

Markup-Sprachen und semi-strukturierte Daten, SS 2008
Übungsblatt 7

Abgabe: bis Do, 12.06.2007, 14 Uhr

Besprechung am Do/Fr 12./13.06.2008

Hinweis: Die angegebenen Dateien befinden sich wie immer unter "Material" bei den Übungsblättern auf der Homepage. Es wird dringend empfohlen Ihre XSLT-Stylesheets mit einem Prozessor wie z.B. `xsltproc` zu testen.

Aufgabe 7-1 schriftlich bearbeiten
XSLT: Pattern Matching

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

Schreiben Sie ein XSLT Stylesheet `report-html.xsl`, welches das gegebene XML-Dokument zur Darstellung in XHTML wie das angegebene `report-html.xml` umwandelt. Dabei soll wie folgt verfahren werden:

- Der Titel des äußersten Abschnitts (Schachtelungstiefe 1) wird in einem `<h1>`-Element gesetzt. Außerdem kommt er in das `<title>` im `<head>`.
- Die Titel auf Schachtelungstiefe 2 werden mit `<h2>`-Elementen gesetzt.
- Die Titel mit Schachtelungstiefe 3 oder größer mit `<h3>`-Elementen gesetzt.

Bemerkung zur Ausgabe: Sie sollten ihrem XSLT-Stylesheet folgendes hinzufügen um eine DOCTYPE-Deklaration für XHTML zu erhalten:

```
<xsl:output
  method="xml"
  encoding="iso-8859-1"
  doctype-public="-//W3C//DTD XHTML 1.0 Transitional//EN"
  doctype-system="http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"
/>
```

Aufgabe 7-2 schriftlich bearbeiten
XSLT: Verschiedene Transformationen

Gegeben ist das XML-Dokument `books.xml`, das Informationen über Bücher enthält. Schreiben Sie die folgenden XSLT Stylesheets, die `books.xml` wie angegeben transformieren:

- a) `books-under50.xsl`: erzeugt eine Liste der Bücher (einfach Kopien der jeweiligen `<book>`-Elemente) mit einem Preis unter 50. Das erwartete Resultat für `books.xml` ist als `books-under50.xml` gegeben.

- b) **books-avg.xsl**: erzeugt eine Liste mit allen Herausgebern und den jeweiligen Durchschnittspreisen für ihre Bücher. Das erwartete Resultat für **books.xml** ist als **books-avg.xml** gegeben.
- c) **books-1-2.xsl**: gibt alle Elemente zwischen dem **book**-Element mit Attributwert **no="1"** und dem **book**-Element mit Attributwert **no="2"** aus.

Aufgabe 7-3 XSLT: Maximale Tiefe des Dokumentbaums

Schreiben Sie ein XSLT Stylesheet, das die maximale Tiefe des Dokumentbaums des Eingabedokuments berechnet. (Attribute und Textknoten werden nicht gerechnet.) Als Resultat soll ein Dokument

`<maxdepth> Integer </maxdepth>`

herauskommen, wobei *Integer* die Tiefe ist. Zur Kontrolle: **books.xml** aus der vorherigen Aufgabe hat die Tiefe 4, **maxdepth1.xml** hat die Tiefe 1.

Aufgabe 7-4 XSLT: spaltenweise aufgebaute Tabelle transformieren

Gegeben seien folgende 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 XSLT-Stylesheet **spalten-zeilen.xslt**, das eine derart aufgebaute Tabelle in eine HTML-Tabelle transformiert (für das Ergebnis siehe **spalten-zeilen.xml**).

Sie dürfen 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 7-5 XSLT: Rekursion über Templates

Implementieren Sie die Fakultätsfunktion in XSLT. Das Eingabedokument soll das Format

`<fac> Integer </fac>`

haben, das Ausgabedokument das Format

`<result> Integer </result>`