

Markup-Sprachen und semi-strukturierte Daten, SS 2008
Übungsblatt 9

Abgabe: bis Do, 26.06.2007, 14 Uhr

Besprechung am Do/Fr 26./27.06.2008

Aufgabe 9-1 schriftlich bearbeiten
XQuery: Ungleiche Gleichheiten

XQuery bietet verschiedene Möglichkeiten Gleichheit zu testen: die Operatoren `eq`, `=` und `is`, sowie die (boolsche) Funktion `fn:deep-equal`.

- a) Beschreiben Sie in die einzelnen Operatoren in jeweils zwei bis drei Sätzen und heben dabei ihre Unterschiede hervor.

Welche Ausgabe bzw. Fehler erzeugen jeweils die folgenden XQuery Programme? Begründen Sie jeweils Ihre Antwort kurz. (Es tritt nur die Ziffer 0 auf, nie der Großbuchstabe 0)

b)

```
let $a := (<bond><james>00</james><bond>7</bond></bond>,  
          <agent>006</agent>,  
          <nummer>7</nummer>,  
          <james_bond>007</james_bond>,  
          "007")  
for $b in $a  
return <result>{$b eq <blond>007</blond>}</result>
```

c)

```
for $sieben in ( 007, 7, 7.0, <num>7</num> )  
return <result>{$$sieben eq 007}</result>
```

d)

```
let $sieben := ( 007, 7, 7.0, <num>7</num> )  
return <result>{$$sieben eq 007}</result>
```

e)

```
for $a in (<a>7</a>, <b>7.0</b>, <c>6.9999</c>, <d>007</d>)  
return <result>{ $a = 7.0 }</result>
```

f)

```
let $a := (<a>7</a>, <b>7.0</b>, <c>seven</c>, <d>007</d>)  
for $b in $a, $c in $a  
return <result>{$a = $b}, {$b = $c}</result>
```

g)

```
let $a := <a> <b>7</b> <c>8</c> <b>7</b> </a>  
let $b1 := $a/b[1]  
for $b in $a/*  
return <result>{$b1 is $b}</result>
```

h)

```
let $a := <a> <b>7</b> <c>8</c> <b>7</b> </a>  
let $b1 := $a/b[1]  
for $b in $a/*  
return <result>{fn:deep-equal($b1,$b)}</result>
```

Aufgabe 9-2 schriftlich bearbeiten
Xcerpt: Anfrage-Terme

Gegeben sei ein Xcerpt-Anfrageterm t_q und ein Datenterm t_d wie folgt:

$t_d = e [c, d\{a,b\}, b, d\{b,c\}, d\{d\{b\},b\}, d\{b\}, d\{a,d,d\{b\}\}]$

$t_q = e [[\text{var } Y, \text{var } X \rightarrow d\{ \text{var } Y, \text{var } Z \}]]$

Welche Belegungen können die Variablen x , y und z haben? (Beachten Sie, dass die Belegung der Variablen voneinander abhängig sind, man bekommt also eine Menge von Variablenbelegungen als Ergebnis)

Aufgabe 9-3 **Xcerpt: Regelauswertung**

Gegeben sei folgende Xcerpt-Regel (der Anfrageteil ist der gleiche wie in der vorherigen Aufgabe):

```
CONSTRUCT
  results[ all r [ var X, all s [ var Y, "abc", var Z ] ]
FROM
  e [ [ var Y, var X -> d{ var Y, var Z } ] ]
END
```

Welches Ergebnis konstruiert diese Regel mit dem Datenterm der vorherigen Aufgabe als Eingabe, also mit der oben ermittelten Menge von Variablenbelegungen?

Aufgabe 9-4 **XQuery: Numeriertes Inhaltsverzeichnis**

Gegeben ist das von Übungsblatt 7 bekannte XML-Dokument `report.xml`, das verschachtelte Abschnitte (`<section>`) enthält, die jeweils einen Titel (`<title>`) und etwas Text haben (sowie natürlich Unterabschnitte).

Schreiben Sie ein XQuery-Programm `report-toc.xq`, das für das gegebene Dokument ein Inhaltsverzeichnis erstellt. Dabei sollen die Überschriften nach ihrer Position und Tiefe im Dokument numeriert werden, also z.B. “1”, “1.1”, “1.1.1”, ..., “1.2”, “1.2.1”, Allerdings verwenden wir statt dem Punkt als Trennzeichen einzelne `num`-Elemente, also z.B.

```
<num>1</num>,
<num>1</num><num>1</num>,
<num>1</num><num>1</num><num>1</num>, ...
<num>1</num><num>2</num>,
<num>1</num><num>2</num><num>1</num>, ...
```

Die Datei `report-toc.xml` gibt ein Beispiel, wie das Inhaltsverzeichnis aussehen soll.

Hinweise: Das XQuery-Konstrukt “at” in FOR-Schleifen hilft Ihnen bei der Erzeugung der Numerierung. Ferner bietet es sich an mit rekursiven Funktionen zu arbeiten, damit das Dokument mit beliebigen Schachtelungstiefen bearbeitet werden können. (Achtung: im Gegensatz zur Aufgabe 7-1 soll hier *nicht* auf der dritten “abgebrochen” werden!)

Aufgabe 9-5 **XQuery: Fußball-Tabelle**

Gegeben ist das XML-Dokument `em.xml`, das für teilnehmende Mannschaften an den Gruppenspielen eines Fußball-Turniers ihre erreichten Punkte, Tore und Gegentore enthält. Die Gruppe eines Teams wird im Attribut “gruppe”, der Name des Teams im Attribut “name” angegeben.

Schreiben Sie eine XQuery-Funktion **ergebnis**, welche die Gruppentabelle — mit den Mannschaften in Reihenfolge ihrer Platzierungen — errechnet. Als Parameter erhält die Funktion das Root-Element (“em”) des Dokuments, sowie und einen String, der angibt, für welche Gruppe (“A”, “B”, etc.) das Ergebnis berechnet werden soll.

Die Platzierungen der Mannschaften werden dabei durch folgende (vereinfachte) Regeln bestimmt:

- Je höher die Punktezahl desto besser die Platzierung.
- Bei Punktegleichstand zählt die Differenz von Toren und Gegentoren (je größer, desto besser die Platzierung).
- Bei gleicher Tordifferenz zählt die Anzahl der (erzielten) Tore (je größer, desto besser die Platzierung).

Erweitern Sie ihr XQuery-Programm, so daß die Funktion **ergebnis** für die Gruppe “B” aufgerufen wird. Ein mögliches Ergebnis ist mit der Datei **em-ergebnis-b.xml** gegeben.