

Markup-Sprachen und semi-strukturierte Daten, SS 2008  
Übungsblatt 8

Abgabe: bis Do, 19.06.2007, 14 Uhr  
Besprechung am Do/Fr 19./20.06.2008

**Aufgabe 8-1** schriftlich bearbeiten  
**XQuery: Verständnis**

Geben Sie jeweils an, ob die folgenden XQuery Programme gültig sind. Geben Sie für gültige Programme zusätzlich das Ergebnis an, für nicht gültige Programme eine Begründung der Ungültigkeit.

a) `<result>for $i in (1,2,3) return $i</result>`

b) `<result>{ for $a in (1,2,3) return $a/position() }</result>`

c) `<result> {  
 let $list := (for $a in (1,2,3) return <a>{$a}</a>) return  
 for $a1 in $list return $a1/position()  
} </result>`

d) `<ergebnis> {  
 let $a := (<a>1</a>,<a>2</a>,<a>3</a>) return $a/position()  
} </ergebnis>`

e) `<ergebnis> {  
 let $a := (<a>1</a>,<a>2</a>,<a>3</a>) return $a/last()  
} </ergebnis>`

f) `<ergebnis> {  
 let $a := (<a>1</a>,<a>2</a>,<a>3</a>) return  
 ($a/last(), $a/position()) } </ergebnis>`

g) `<ergebnis> {  
 let $list := for $i in (1,2,3) return <i>{ $i }</i>  
 return ($list, $i) } </ergebnis>`

h) `<ergebnis> {  
 let $a := (<a>1</a>,<a>2</a>,<a>3</a>) return <c>{ $a/position() }</c>  
} </ergebnis>`

### Aufgabe 8-2 schriftlich bearbeiten

#### XQuery: spaltenweise aufgebaute Tabelle transformieren

Gegeben sei nochmal eine spaltenweise aufgebaute Tabelle wie in Aufgabe 7-4 mit den folgenden Element-Deklarationen für *spaltenweise* aufgebaute Tabellen:

```
<!ELEMENT tabelle      (spalten) >
<!ELEMENT spalten      (spalte*) >
<!ELEMENT spalte       (zelle*)  >
<!ELEMENT zelle        (#PCDATA) >
```

Die Datei `spalten.xml` enthält ein XML-Dokument, das bzgl. der obigen DTD valide ist (mit Wurzel `tabelle`).

Schreiben Sie ein XQuery-Programm `spiegeln-loesung.xq`, das eine derart aufgebaute Tabelle in eine HTML-Tabelle transformiert (für das Ergebnis siehe `zeilen.xml`).

Wie in Aufgabe 7-4 dürfen Sie dabei annehmen, dass die Tabelle im Eingabedokument keine Unregelmäßigkeiten hat, also keine Zellen “fehlen”. In anderen Worten: Die `spalte`-Elemente haben alle die gleiche Anzahl an `zelle`-Kindelementen.

### Aufgabe 8-3 XQuery: Summieren der Zeilen und Spalten einer Matrix

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

### Aufgabe 8-4 XQuery

Geben Sie eine Anfrage an, die für die XML-Datei `report.xml` als Ergebnis die Instrumente (`instrument`), die in den ersten zwei Aktionen (`action`) nach dem zweiten Einschnitt (`incision`) benutzt wurden, zurückliefert.