

| | | | |
|----------|---|--------------|--------------|
| Gruppe A | PRÜFUNG AUS "SEMISTRUKTURIERTE DATEN" 184.705 | | 08. 01. 2013 |
| Kennnr. | Matrikelnr. | Familienname | Vorname |

Arbeitszeit: 100 Minuten. Aufgaben sind auf den Angabeblättern zu lösen; Zusatzblätter werden nicht gewertet. Bitte tragen Sie Namen, Studienkennzahl und Matrikelnummer ein. Halten Sie Ihren Studentenausweis bereit.

Aufgabe 1:

(9)

Betrachten Sie die folgende XML-Schema Datei **test.xsd**:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="x" type="xtype"/>

  <xsd:complexType name="xtype">
    <xsd:choice>
      <xsd:sequence>
        <xsd:element name="x" type="xtype"/>
        <xsd:element name="y" type="xsd:int"/>
      </xsd:sequence>
      <xsd:sequence>
        <xsd:element name="y" type="xsd:NMTOKEN"/>
        <xsd:element name="x" minOccurs="0" maxOccurs="2" type="xtype"/>
      </xsd:sequence>
    </xsd:choice>
  </xsd:complexType>

</xsd:schema>
```

Betrachten Sie weiters die sechs verschiedenen XML-Dateien, die unten angeführt sind.

Sie können davon ausgehen, dass alle folgenden XML-Dateien wohlgeformt sind. Es geht also lediglich darum, ihre Gültigkeit bezüglich **test.xsd** zu entscheiden.

Kreuzen Sie an, welche der folgenden XML-Dateien gültig bezüglich **test.xsd** sind.

- | | | |
|--|------------------------------|--------------------------------|
| 1. <x><y>A</y></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |
| 2. <x><y>A B</y></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |
| 3. <x><y>A</y><y>1</y></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |
| 4. <x><y>A</y><x><y>A</y></x></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |
| 5. <x><x><x><y>A</y></x><y>0</y></x></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |
| 6. <x><x><x><y>A</y></x><y>1</y></x><y>2</y></x> | gültig <input type="radio"/> | ungültig <input type="radio"/> |

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Aufgabe 2:

(12)

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

1. Wird ein leeres XSLT-Stylsheet auf ein XML-Dokument angewandt, wird standard-mäßig ein Fehler geliefert. wahr falsch
2. Das Resultat eines XPath Ausdrucks ist immer in Dokument-Order. wahr falsch
3. XQuery und XPath 2.0 bauen auf demselben Datenmodell auf. wahr falsch
4. SAX ist keine W3C-Recommendation, aber ein de-facto Standard. wahr falsch
5. Ein XSLT Stylesheet muss ein wohlgeformtes XML-Dokument sein. wahr falsch
6. SAX Filter dienen zur Umleitung der Ausgabe in eine Datei. wahr falsch
7. HTML ist eine Teilmenge der Sprache XML. wahr falsch
8. XML ist eine Teilmenge der Sprache HTML. wahr falsch

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Die folgenden Aufgaben 3 – 8 beziehen sich auf das XML-Dokument **kurznachrichten.xml**, das Sie auf der letzten Seite dieser Prüfungsangabe finden.

Aufgabe 3:

(12)

Vervollständigen Sie das DTD Dokument **kurznachrichten.dtd**, sodass XML-Dokumente in der Gestalt von **kurznachrichten.xml** (siehe Anhang) bezüglich dieser DTD gültig sind. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Das Element **kurznachrichten** ist das Wurzelement und besteht aus genau einem **users**-Element gefolgt von beliebig vielen **follows**-Elementen. Es soll auch kein **follows**-Element erlaubt sein.
- Das Element **users** besteht aus mindestens einem **user**-Element welches wiederum **nachricht**-Elemente beinhaltet. Ein **nachricht**-Element hat gemischten Inhalt und darf leere Elemente der Form **link** enthalten.
- Das Element **follows** besteht aus genau zwei **ref**-Elementen, und beschreibt welche User die Nachrichten anderer User verfolgen. Die Attribute der **ref**-Elemente verweisen auf das **kuerzel** Primärschlüssel-Attribut des **user**-Elements.
- Spezifizieren Sie die Attribute (wählen Sie sinnvolle Typen aus!) sowie die nicht näher erläuterten Häufigkeiten entsprechend dem **kurznachrichten.xml** Dokument im Anhang. Alle Attribute sollen verpflichtend sein. Versuchen Sie weiters entsprechende Fremdschlüssel zu finden und in der DTD abzubilden.

Datei **kurznachrichten.dtd**:

Aufgabe 4:

(6)

Stellen Sie sich vor, Sie definieren eine XML-Schema Datei für Dokumente in der Gestalt von **kurznachrichten.xml**. Welche Aussagen würden auf ein solches XML-Schema, bzw. allgemein zum Thema XML-Schema, zutreffen:

1. Das **datum** Attribut des **nachricht**-Elements lässt sich in XML-Schema genauer spezifizieren als in DTDs. wahr falsch
2. Nehmen Sie an Sie wollen die Anzahl der **link**-Elemente im gemischten Element **nachricht** auf maximal zwei beschränken. XML-Schema bietet hierfür die nötige Funktionalität. wahr falsch
3. In einem XML Schema für **kurznachrichten.xml** muss ein TargetNamespace angegeben werden wahr falsch
4. In XML-Schema dürfen gemischte Elemente auch gemischte Elemente als Kind-Elemente besitzen. wahr falsch

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Aufgabe 5:

(10)

Betrachten Sie die folgenden XPath-Abfragen angewandt auf das Dokument **kurznachrichten.xml** (siehe Anhang).

- Falls der angegebene XPath Ausdruck keine Knoten selektiert, notieren Sie im entsprechenden Feld "leere Ausgabe".
- Falls als Ergebnis **user** oder **ref** Elemente selektiert werden, geben Sie jeweils das Attribut **kuerzel** an.
- Falls als Ergebnis mehrere Elemente selektiert werden, trennen Sie die jeweiligen Ausgaben durch Leerzeichen.

Betrachten Sie dazu folgendes Beispiel:

