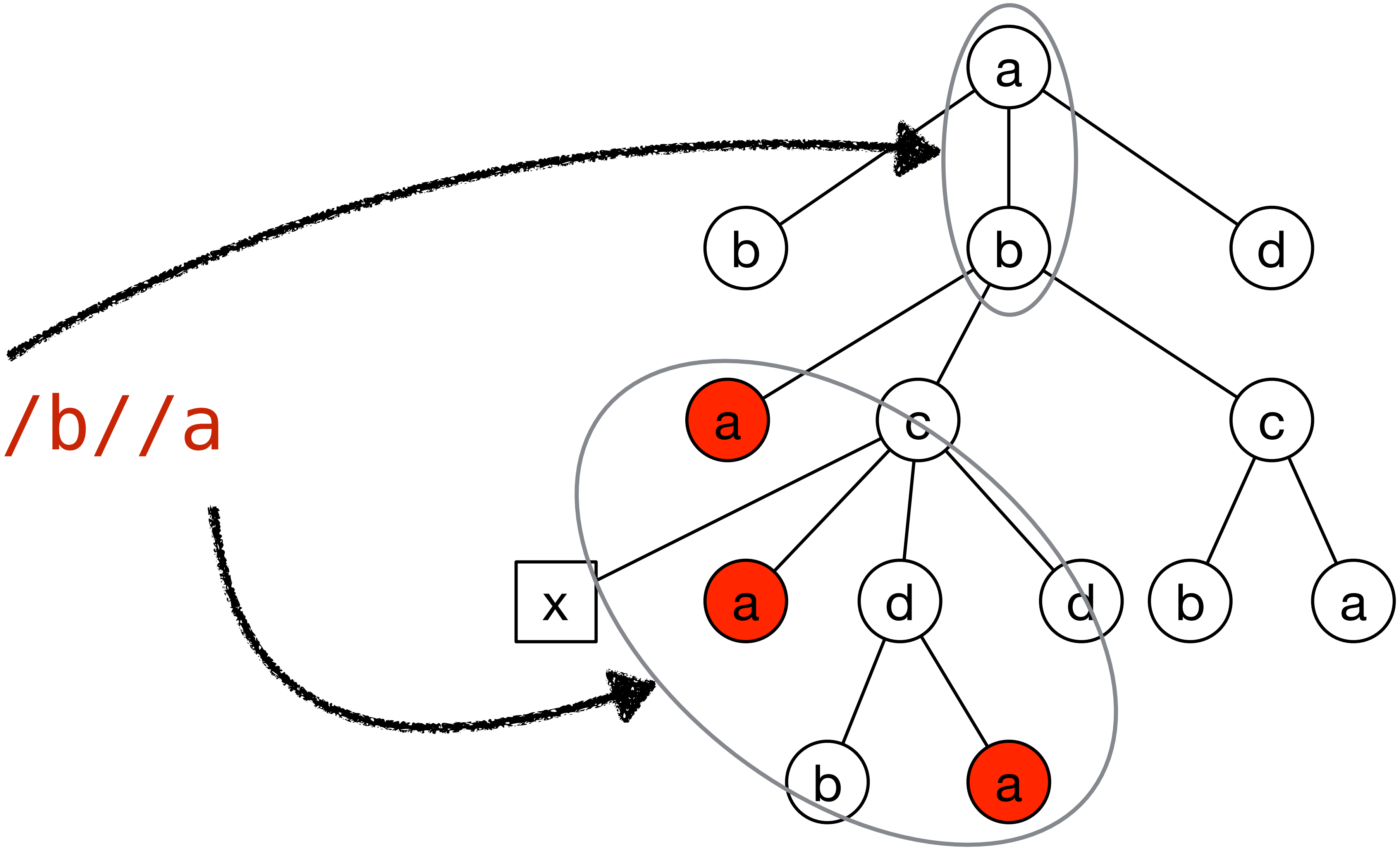
XPath: Examples

Q2 := /a/b//a



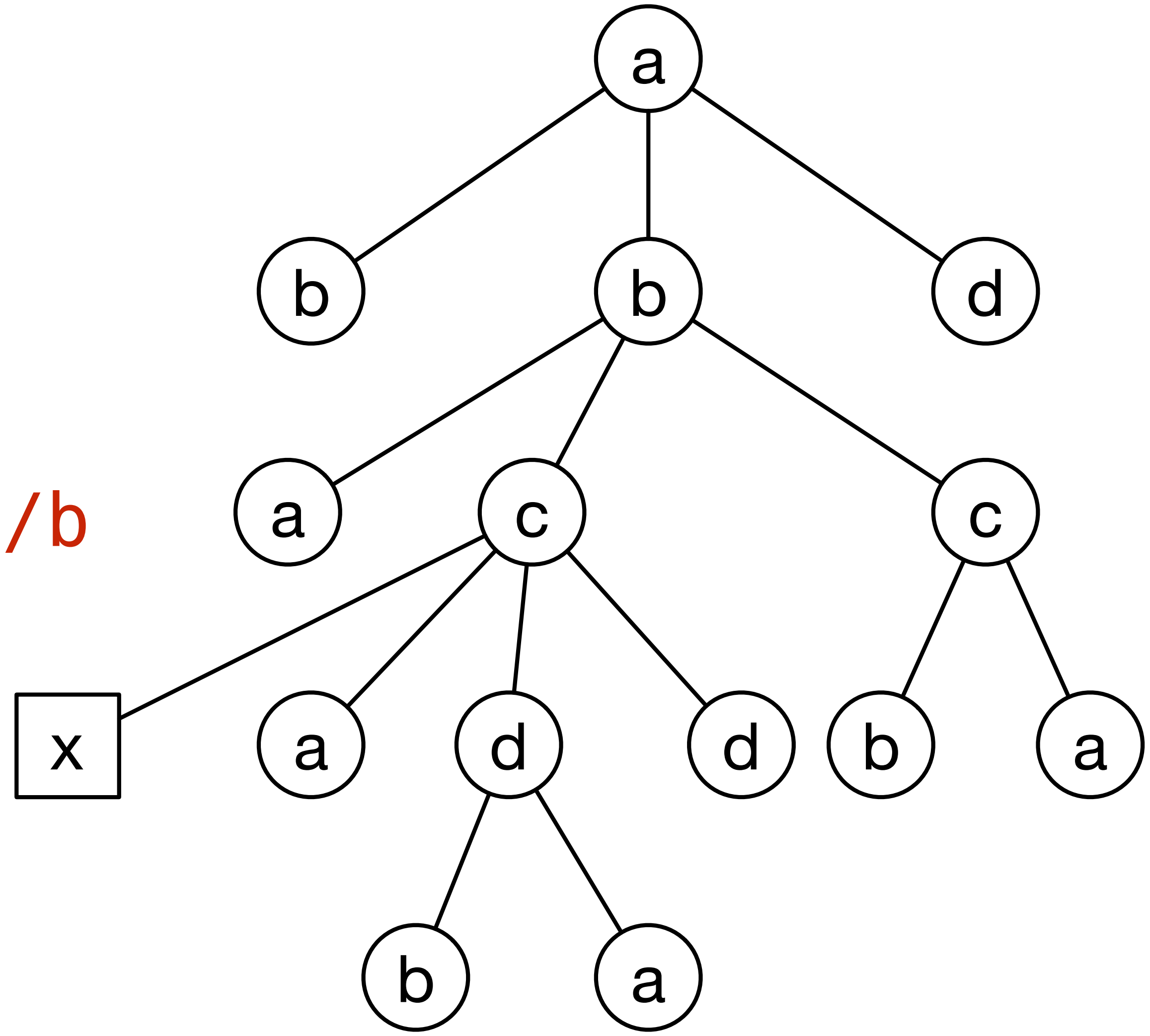
XPath: Examples

Q2 := /a/b//a



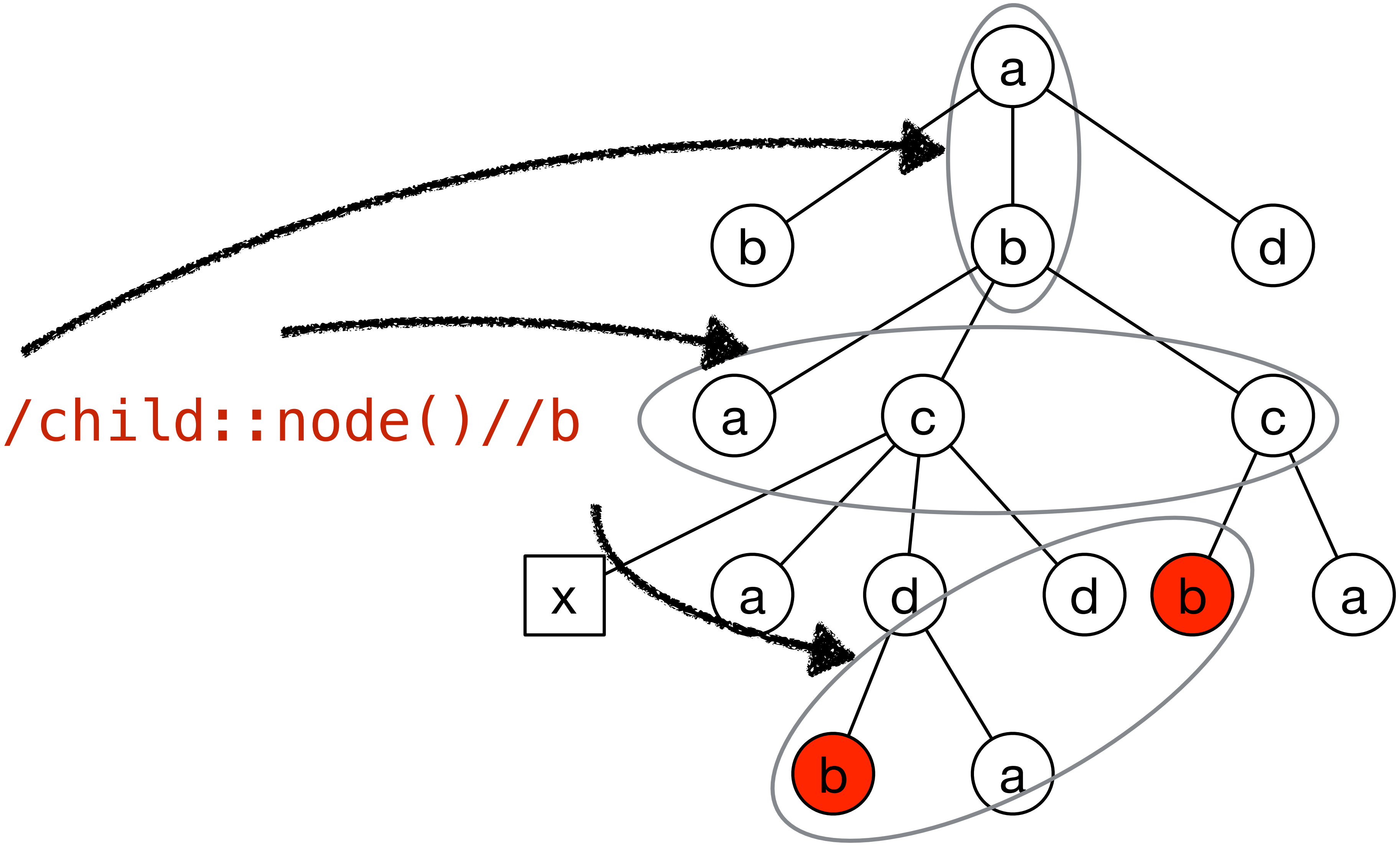
XPath: Examples

```
Q3 := /a/b/child::node()//b
```



