

Exercises for the book

Semantic Web Databases

by Sven Groppe

Solutions

Chapter 12 – Comparison of the XML and Semantic Web Worlds

1. The translated XQuery query:

```
declare namespace _='blankNodes';
declare variable $__entry__:=doc('tmpXMLdata.xml')/translatedData/entry;
for $__currentEntry__0 in $__entry__
  let $x := $__currentEntry__0/s/text()
  let $x_datatype := $__currentEntry__0/s_datatype/text()
  let $title := $__currentEntry__0/o/text()
  let $title_datatype := $__currentEntry__0/o_datatype/text()
  for $__currentEntry__1 in $__entry__
    let $author := $__currentEntry__1/o/text()
    let $author_datatype := $__currentEntry__1/o_datatype/text()
    let $__optional0:=
      if(fn:empty(
        for $__currentEntry__2 in $__entry__
          let $price := $__currentEntry__2/o/text()
          let $price_datatype := $__currentEntry__2/o_datatype/text()
          where ($price castable as xs:double) and
            ($price_datatype='http://www.w3.org/2001/XMLSchema#integer'
              or $price_datatype='http://www.w3.org/2001/XMLSchema#double')
            and $price<30
            and $__currentEntry__2[p/text()='http://www.luposdate.org/price']
            and $__currentEntry__2[s/text()=$x]
          return (<binding><var>price</var><value>{$price}</value></binding>)))
      then (<noBinding><var>price</var></noBinding>)
    else(
      for $__currentEntry__2 in $__entry__
        let $price := $__currentEntry__2/o/text()
        let $price_datatype := $__currentEntry__2/o_datatype/text()
        where ($price castable as xs:double) and
          ($price_datatype='http://www.w3.org/2001/XMLSchema#integer'
            or $price_datatype='http://www.w3.org/2001/XMLSchema#double')
          and $price<30
          and $__currentEntry__2[p/text()='http://www.luposdate.org/price']
          and $__currentEntry__2[s/text()=$x]
        return(<binding><var>price</var><value>{$price}</value></binding>))
    where $__currentEntry__0[p/text()='http://purl.org/dc/elements/1.1/title']
      and $__currentEntry__1[p/text()='http://purl.org/dc/elements/1.1/autor']
      and $__currentEntry__1[s/text()=$x]
  return <result>
    <binding><var>title</var><value>{$title}</value></binding>
    { for $__i_3 in (
      $__optional0/var[text()='price']/following-sibling::value/text())
      return <binding><var>price</var><value>{$__i_3}</value></binding>}
    <binding><var>author</var><value>{$author}</value></binding>
  </result>
```

The translated XSLT stylesheet:

```
<xsl:stylesheet version='2.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
  xmlns:xs='http://www.w3.org/2001/XMLSchema'> <xsl:output method='xml'/>
```

```

<xsl:variable name='__entry__' select='/translatedData/entry'/>
<xsl:template match='/>
<results>
  <xsl:for-each select='$__entry__'>
    <xsl:variable name='__currentEntry__0' select='.'/>
    <xsl:variable name='x' select='$__currentEntry__0/s'/>
    <xsl:variable name='x_datatype' select='$__currentEntry__0/s_datatype'/>
    <xsl:variable name='title' select='$__currentEntry__0/o'/>
    <xsl:variable name='title_datatype' select='$__currentEntry__0/o_datatype'/>
    <xsl:for-each select='$__entry__'>
      <xsl:variable name='__currentEntry__1' select='.'/>
      <xsl:variable name='author' select='$__currentEntry__1/o'/>
      <xsl:variable name='author_datatype' select='$__currentEntry__1/o_datatype'/>
      <xsl:variable name='__test__3'>
        <xsl:for-each select='$__entry__'>
          <xsl:variable name='__currentEntry__2' select='.'/>
          <xsl:variable name='price' select='$__currentEntry__2/o'/>
          <xsl:variable name='price_datatype' select='$__currentEntry__2/o_datatype'/>
          <xsl:if test='$__currentEntry__2/s/text()=$x/text() and
            $__currentEntry__2/p/text()="http://www.luposdate.org/price"
            and ($price castable as xs:double) and
            ($price_datatype="http://www.w3.org/2001/XMLSchema#integer"
            or $price_datatype="http://www.w3.org/2001/XMLSchema#double")
            and 30>$price'>
            <binding><var>price</var><value><xsl:value-of select='$price'/></value></binding>
          </xsl:if>
        </xsl:for-each>
      </xsl:variable>
      <xsl:variable name='__optional0'>
        <xsl:choose>
          <xsl:when test='not($__test__3)'>
            <noBinding><var>price</var></noBinding>
          </xsl:when>
          <xsl:otherwise><xsl:copy-of select='$__test__3'/></xsl:otherwise>
        </xsl:choose>
      </xsl:variable>
      <xsl:if test='$__currentEntry__0/p/text()="http://purl.org/dc/elements/1.1/title"
        and $__currentEntry__1/s/text()=$x/text() and
        $__currentEntry__1/p/text()="http://purl.org/dc/elements/1.1/autor"'>
        <result>
          <binding><var>title</var><value><xsl:value-of select='$title'/></value></binding>
          <xsl:for-each select=
            '$__optional0/binding/var[.="price"]/following-sibling::value'>
            <binding><var><xsl:text>price</xsl:text></var>
              <value><xsl:value-of select='.'/></value>
            </binding>
          </xsl:for-each>
          <binding><var>author</var><value><xsl:value-of select='$author'/></value></binding>
        </result>
      </xsl:if>
    </xsl:for-each>
  </xsl:for-each>
</results>
</xsl:template>
</xsl:stylesheet>

