Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

21. Februar 2017

Vorlesung "XML und Datenbanken"
— Klausur —

Name: ______________________________

Matrikelnummer: ______________________________

Studiengang: ______________________________

Aufgabe	Punkte	von	Zeit
1 (DTD)		10	15 min
2 (DTDs vs. XML Schema)		3	5 min
3 (XML Schema)		12+2	20 min
4 (Software, Literatur)		3	5 min
5 (XPath)		9	20 min
6 (XQuery)		8	15 min
7 (XSLT)		6	10 min
Summe		51	90 min

- Ich fühle mich gesundheitlich in der Lage, diese Prüfung abzulegen. (Bitte sprechen Sie mit dem Aufsichtspersonal, falls Sie sich krank fühlen. Eine Krankmeldung muss vor Ende der Prüfung erfolgen.)
- Ich erkläre, dass ich diese Prüfung nicht bereits endgültig nicht bestanden habe.

Unterschrift: ______________________________

Hinweise:

- Bearbeitungsdauer: 90 Minuten
- Es dürfen drei DIN A4-Blätter mit Notizen verwendet werden (Vorder- und Rückseite, also 6 Seiten, auch gedruckt). Weitergehene schriftliche Unterlagen (z.B. Bücher, Skripte), Notebooks, PDAs, etc. dürfen nicht verwendet werden. Mobiltelefone bitte ausschalten (oder mit Aufsicht besprechen).
- Die Klausur hat 13 Seiten. Bitte prüfen Sie die Vollständigkeit.
- Bitte benutzen Sie den vorgegebenen Platz. Wenn Sie auf die Rückseite oder ein Zusatzblatt ausweichen müssen, markieren Sie klar, dass es eine Fortsetzung gibt.
- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Versuchen Sie, alle Aufgaben zu bearbeiten. Wenn Sie nichts hinschreiben, haben Sie den Punkt für die jeweilige Teilaufgabe auf jeden Fall verloren.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!
- Schreiben Sie lesbar! Verwenden Sie keinen roten Stift. Bleistift ist nicht verboten, aber problematisch, wenn Sie sich nach der Klausureinsicht beschweren wollen.
- Mit 60% der Punkte haben Sie garantiert bestanden, die Grenze wird ggf. gesenkt.
- Sie dürfen vor 11^{30} nicht gehen, danach nicht mehr auf Toilette.

Aufgabe 1 (DTD) 10 Punkte

Die folgenden Daten sind für ein rudimentäres Textadventure-Spiel gedacht. In solchen Spielen kann man sich u.a. durch Angabe von Himmelsrichtungen (N, O, S, W) durch einen gerichteten Graphen aus "Räumen" bewegen, das sind Positionen in der Spielwelt (auch eine Lichtung im Wald zählt als "Raum"). Man kann Objekte finden und nehmen, und verschiedene Aktionen durchführen. Entwerfen Sie eine DTD, bezüglich der die folgende XML-Datei validiert werden kann. Beachten Sie auch die Anforderungen auf der nächsten Seite.

```
<Spiel>
  <Raum ID="Lichtung">
    <Text>Du bist auf einer Lichtung im Wald.</Text>
    <Weg Richtung="N" nach="Waldweg">Ein Weg führt nach Norden.</Weg>
    <Weg Richtung="S" nach="Lichtung">Im Süden ist ein Trampelpfad,
                                      der eine Biegung macht.</Weg>
  </Raum>

  <Raum ID="Waldweg">
    <Text>Du bist auf einem Waldweg.</Text>
    <Objekt ID="Pilze" Flags="giftig">Rote Pilze mit weissen Tupfen</Objekt>
    <Weg Richtung="N" nach="WegImMoor">Der Weg führt weiter
                                       nach Norden.</Weg>
    <Weg Richtung="O" nach="Hütte">Im Osten ist eine Hütte.
                                   Die Tür ist offen.</Weg>
  </Raum>

  <Raum ID="Hütte">
    <Text>Du bist in einer kleinen Holzhütte.</Text>
    <Weg Richtung="W" nach="Waldweg">Die Tür liegt im Westen.</Weg>
    <Objekt ID="Taschenlampe" Flags="leuchtet">Eine Taschenlampe</Objekt>
  </Raum>

  <Raum ID="WegImMoor">
    <Text>Um den Weg ist ein Moor.</Text>
    <Weg Richtung="S" nach="Waldweg">Im Süden ist der Wald.</Weg>
    <Weg Richtung="W" nach="Höhle">Der Weg biegt ab
                                   zu einer Höhle im Westen.</Weg>
    <Objekt ID="Beeren" Flags="essbar" Punkte="20">Leckere
                                                   Blaubeeren</Objekt>
  </Raum>

  <Raum ID="Höhle" Flags="Dunkel">
    <Text>Du bist in einer Höhle.</Text>
    <Objekt ID="Schatz" Punkte="100">Ein wertvoller Goldschatz</Objekt>
    <Weg Richtung="O" nach="WegImMoor">Der Ausgang liegt im Osten.</Weg>
  </Raum>
</Spiel>
```