```
//ref
```

```
a b b a b f c a d c e c f b
```

Geben Sie nun die entsprechende Ausgaben der folgenden XPath-Abfragen an.

```
//user[last()]
```

```
//user[*]
```

```
//ref[2]
```

```
(//ref)[2]
```

```
//user[@kuerzel=//follows/ref[2]/@kuerzel]
```

Aufgabe 6:

(8)

Betrachten Sie folgende-XQuery Abfrage **kurznachrichten.xq**:

```
for $u in //user
let $k := $u/@kuerzel
order by $k descending
return element {string($k)}{
  count(//follows[ref[2]/@kuerzel=$k])
}
```

Geben Sie nun die Ausgabe von **kurznachrichten.xq** angewandt auf **kurznachrichten.xml** an.

Die exakte Behandlung von Whitespaces ist für dieses Beispiel nicht relevant.

Erstellen Sie ein XSLT-Stylesheet **kurznachrichten.xsl**, das angewandt auf Dokumente der Gestalt **kurznachrichten.xml** zu jedem User dessen Followers anführt. Beispielsweise ist der User mit Kürzel **a** ein Follower des Users mit Kürzel **b**, wenn im Quelldokument folgendes angegeben ist:

```
<follows><ref kuerzel="a"/><ref kuerzel="b"/></follows>
```

Für das Dokument **kurznachrichten.xml** soll folgende Ausgabe erzeugt werden:

```
<followers>
  <user kuerzel="a">
    <follower kuerzel="b"/>
    <follower kuerzel="c"/>
  </user>
  <user kuerzel="b">
    ...
  </followers>
```

Vervollständigen Sie hier das XSLT-Stylesheet **kurznachrichten.xsl**. Die Verwendung von Kontrollstrukturen wie `xsl:for-each` ist für die Lösung grundsätzlich erlaubt, aber nicht erforderlich! Sie brauchen sich nicht um Whitespaces etc. zu kümmern.

Datei **kurznachrichten.xsl**:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">

</xsl:template>

  <xsl:template match="user">

</xsl:template>

  <xsl:template match="follows">

</xsl:template>

</xsl:stylesheet>
```

Aufgabe 8:

(9)

Vervollständigen Sie den folgenden SAX-Handler, der angewandt auf Dokumente der Gestalt **kurznachrichten.xml** bei jeder Nachricht überprüft, ob sie maximal 140 Zeichen lang ist.

Dabei zählt für jede Nachricht (Element `nachricht`)

- jedes Zeichen (Character), das Kindknoten der Nachricht ist, als 1 Zeichen
- jedes Element `link`, unabhängig von der Länge des `href` Attributs, als 20 Zeichen

Für jede Nachricht, in der mehr als 140 Zeichen vorkommen, soll in die Standardausgabe eine Meldung `Nachricht zu lang!` ausgegeben werden.

```
public class NachrichtHandler extends DefaultHandler {
    //Hier wird vermutlich zumindest eine Variable benötigt!

    public void endElement(String uri, String localName, String qName) throws SAXException {

    }

    public void characters(char[] ch, int start, int length) throws SAXException {

    }
}
```


Sie können diese Seite abtrennen!

Datei kurznachrichten.xml:

```
<kurznachrichten>
  <users>
    <user kuerzel="a" name="Alice">
      <nachricht datum="2012-03-28">Hallo!</nachricht>
      <nachricht datum="2012-03-28">@b Wie geht's?</nachricht>
      <nachricht datum="2012-12-24">Frohe Weihnachten!</nachricht>
    </user>
    <user kuerzel="b" name="Bob">
      <nachricht datum="2013-01-01">Frohes Neues Jahr!</nachricht>
      <nachricht datum="2013-01-04">
        @f Schau auf <link href="http://www.tuwien.ac.at/">!</nachricht>
    </user>
    <user kuerzel="c" name="Carol"/>
    <user kuerzel="d" name="Dave"/>
    <user kuerzel="e" name="Erin"/>
    <user kuerzel="f" name="Frank"/>
  </users>
  <follows><ref kuerzel="a"/><ref kuerzel="b"/></follows>
  <follows><ref kuerzel="b"/><ref kuerzel="a"/></follows>
  <follows><ref kuerzel="b"/><ref kuerzel="f"/></follows>
  <follows><ref kuerzel="c"/><ref kuerzel="a"/></follows>
  <follows><ref kuerzel="d"/><ref kuerzel="c"/></follows>
  <follows><ref kuerzel="e"/><ref kuerzel="c"/></follows>
  <follows><ref kuerzel="f"/><ref kuerzel="b"/></follows>
</kurznachrichten>
```

Gesamtpunkte: 75