

Web-Informationssysteme, WS 2009/10
Übungsblatt 10

Besprechung am Di 19.01.2010

Aufgabe 10-1 XPath, ein steiniger Pfad

Seien p_1 und p_2 XPath-Ausdrücke, deren Auswertung je eine Knotenmenge liefert. Die einzige Mengenoperation, die XPath 1.0 dafür anbietet, ist $p_1 \mid p_2$ für die Vereinigung. Oft wäre es aber nützlich, auch Mengenoperationen wie Schnitt und Differenz zur Verfügung zu haben.

- Wie kann man mit XPath 1.0 testen, ob p_1 eine **Teilmenge** der Knotenmenge von p_2 auswählt?
- Wie kann man mit XPath 1.0 den **Schnitt** zwischen den Knotenmengen berechnen, die durch p_1 und p_2 beschrieben sind?
- Wie kann man mit XPath 1.0 die **Mengendifferenz** zwischen den Knotenmengen berechnen, die durch p_1 und p_2 beschrieben sind?

Das folgende XML-Dokument repräsentiert einen Ausschnitt aus einer Netzwerkdarstellung. Die *id*-Attribute müssen vom Typ *ID* und die *links*-Attribute vom Typ *IDREFS* sein.

```
<!DOCTYPE net SYSTEM "net.dtd">
2 <net>
  <vertex id="n0" links="n1 n3" />
4  <vertex id="n1" links="n0 n2 n4" />
  <vertex id="n2" links="n1 n4" />
6  <vertex id="n3" links="n0 n4" />
  <vertex id="n4" links="n2 n3" />
8  <vertex id="n5" links="" />
</net>
```

Geben Sie XPath-Ausdrücke an, die die folgenden Knotenmengen selektieren:

- alle vertex-Elemente, die von dem vertex-Element mit Attributwert 'n0' über eine Referenz erreichbar sind.
- alle vertex-Elemente, die eine Referenz auf sich selbst haben.
- alle vertex-Elemente, die nicht referenziert werden.

Aufgabe 10-2 XPath: Zuordnung von Knotenmengen

Betrachten Sie das XML-Dokument `book.xml` in der Anlage.

Wir betrachten folgende Knotenmengen, wobei p_i für das p -Element in Zeile i des obigen Dokuments steht.

$$K_1 = \emptyset \quad K_2 = \{p_{13}\} \quad K_3 = \{p_{34}\} \quad K_4 = \{p_{22}, p_{34}\} \quad K_5 = \{p_{13}, p_{22}, p_{34}\}$$

Geben Sie für jeden der folgenden Pfadausdrücke an, welche dieser Knotenmengen sein Ergebnis auf dem obigen Dokument ist:

```

A1: /descendant::section/child::p[ position()=2 ]
A2: /descendant::section/descendant::p[ position()=2 ]
A3: /descendant::section//p[ position()=2 ]
A4: (/descendant::section/descendant::p)[ position()=2 ]
A5: (/descendant::section/child::p ) [ position()=2 ]
A6: /descendant::section/child::p[ contains(self::node(),"SSD") ][ position()=2 ]
A7: /descendant::section/child::p[ position()=2 ][ contains(self::node(),"SSD") ]

```

Aufgabe 10-3 XQuery: Summieren

Gegeben sei eine Matrix natürlicher Zahlen wie in der Datei `matrix.xml`, die nicht notwendigerweise quadratisch sein muss, aber zumindest regulär (d.h. gleich viele Zellen pro Zeile, und keine leeren Zellen).

matrix.xml

```

<?xml version="1.0"?>
<matrix>
  <row>
    <cell>1</cell>
    <cell>2</cell>
    <cell>3</cell>

  </row>
  <row>
    <cell>4</cell>
    <cell>5</cell>
    <cell>6</cell>

  </row>
  <row>
    <cell>7</cell>
    <cell>8</cell>
    <cell>9</cell>

  </row>
</matrix>

```

matrix-summed-up.xml

```

<?xml version="1.0"?>
<matrix>
  <row>
    <cell>1</cell>
    <cell>2</cell>
    <cell>3</cell>
    <cell>6</cell>

  </row>
  <row>
    <cell>4</cell>
    <cell>5</cell>
    <cell>6</cell>
    <cell>15</cell>

  </row>
  <row>
    <cell>7</cell>
    <cell>8</cell>
    <cell>9</cell>
    <cell>24</cell>

  </row>
  <row>
    <cell>12</cell>
    <cell>15</cell>
    <cell>18</cell>
    <cell>45</cell>

  </row>
</matrix>

```

Schreiben Sie ein XQuery-Programm, welches zu jeder Zeile und zu jeder Spalte die jeweilige Summe hinzufügt, und schließlich noch die Summe aller Zellen in die letzte Zelle der letzten Zeile hinzufügt. Das Ergebnis auf der Beispieldatei sollte wie `matrix-summed-up.xml` aussehen.