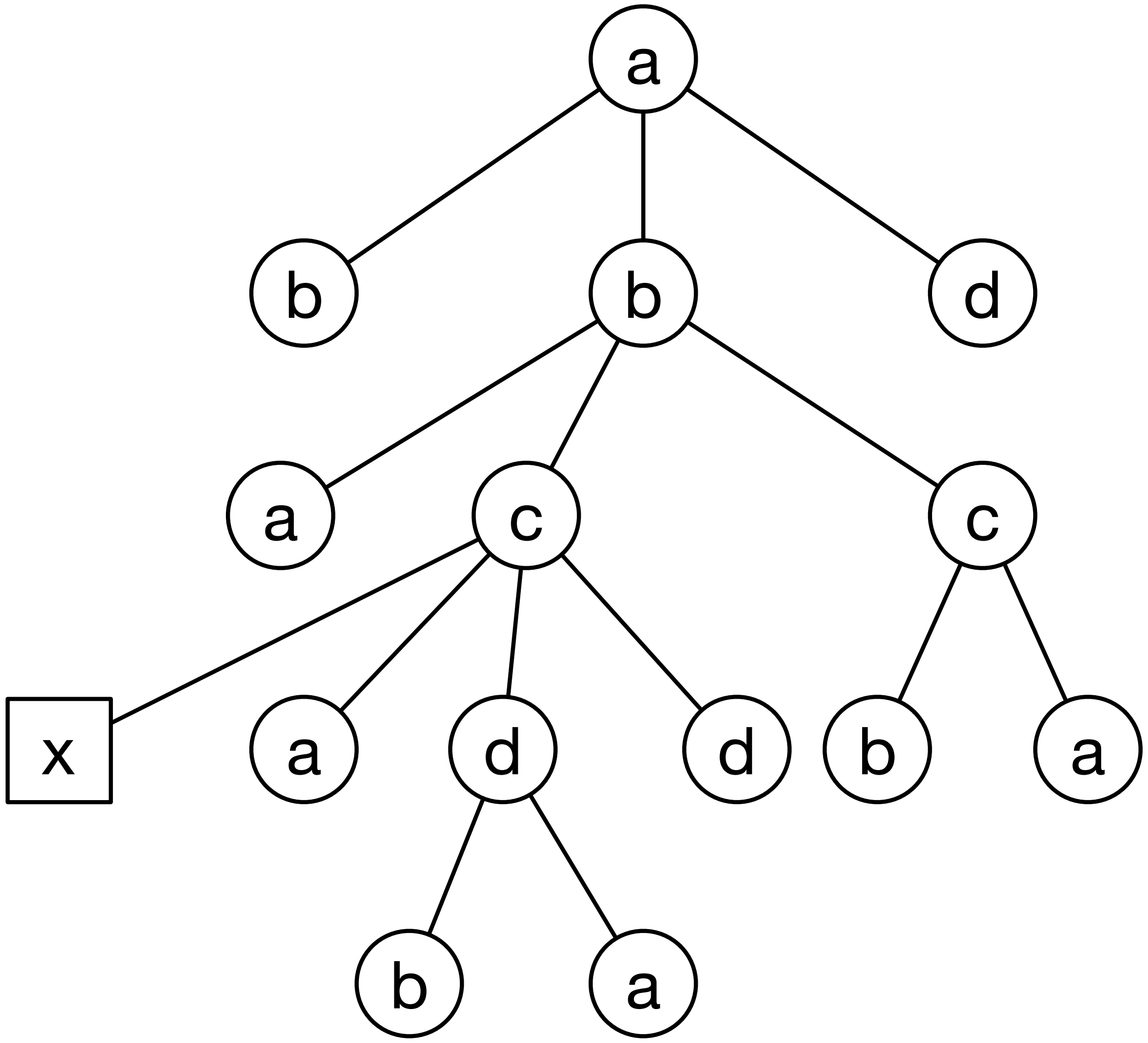
XPath: Examples

Q3 := /a/b/child::node()//b



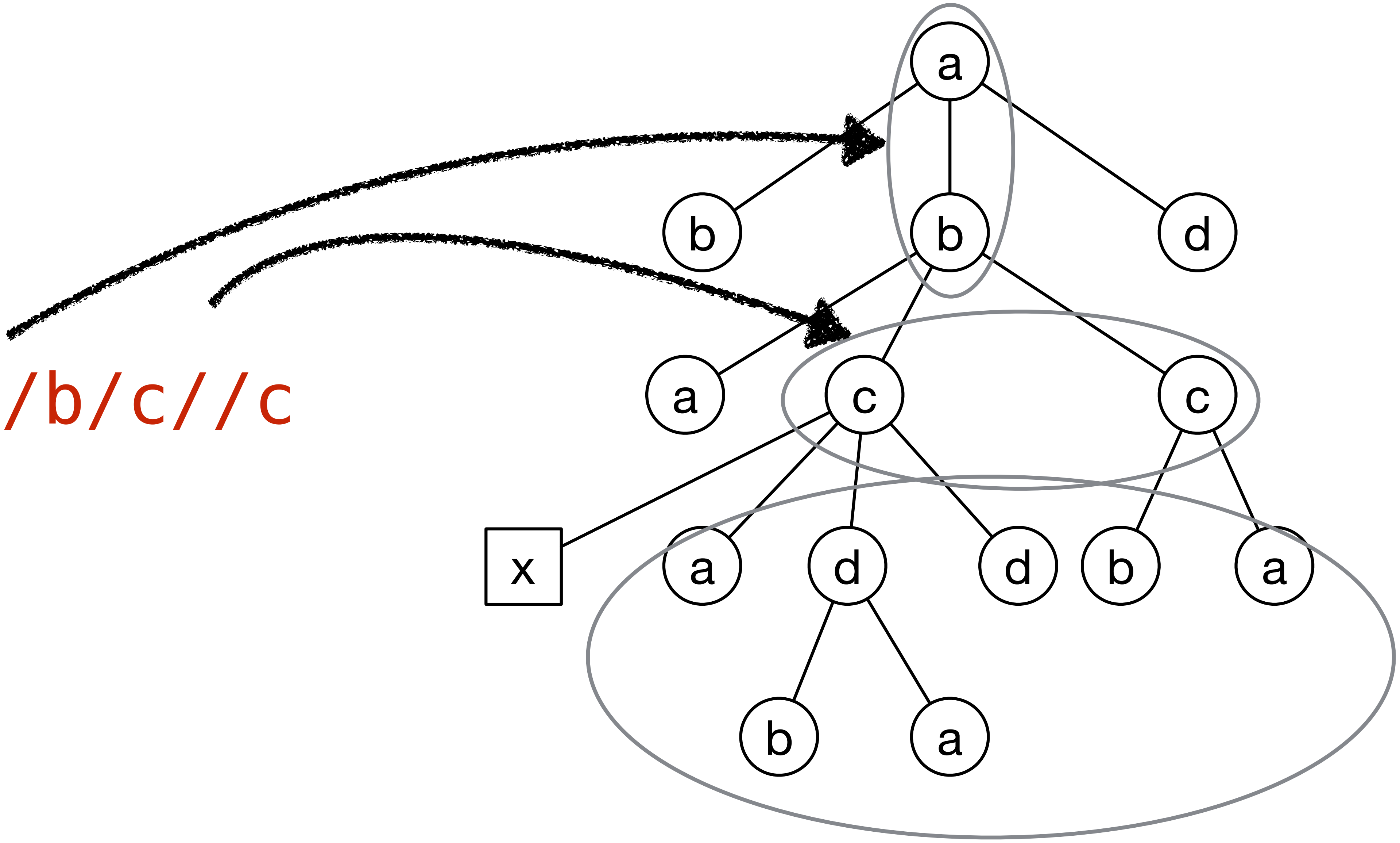
XPath: Examples

Q4 := /a/b/c//c



XPath: Examples

Q4 := /a/b/c//c



XPath: Predicates

$$\text{axis} :: \text{ntest}\boxed{[\text{pred}_1] \cdots [\text{pred}_n]}$$

Predicates are tested against a node, and are optional.

- they have higher precedence than the XPath step,
- are evaluated left to right,
- they may be any XQuery expression which evaluates to a value *v*

XPath: Predicates

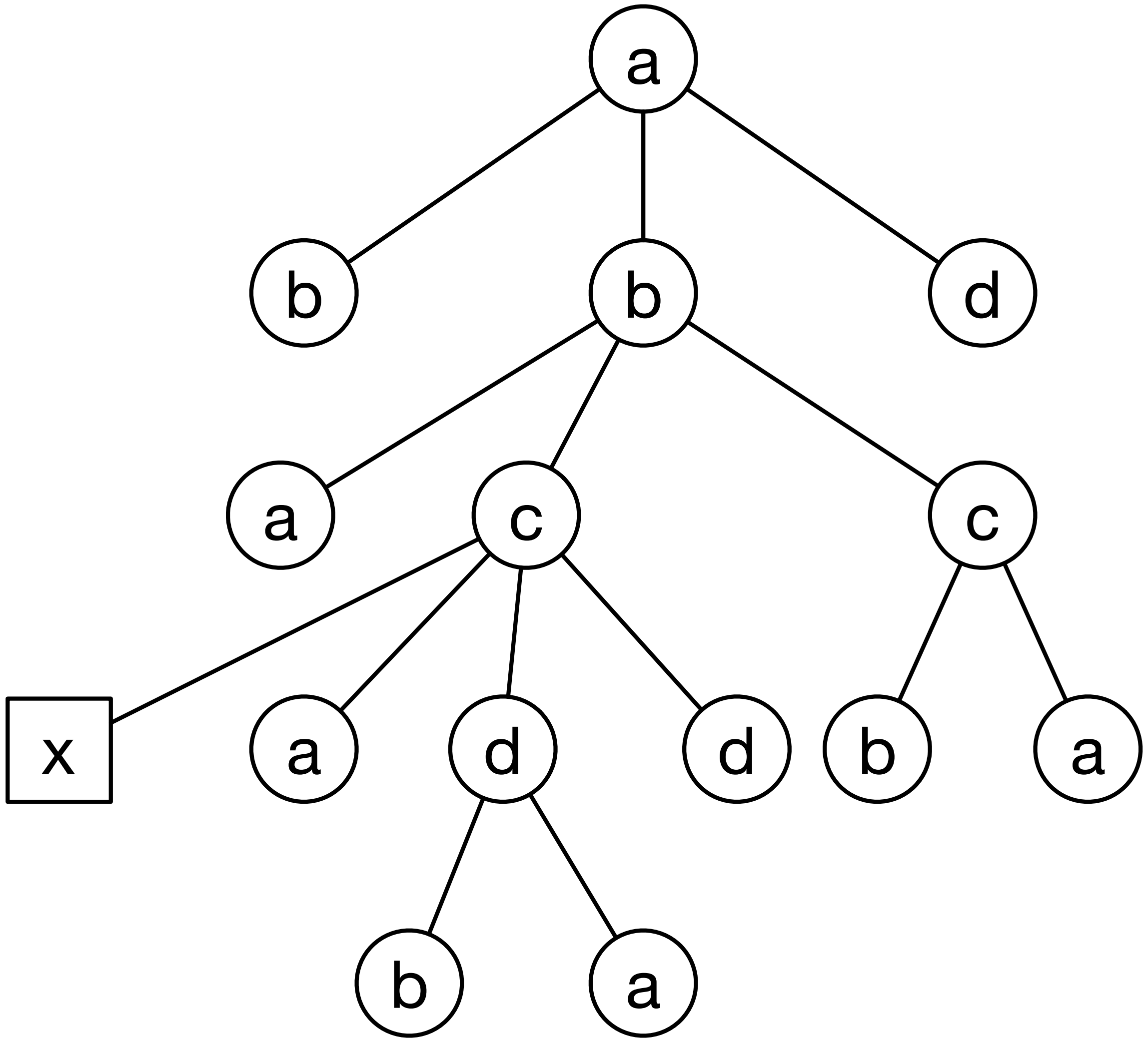
$$\text{axis} :: \text{ntest}[\text{pred}_1] \cdots [\text{pred}_n]$$

XPath calculates an effective boolean value $\mathbf{ebv(v)}$, depending on the value *v*:

v	ebv(v)
`()`	`false()`
`0, NaN`	`false()`
`"“`	`false()`
`false()`	`false()`
`x`	`true()`
`(x1,x2,…,xn)`	`true()`