```

2. The XML data transformed into RDF data using the proposed numbering scheme and its visual representation:

```

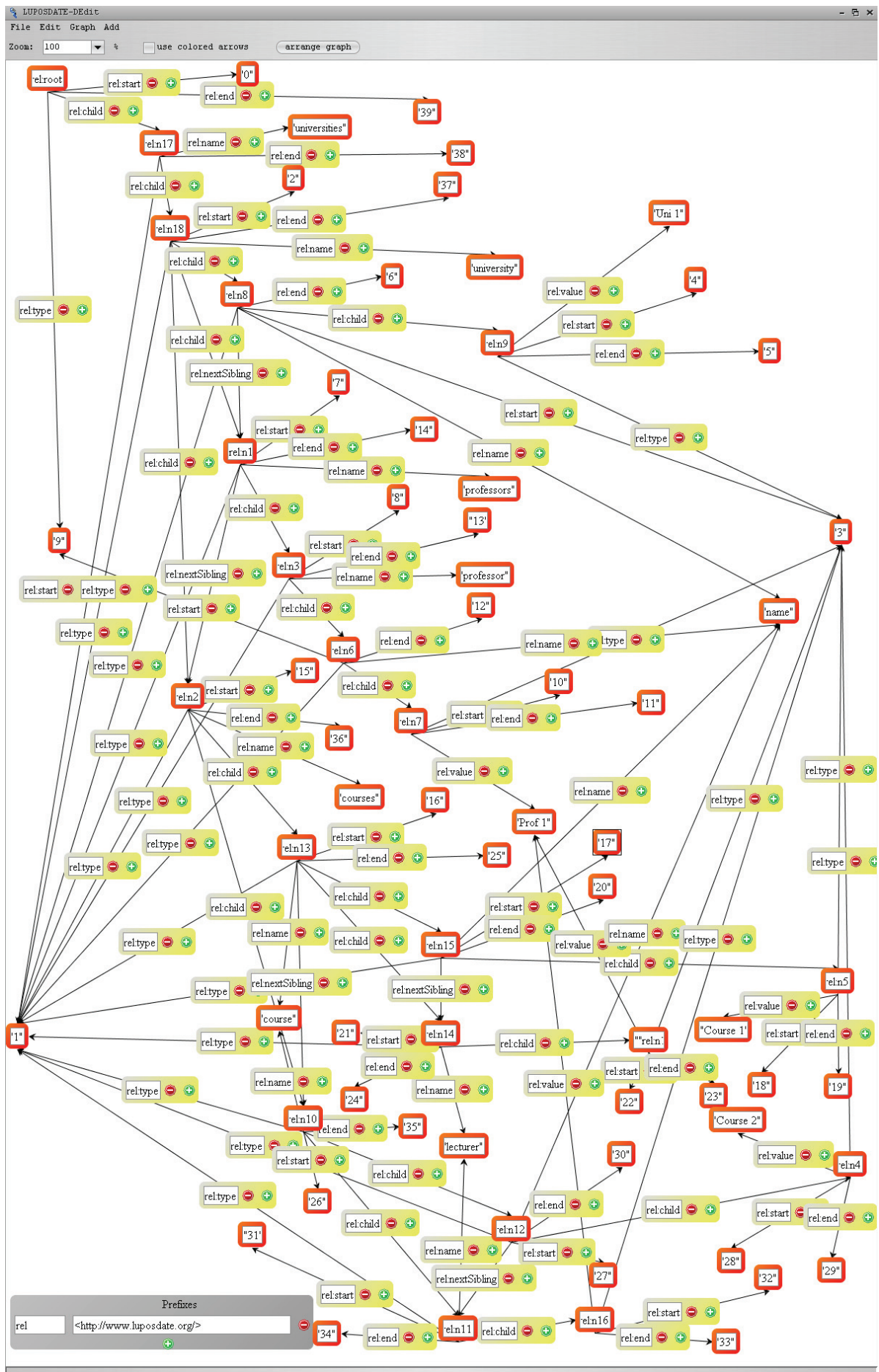
@prefix rel: <http://www.luposdate.org/>.
rel:n1 rel:nextSibling rel:n2.
rel:n1 rel:end "14" .