- Das `Spiel`-Element enthält eine Liste von `Raum`-Elementen (mindestens eins).
- `Raum`-Elemente müssen zuerst ein `Text`-Element enthalten (genau eins), und dann eine Folge von `Weg` und `Objekt`-Elementen in beliebiger Reihenfolge und beliebig häufig. Die Liste kann auch leer sein. Es ist möglich, dass erst einige Objekte kommen, dann einige Wege, dann wieder Objekte, u.s.w.
- Die Elemente `Text`, `Weg` und `Objekt` haben Text als Inhalt.
- Werte des Attributs `ID` sind Bezeichner und müssen eindeutig sein. Jeder `Raum` und jedes `Objekt` muss eine `ID` haben. Die Verweise auf Räume im Attribut `nach` sollen auf existierende Räume verweisen. (Wenn Anforderungen nicht erfüllt werden können, versuchen Sie möglichst nah zu kommen, siehe auch Aufgabe 2).
- Das Attribut `Richtung` von `Weg` kann nur einen der Werte `N`, `O`, `S` und `W` annehmen.
- `Weg` muss die Attribute `Richtung` und `nach` haben.
- Das Attribut "`Flags`" von `Raum` und `Objekt` kann eine beliebige Zeichenkette sein und ist optional.
- Das Attribut "`Punkte`" von `Objekt`" soll eine Zahl sein (wählen Sie den kleinsten Datentyp, der alle Ziffernfolgen erlaubt), und hat den Default-Wert `0`.

Lösung von Aufgabe 1 (DTD):

Aufgabe 2 (DTDs vs. XML Schema) 3 Punkte

Nennen und erläutern Sie kurz drei Beschränkungen von DTDs, die mit zur Entwicklung von XML Schema geführt haben (d.h. die es in XML Schema nicht gibt). Berücksichtigen Sie dabei insbesondere Einschränkungen, die sich bei Aufgabe 1 nicht vermeiden lassen. Es gibt nicht mehr Punkte, wenn Sie mehr als drei Beschränkungen nennen. Bei zu ähnlichen Beschränkungen kann allerdings ein Punkt abgezogen werden.

a) ______________________________

b) ______________________________

c) ______________________________

Aufgabe 3 (XML Schema) 12+2 Punkte

Es sind Teile einer XML Schema Definition für die obige Anwendung zu definieren. Der Rahmen und die Definition der Elemente `Weg` und `Objekt` seien schon gegeben:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="Weg"> ... </xs:element>
    <xs:element name="Objekt"> ... </xs:element>
```

a) Definieren Sie das Element `Text`. Es hat reinen Text als Inhalt.

b) Definieren Sie das Element `Raum`. Zur Vereinfachung sei vorausgesetzt, dass es zuerst ein `Text`-Element enthält, dann 0 oder mehr `Weg`-Elemente, und dann 0 oder mehr `Objekt`-Elemente. (Im Gegensatz zu Aufgabe 1 können Sie also hier eine bestimmte Reihenfolge zwischen `Weg` und `Objekt`-Elementen annehmen). Definieren Sie das Attribut `ID` so, dass es beliebige Zeichenketten erlaubt, nicht nur gültige IDs.

c) Definieren Sie das Element `Spiel` als Folge von Raum-Elementen mit einem Schlüssel, der die Eindeutigkeit der `ID`-Attributwerte der `Raum`-Elemente sicherstellt. Den Fremdschlüssel für "`nach`" brauchen Sie nicht zu definieren. Falls Sie es aber korrekt definieren, gibt es dafür zwei Bonuspunkte.

Aufgabe 4 (Software, Literatur) — 3 Punkte

a) Nennen Sie ein Programm, eine Bibliothek oder eine Webseite, mit der man XML Dokumente bezüglich eines XML Schemas validieren kann. Es wird nur die erste Antwort gewertet (keine Extrapunkte für mehrere Nennungen).

b) Nennen Sie ein Programm, eine Bibliothek oder eine Webseite, mit der man XQuery Anfragen ausprobieren kann. (Auch hier ist nur ein Punkt erreichbar.)

c) Nennen Sie eine Literaturstelle außer dem Skript (ein Lehrbuch oder eine Webseite), die Sie genutzt haben.

Aufgabe 5 (XPath) 9 Punkte

Sei das XML-Dokument von Aufgabe 1 betrachtet (Adventurespiel-Daten). Schreiben Sie folgende Anfragen in XPath 2.0. Das Dokument ist nicht validiert. Selbstverständlich sollen Ihre Anfragen nicht nur mit dem obigen Dokument funktionieren, sondern mit allen Dokumenten gleicher Struktur.

a) Welche Räume haben eine Verbindung zu dem Raum “`Waldweg`”? D.h. `Waldweg` ist der Wert des `nach`-Attributes in einem `Weg`-Kindknoten des gesuchten Raumes. Ihr Ausdruck soll aber nicht die `Raum`-Elemente liefern, sondern das Kind-Element `Text` dieser Räume.

b) Welche Objekte haben mehr Punkte als das Objekt mit der `ID` “`Beeren`”? Liefern Sie den Inhalt (Textknoten oder Datenwert) des `Objekt`-Knotens, nicht den ganzen Element-Knoten. Im Beispiel ist das Ergebnis:

```
Ein wertvoller Goldschatz
```

Denken Sie daran, dass das Dokument nicht validiert ist.

c) Liefern Sie den ersten und den letzten `Objekt`-Knoten des ganzen Dokumentes (in der Dokumentordnung).

Aufgabe 6 (XQuery) 8 Punkte

Gegeben sei wieder das XML-Dokument aus Aufgabe 1. Schreiben Sie folgende Anfragen in XQuery.

a) Berechnen Sie für jeden Raum die Anzahl der Ausgänge (das sind `Weg`-Kindelemente). Produzieren Sie eine HTML Tabelle in der unten gezeigten Form (jeweils mit der ID des Raums und der berechneten Anzahl). Sortieren Sie das Ergebnis nach der Anzahl, größte zuerst, und bei gleicher Anzahl nach der ID. Das Ergebnis soll also so aussehen (Einrückung und Zeilenaufteilung sind egal):

```
<table>
    <tr><td>Lichtung</td><td>2</td></tr>
    <tr><td>Waldweg</td><td>2</td></tr>
    <tr><td>WegImMoor</td><td>2</td></tr>
    <tr><td>Hoehle</td><td>1</td></tr>
    <tr><td>Huette</td><td>1</td></tr>
</table>
```

Tipp: Die Funktion zur Bestimmung der Länge einer Sequenz heisst `count(...)`.

b) Angenommen, Sie möchten die Wege anders darstellen, und zwar mit Elementen `N`, `O`, `S`, `W` entsprechend der Himmelsrichtung:

```
<Spiel>
  <Raum ID="Lichtung">
    <Text>Du bist auf einer Lichtung im Wald.</Text>
    <N nach="Waldweg">Ein Weg führt nach Norden.</N>
    <S nach="Lichtung">Im Süden ist ein Trampelpfad, ... </S>
  </Raum>

  <Raum ID="Waldweg">
    <Text>Du bist auf einem Waldweg.</Text>
    <N nach="WegImMoor">Der Weg führt weiter nach Norden.</N>
    <O nach="Hütte">Im Osten ist eine Hütte. Die Tür ist offen.</O>
  </Raum>

  ...
</Spiel>
```

Es ist also der Wert des Attributes `Richtung` der neue Element-Name geworden. Die `Objekt`-Elemente sind ausgeblendet. Schreiben Sie eine XQuery-Anfrage, die diese Umwandlung bewirkt.

Aufgabe 7 (XSLT) 6 Punkte

Schreiben Sie ein XSLT-Stylesheet, das die Raum-Daten in folgende Form überführt. Es handelt sich also um eine "Unordered List" aller Räume, wobei jeweils die ID das Raums fett gedruckt ist (mit `strong`), dann ein Zeilenumbruch (`br`), und anschließend der Text des Raums. Der Rahmen des Stylesheets ist auf der nächsten Seite gegeben. Dort ist auch Platz für die Lösung.

```
<html>
    <head><title>Raeume</title></head>
    <body>
        <ul>
            <li><strong>Lichtung</strong><br />
                Du bist auf einer Lichtung im Wald.</li>
            <li><strong>Waldweg</strong><br />
                Du bist auf einem Waldweg.</li>
            <li><strong>Hütte</strong><br />
                Du bist in einer kleinen Holzhütte.</li>
            <li><strong>WegImMoor</strong><br />
                Um den Weg ist ein Moor.</li>
            <li><strong>Höhle</strong><br />
                Du bist in einer Höhle.</li>
        </ul>
    </body>
</html>
```

Lösung zu Aufgabe 7 (XSLT):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
    version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:html="http://www.w3.org/1999/xhtml"
    xmlns="http://www.w3.org/1999/xhtml"
    exclude-result-prefixes="html">
<xsl:output
    method="xml"
    doctype-system=
      "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd"
    doctype-public="-//W3C//DTD XHTML 1.1//EN" />
```