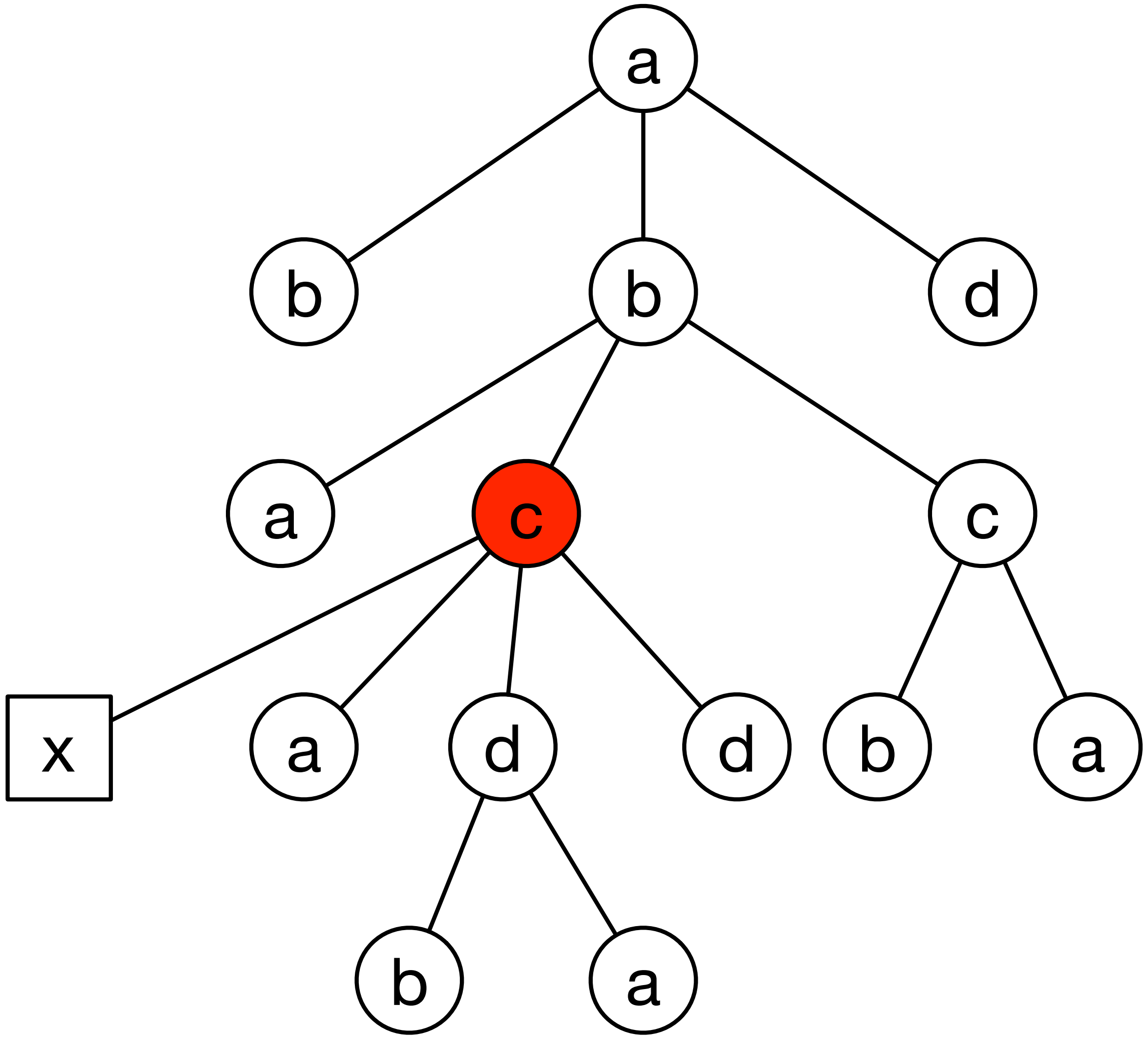
XPath: Examples

```
Q5 := /a/b/c[./d]
```



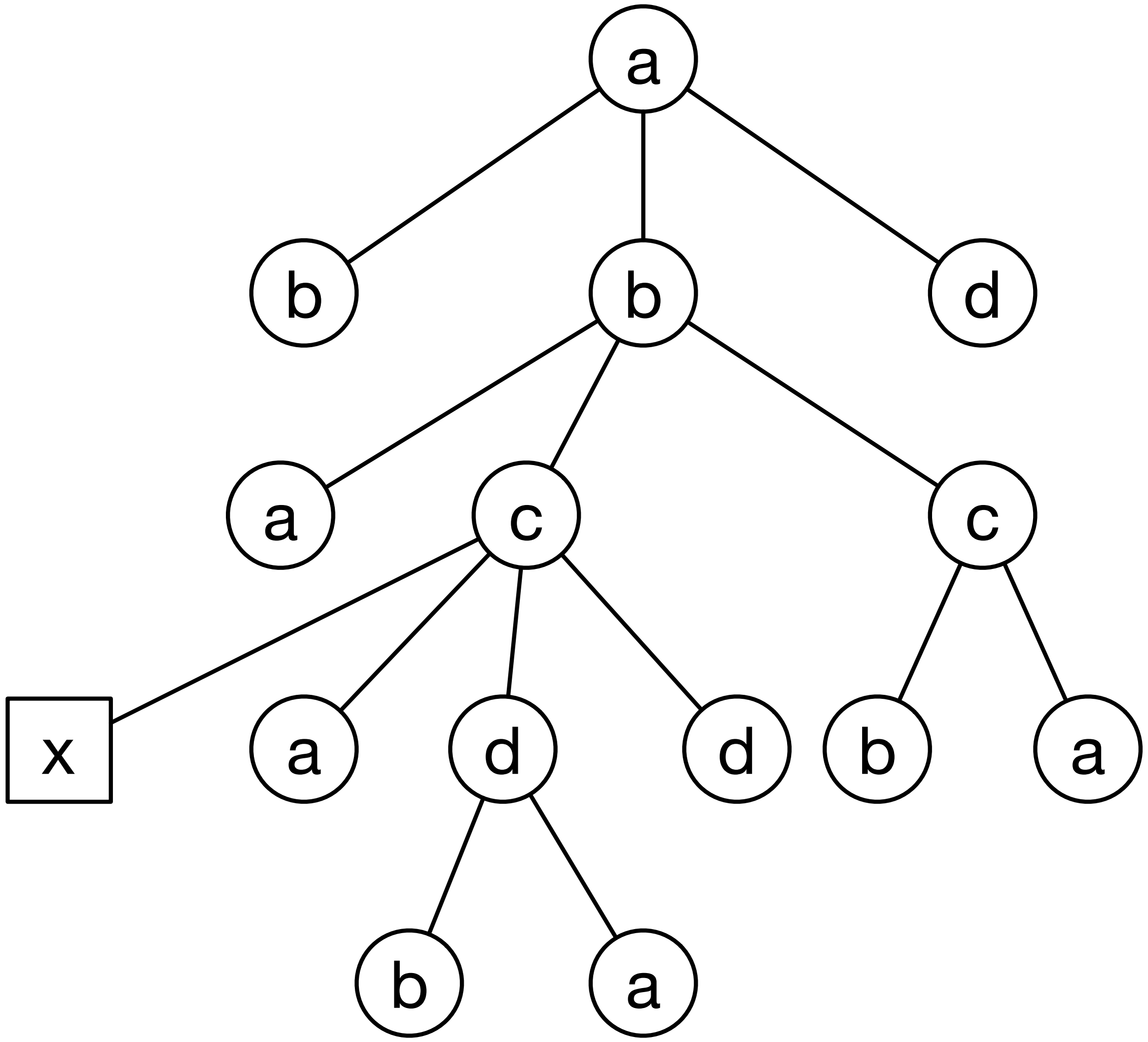
XPath: Examples

`Q5 := /a/b/c[./d]`



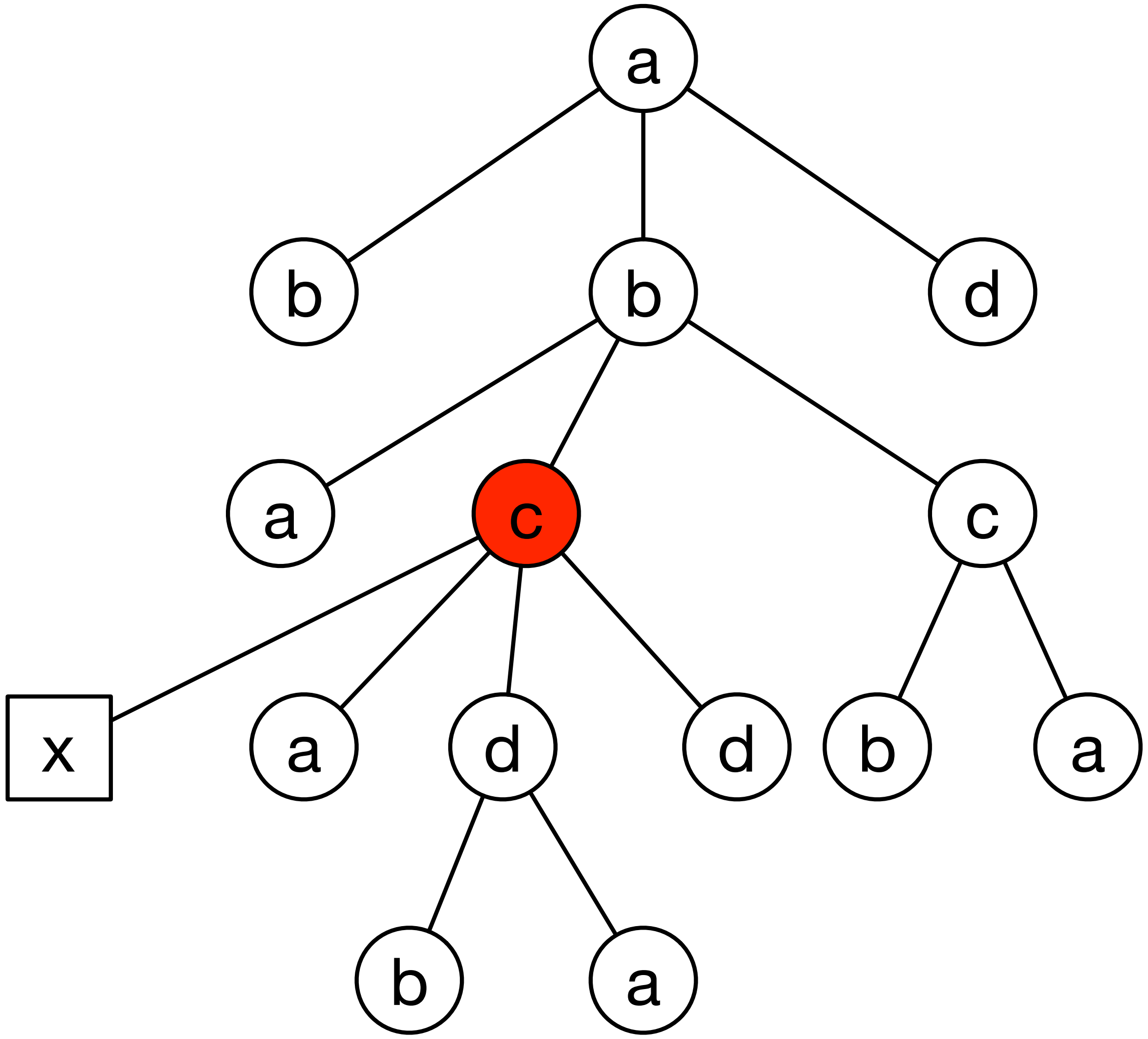
XPath: Examples

```
Q6 := /a/b/c[a and d]
```



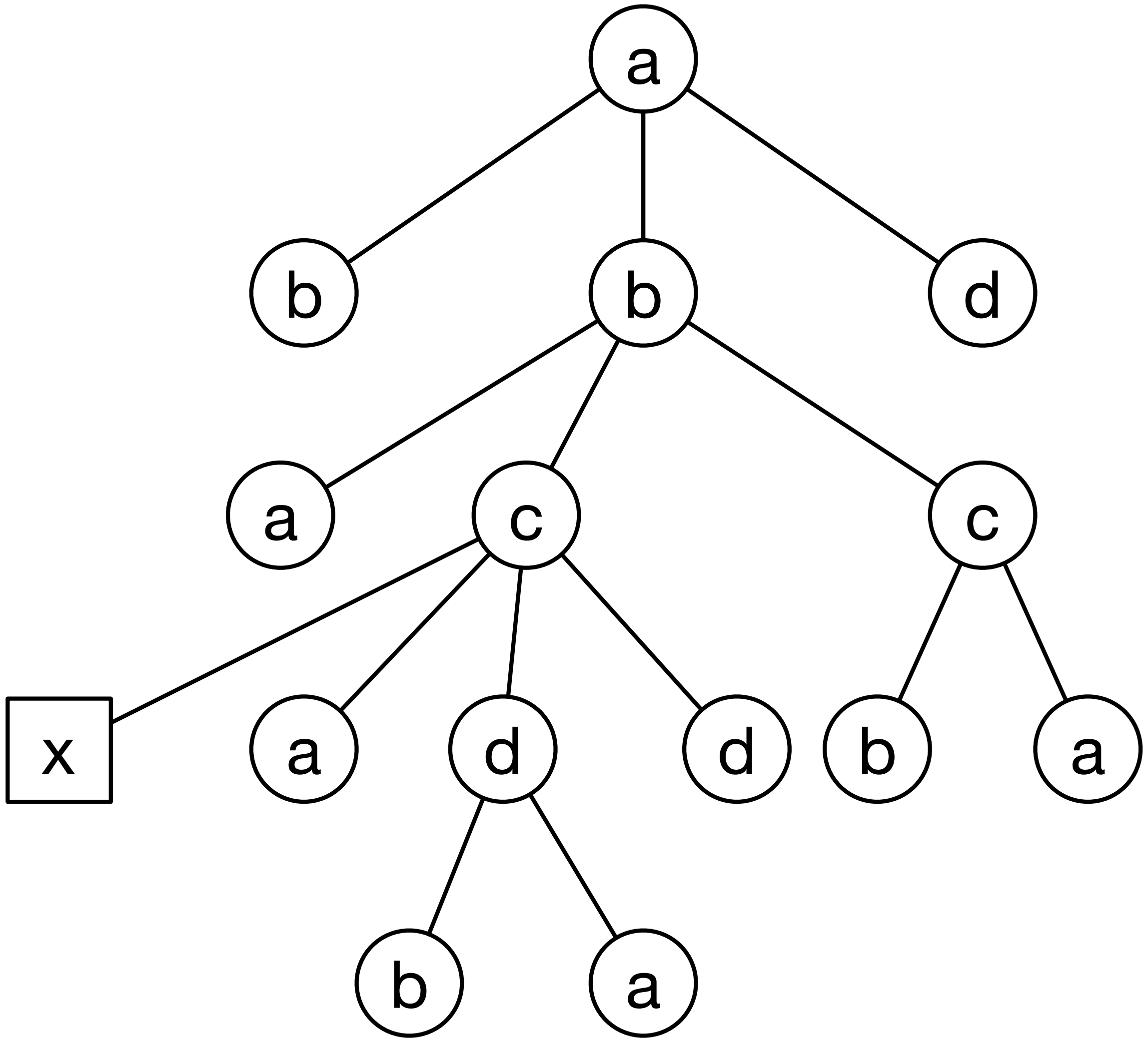
XPath: Examples

```
Q6 := /a/b/c[a and d]
```



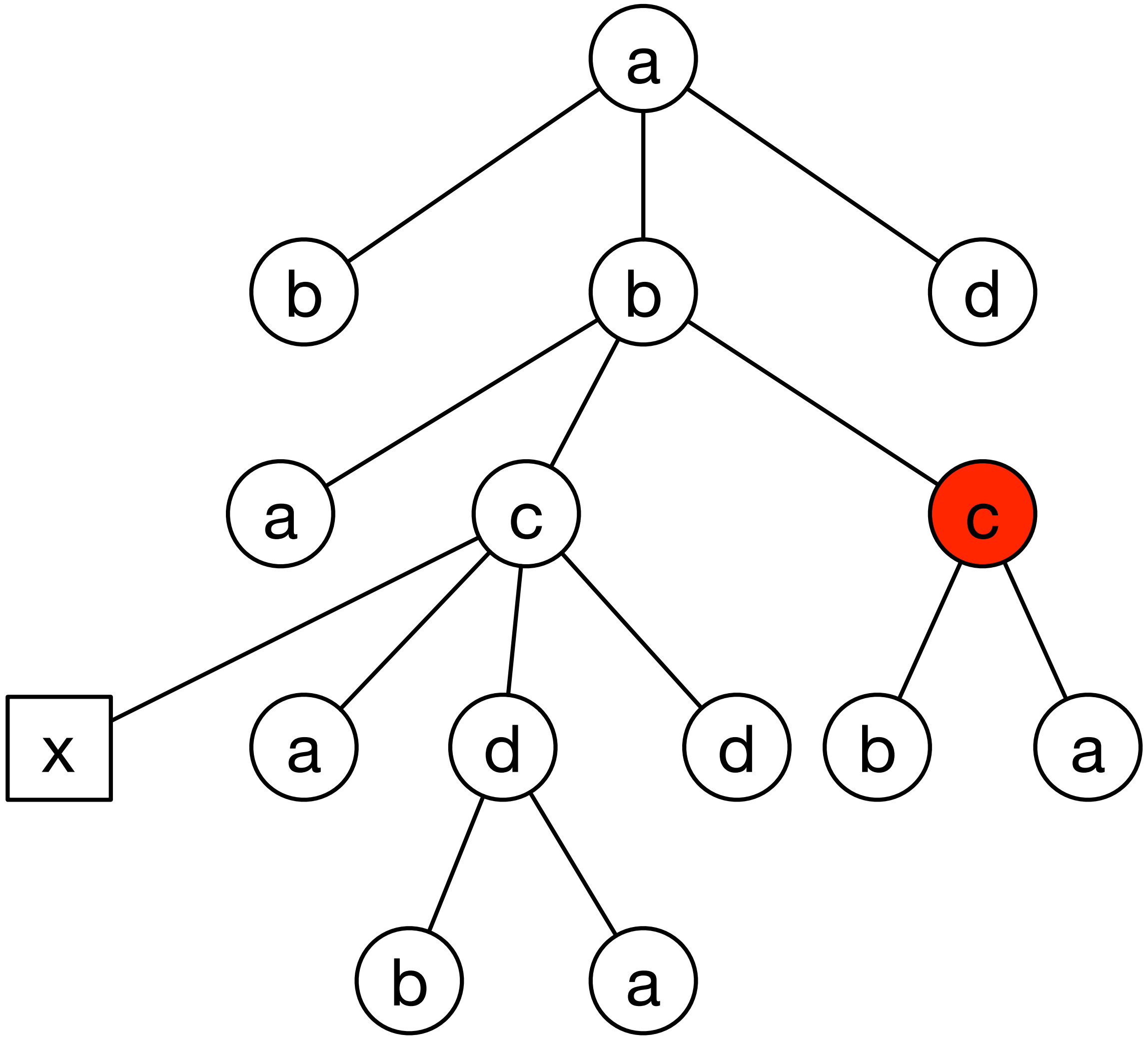
XPath: Examples

Q7 := /a/b/c[not(d)]



XPath: Examples

Q7 := /a/b/c[not(d)]



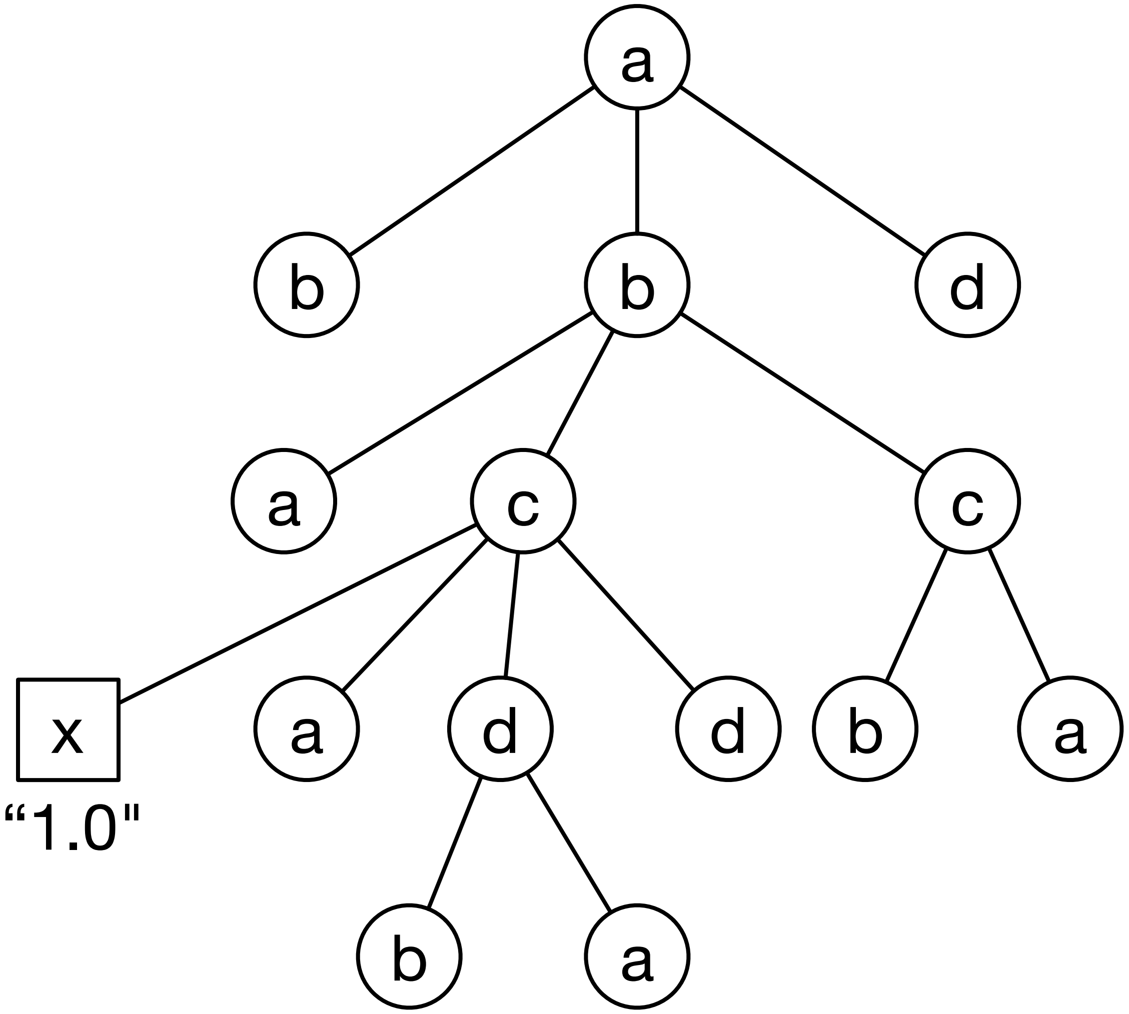
XPath: Predicates

$$\text{axis} :: \text{ntest}[\text{pred}_1] \cdots [\text{pred}_n]$$

Predicates allow testing for attribute values.

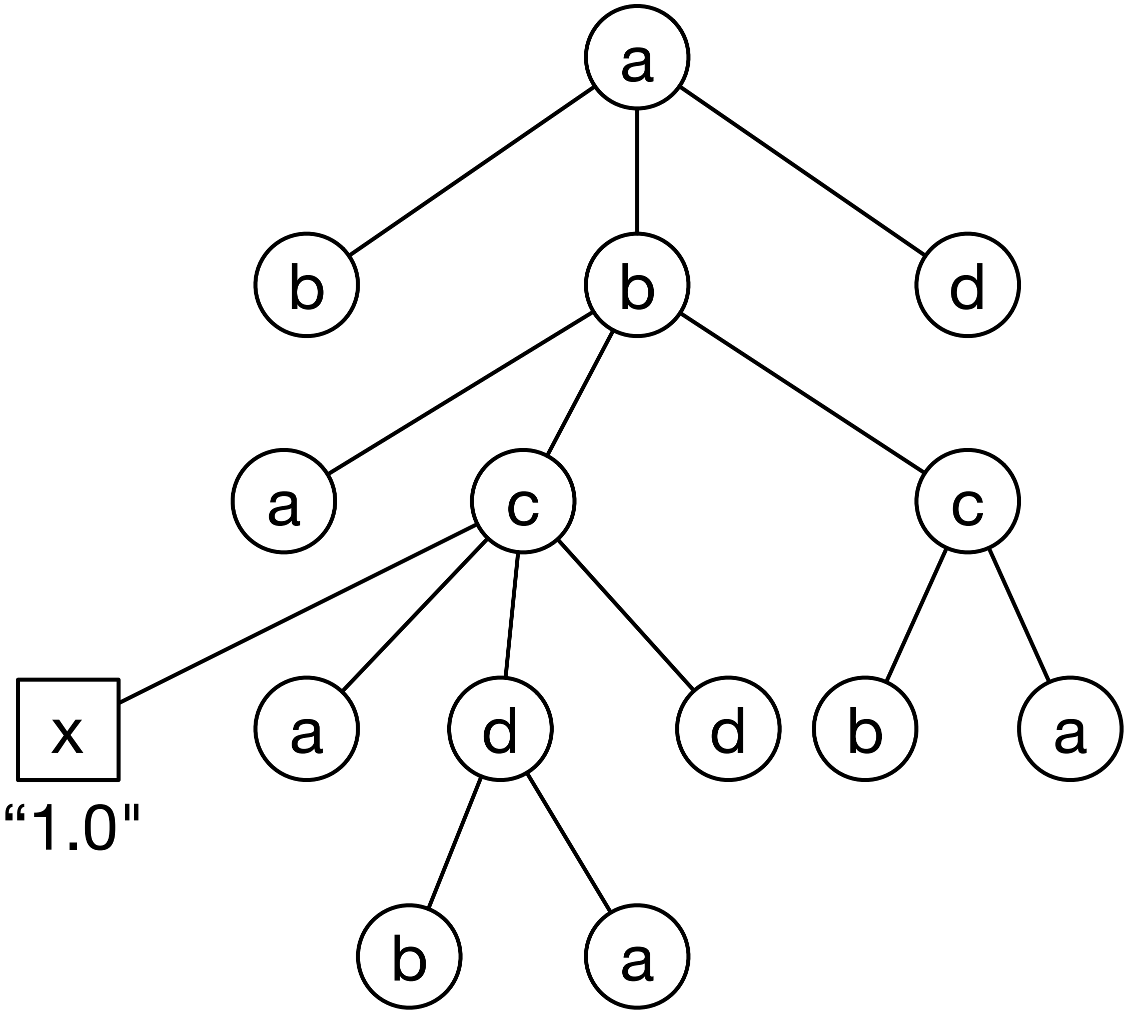
XPath: Examples

```
Q8 := /a/b/c[@x=1]
```



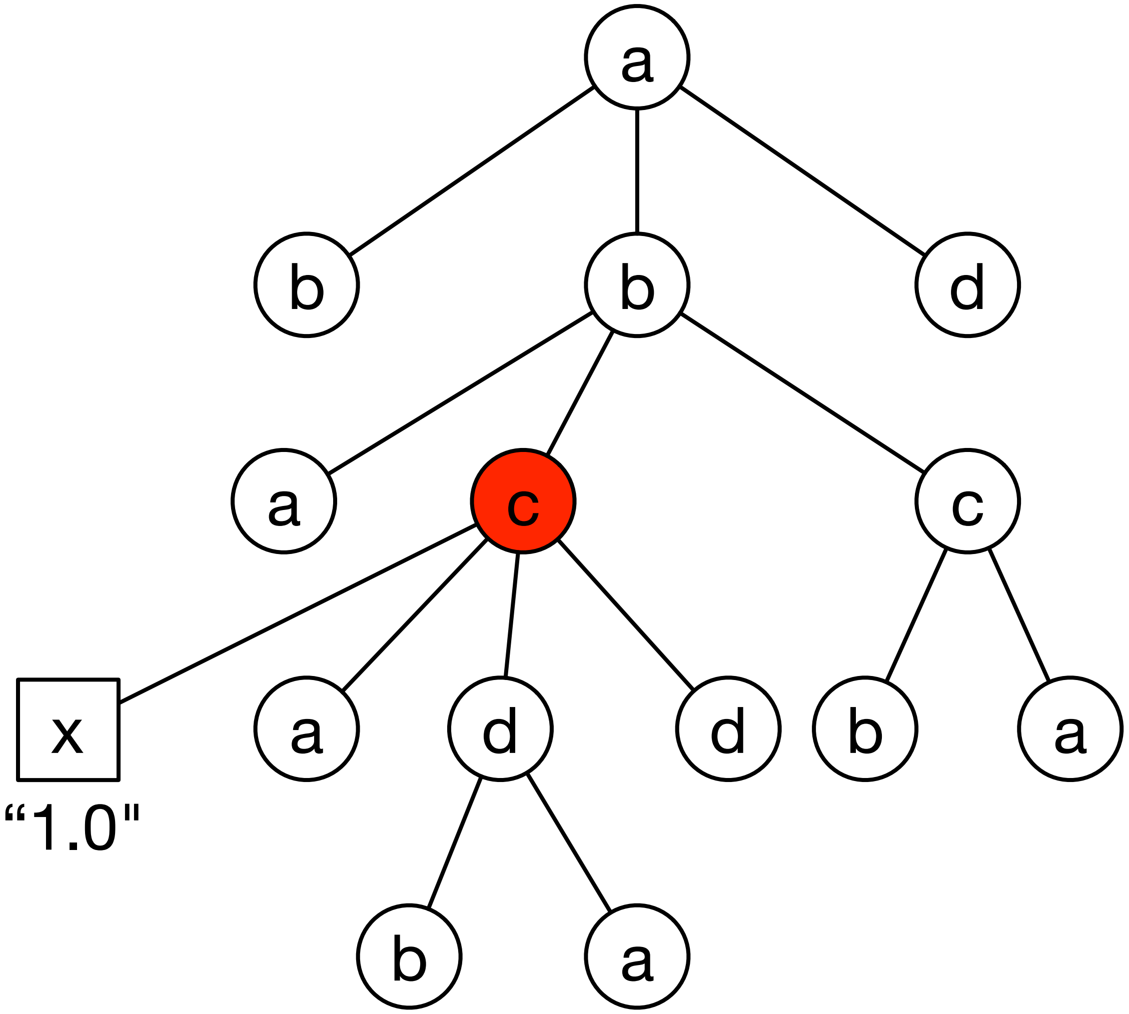
XPath: Examples

```
Q9 := /a/b/c[@x=“1.0”]
```



XPath: Examples

Q9 := /a/b/c[@x=“1.0”]



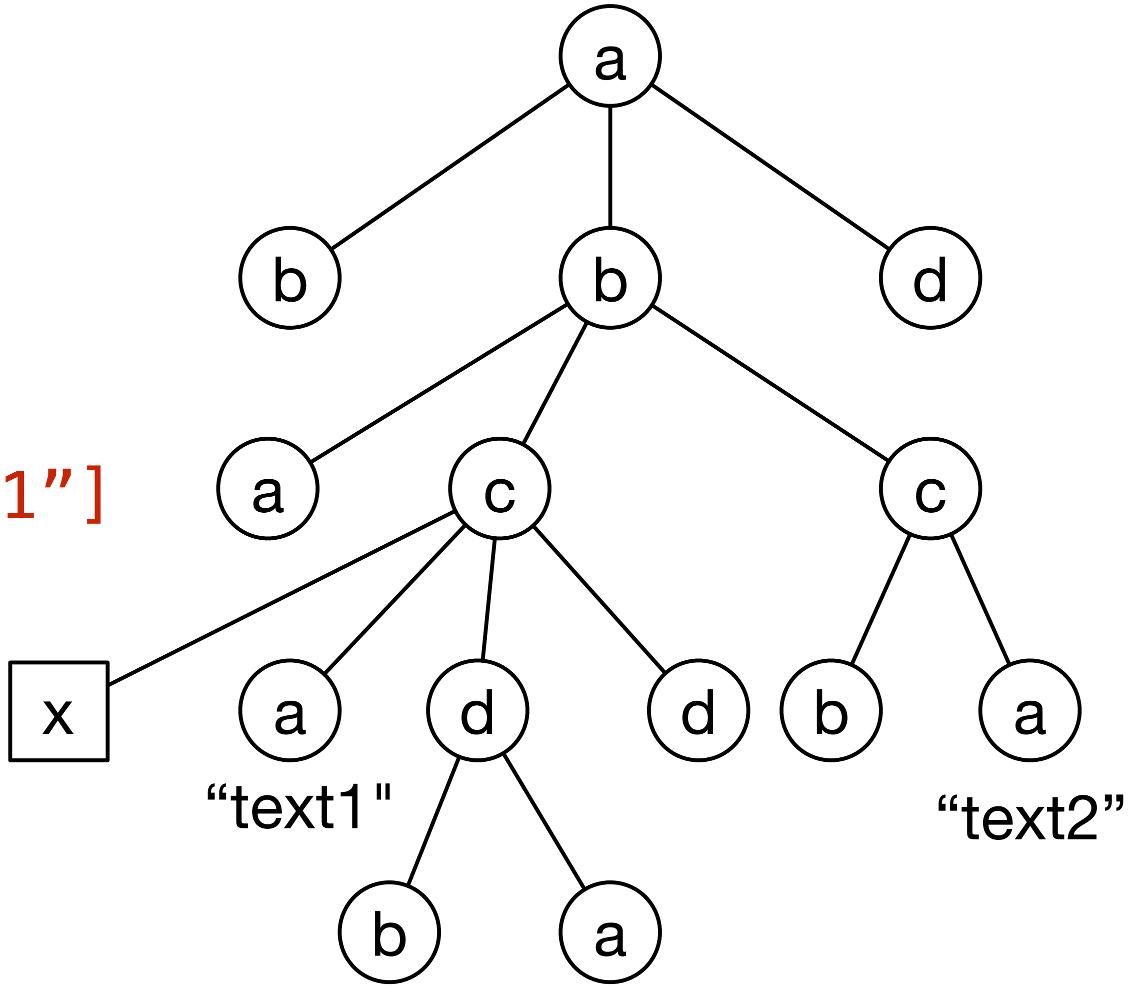
XPath: Predicates

$$\text{axis} :: \text{ntest}[\text{pred}_1]\cdots[\text{pred}_n]$$

Predicates allow testing for text in nodes.

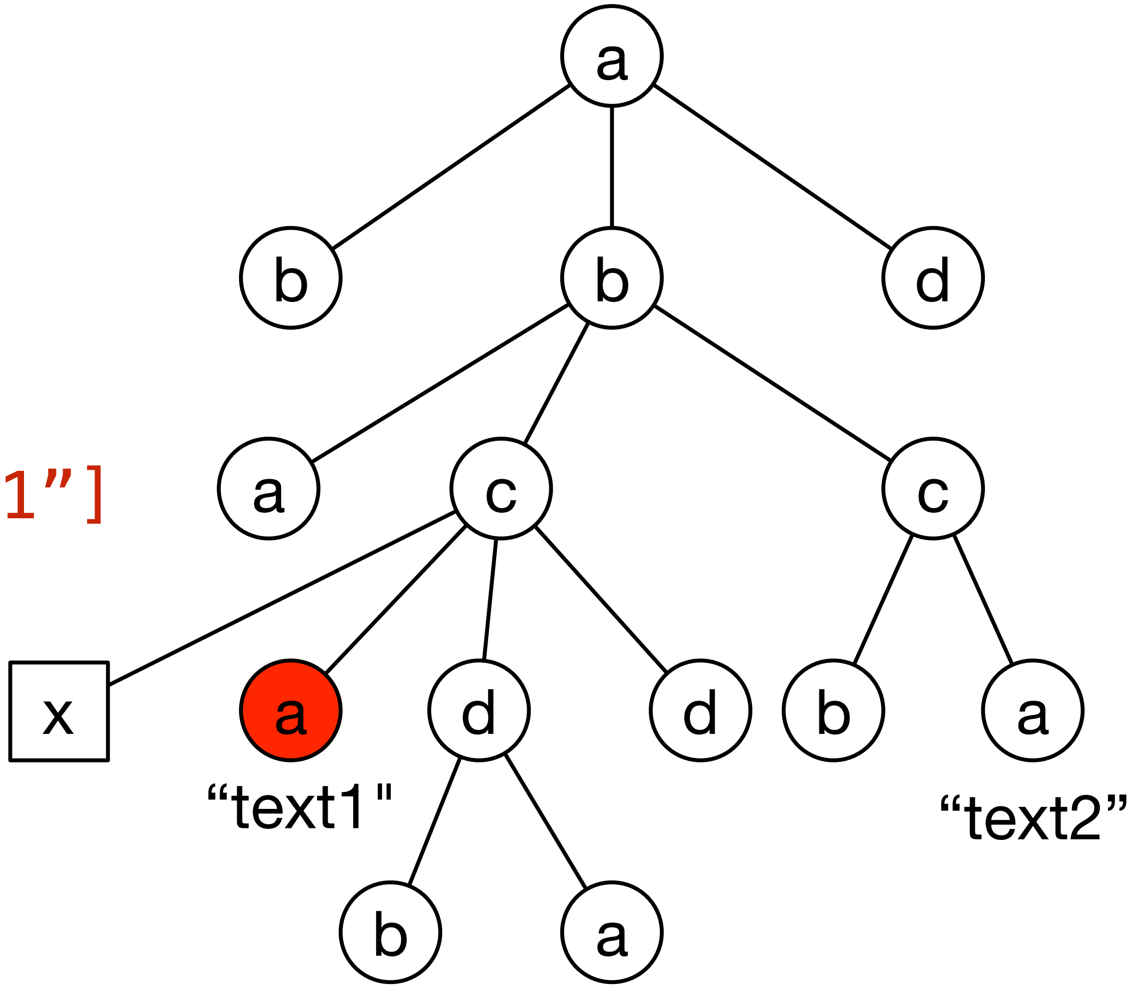
XPath: Examples

```
Q10 := /a/b/c//*[text()=“text1”]
```



XPath: Examples

Q10 := /a/b/c//*[text()=“text1”]



XPath: Useful Predicate Functions

$$\text{axis} :: \text{ntest}[\text{pred}_1] \cdots [\text{pred}_n]$$

Predicates allow functions for testing on node sets:

Function	Semantics
`count(ex)`	counts number of results
`last()`	returns context size from the evaluation context
`position()`	returns context position from the evaluation context
…	

XPath: Useful Predicate Functions

$$\text{axis} :: \text{ntest}\boxed{[\text{pred}_1]\cdots[\text{pred}_n]}$$

Predicates allow functions for testing on strings:

Function	Return value
`concat(s1,…,sn)`	concatenated string
`startswith(str,pre)`	`true()` if `str` starts with `pre`
`contains(str,substr)`	`true()` if `str` contains `substr`
`substring(str,i,j)`	the substring of `str` from `i` to `j`
`stringlength(str)`	the length of `str`
…	

Useful Reading

- XPath Reference http://www.w3.org/TR/xpath/
- March Schol’s slides on XPath http://www.inf.uni-konstanz.de/dbis/teaching/ws0506/database-xml/P8.pdf
- Sebastian Maneth’s slides on XPath http://www.cse.unsw.edu.au/~cs4317/10s1/lectures/06.pdf