

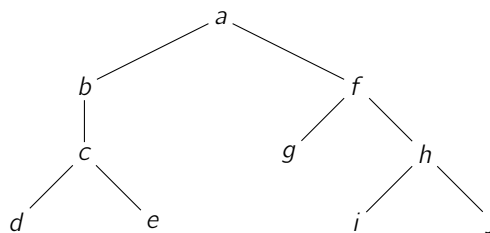


Übungen zur Vorlesung
“Database-Supported XML Processors”
Wintersemester 2008/2009
Jan Rittinger (jan.rittinger@uni-tuebingen.de)

11. Übungsblatt

Ausgabe: 27. Januar 2009 · Besprechung: 3. Februar 2009

Aufgabe 1: XQuery



Baumstrukturen erlauben uns mehrere sinnvolle Reihenfolgen, alle Knoten des Baumes zu besuchen/auszugeben:

- Die *preorder* beschreibt eine Tiefensuche des Baumes von links nach rechts. Dabei wird ein Knoten besucht *bevor* alle seine Kinder besucht werden. Die *preorder* des obigen Baumes wäre *a, b, c, d, e, f, g, h, i, j*.
- Das Gegenstück dazu ist die *postorder*. Ebenfalls in einer Tiefensuche werden vor dem Besuchen eines Knotens alle seine Kinder besucht. Die *postorder* des obigen Baumes wäre *d, e, c, b, g, i, j, h, f, a*.
- In einer *Breitensuche* werden alle Knoten auf gleicher Ebene besucht, bevor die nächsttiefere Ebene besucht wird; für den obigen Baum ergäbe sich: *a, b, f, c, g, h, d, e, i, j*.

Schreiben Sie XQuery-Anfragen (ohne das Konzept der *User-Defined Functions* zu nutzen), die alle Elemente eines XML-Dokuments ausgeben

1. in *preorder*,
2. in umgekehrter *preorder*,
3. in *postorder*,
4. in der Reihenfolge einer *Breitensuche*.

Hinweis: Denken Sie bitte über die Verwendung von `count()` zum Lösen der Aufgabe nach.

Aufgabe 2: XPath Accelerator

In der Vorlesung wurde eine systematische Übersetzung von XPath-Anfragen nach SQL besprochen (Folie 348). Übersetzen Sie nach diesem Schema folgende Anfragen:

1. `/descendant::a/parent::b`
2. `/descendant-or-self::b[child::a]`

Normalisieren (entschachteln) Sie anschließend die Anfragen vollständig (siehe auch Folien 349 und 351).

In der Vorlesung wird auch die symmetrische Äquivalenz von XPath-Anfragen besprochen (z. B. Folie 394). Die obigen Anfragen sind ein weiteres Beispiel für solche äquivalenten Anfragen. Können Sie diese Äquivalenz anhand Ihrer SQL-Anfragen beweisen?