```

```

rel:n1 rel:child rel:n3 .
rel:n1 rel:start "7" .
rel:n1 rel:type "1" .
rel:n1 rel:name "professors".
rel:n4 rel:end "29" .
rel:n4 rel:type "3" .
rel:n4 rel:value "Course 2" .
rel:n4 rel:start "28" .
rel:n5 rel:end "19" .
rel:n5 rel:type "3" .
rel:n5 rel:value "Course 1" .
rel:n5 rel:start "18" .
rel:n6 rel:end "12" .
rel:n6 rel:child rel:n7 .
rel:n6 rel:start "9" .
rel:n6 rel:type "1" .
rel:n6 rel:name "name" .
rel:n8 rel:nextSibling rel:n1 .
rel:n8 rel:end "6" .
rel:n8 rel:child rel:n9 .
rel:n8 rel:start "3" .
rel:n8 rel:type "1" .
rel:n8 rel:name "name" .
rel:n10 rel:end "35" .
rel:n10 rel:child rel:n11 .
rel:n10 rel:child rel:n12 .
rel:n10 rel:start "26" .
rel:n10 rel:type "1" .
rel:n10 rel:name "course" .
rel:n13 rel:nextSibling rel:n10.
rel:n13 rel:end "25" .
rel:n13 rel:child rel:n14 .
rel:n13 rel:child rel:n15 .
rel:n13 rel:start "16" .
rel:n13 rel:type "1" .
rel:n13 rel:name "course" .
rel:n11 rel:end "34" .
rel:n11 rel:child rel:n16 .
rel:n11 rel:start "31" .
rel:n11 rel:type "1" .
rel:n11 rel:name "lecturer" .
rel:n17 rel:end "38" .
rel:n17 rel:child rel:n18 .
rel:n17 rel:start "1" .
rel:n17 rel:type "1" .
rel:n17 rel:name "universities" .
rel:n14 rel:end "24" .
rel:n14 rel:child rel:n19 .
rel:n14 rel:start "21" .
rel:n14 rel:type "1" .
rel:n14 rel:name "lecturer" .
rel:n7 rel:end "11" .
rel:n7 rel:type "3" .
rel:n7 rel:value "Prof 1" .
rel:n7 rel:start "10" .
rel:n9 rel:end "5" .
rel:n9 rel:type "3" .
rel:n9 rel:value "Uni 1" .
rel:n9 rel:start "4" .
rel:n12 rel:nextSibling rel:n11.
rel:n12 rel:end "30" .
rel:n12 rel:child rel:n4 .
rel:n12 rel:start "27" .
rel:n12 rel:type "1" .
rel:n12 rel:name "name" .
rel:n15 rel:nextSibling rel:n14.

```



The translated SPARQL query:

```
PREFIX cd: <http://www.luposdate.org/>
SELECT DISTINCT ?v10
WHERE {
    ?v1 cd:child ?v2.
    ?v2 cd:name "universities".
    ?v7 cd:start ?v3.
    ?v2 cd:start ?v5.
    ?v7 cd:end ?v4.
    ?v2 cd:end ?v6.
    ?v7 cd:name "professor".
    ?v8 cd:child ?v7.
    ?v9 cd:child ?v8.
    ?v9 cd:child ?v10.
    ?v10 cd:name "courses".
    ?v1 cd:type "9".
    FILTER (?v3 >= ?v5).
    FILTER (?v4 <= ?v6). }
```

The visual representation of the query:

