

Gruppe A	PRÜFUNG AUS "SEMISTRUKTURIERTE DATEN" 184.705		12. 10. 2012
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="a" type="atype"/>

  <xsd:complexType name="atype">
    <xsd:choice minOccurs="1" maxOccurs="2">
      <xsd:sequence>
        <xsd:element name="b" minOccurs="0" type="xsd:int"/>
      </xsd:sequence>
      <xsd:choice>
        <xsd:element name="a" minOccurs="2" maxOccurs="2" type="atype"/>
      </xsd:choice>
    </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. <a/>                                  | gültig <input checked="" type="checkbox"/> | ungültig <input type="checkbox"/>            |
| 2. <a><a/></a>                           | gültig <input type="checkbox"/>            | ungültig <input checked="" type="checkbox"/> |
| 3. <a><a><b>0</b></a></a>                | gültig <input type="checkbox"/>            | ungültig <input checked="" type="checkbox"/> |
| 4. <a><a><a/></a></a>                    | gültig <input type="checkbox"/>            | ungültig <input checked="" type="checkbox"/> |
| 5. <a><a/><a/></a>                       | gültig <input checked="" type="checkbox"/> | ungültig <input type="checkbox"/>            |
| 6. <a><a><b>0</b></a><a><b>0</b></a></a> | gültig <input checked="" type="checkbox"/> | ungültig <input type="checkbox"/>            |

(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. XPath ist keine W3C-Recommendation, aber ein de-facto Standard. wahr  falsch
2. Die Ordnung der Knoten im Resultat eines XPath-Ausdrucks ist immer in Document Order. wahr  falsch
3. Der Speicherbedarf eines DOM-Parsers ist unabhängig von der Größe des geparsten XML-Files. wahr  falsch
4. Attributewerte werden von SAX Parsern ignoriert. wahr  falsch
5. Das Typsystem von XQuery basiert auf XML Schema. wahr  falsch
6. Die DTD Elementdeklaration (A?**|**B?**|**C?) ist gleichwertig mit (A**|**B**|**C)?. wahr  falsch
7. Rekursive Definitionen wie z.B. `<!ELEMENT B (A,B?,C)>` sind in DTDs erlaubt. wahr  falsch
8. SGML ist eine aktuelle Weiterentwicklung von XML. 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 – 7 beziehen sich auf das XML-Dokument **forum.xml**, das Sie auf der letzten Seite dieser Prüfungsangabe finden.

**Aufgabe 3:**

(12)

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

- Das Element **forum** ist das Wurzelement und besteht aus genau einem **benutzer**-Element und genau einem **nachrichten**-Element; die Reihenfolge dieser zwei Elemente soll beliebig sein.
- Das Element **benutzer** besteht aus mindestens einem **user**-Element; diese sollen leer sein aber verpflichtende Attribute besitzen.
- Das Element **nachrichten** besteht aus mindestens einem **nachricht**-Element; dessen Attribute sind ebenfalls verpflichtend und der Inhalt des Elements ist gemischt.
- Das **kuerzel**-Attribut des **user**-Elements und das **id**-Attribut des **nachricht**-Elements sind als Primärschlüssel zu deklarieren. Versuchen Sie die entsprechenden Fremdschlüssel zu finden und in der DTD abzubilden.
- Sollten bei bestimmten Elementen oder Attributen keine näheren Angaben bezüglich des genauen Typs vorgegeben sein, wählen Sie selbst einen sinnvollen Typ aus.

Datei **forum.dtd**:

```
<!ELEMENT forum ((benutzer, nachrichten)|(nachrichten,benutzer))>
<!ELEMENT benutzer (user+)>
<!ELEMENT user EMPTY>
<!ATTLIST user
  geschlecht (m | f) #REQUIRED
  kuerzel ID #REQUIRED
  name CDATA #REQUIRED>
<!ELEMENT nachrichten (nachricht+)>
<!ELEMENT nachricht (#PCDATA | a)*>
<!ATTLIST nachricht
  datum CDATA #REQUIRED
  id ID #REQUIRED
  titel CDATA #REQUIRED
  user CDATA #REQUIRED>
<!ELEMENT a (#PCDATA)>
<!ATTLIST a
  nachricht IDREF #IMPLIED
  user IDREF #IMPLIED>
```

#### Aufgabe 4:

(6)

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

1. Zum Spezifizieren des Inhalts des **forum**-Elements eignet sich das **xsd:all**-Konstrukt. wahr  falsch
2. XML-Schema bietet eine reichhaltige Palette von Datentypen an; damit können Dokumente exakter spezifiziert werden, als mittels DTDs. wahr  falsch
3. Der Typ für das **geschlecht**-Attribut wird am besten mittels des **xsd:extension**-Konstrukts spezifiziert. wahr  falsch
4. XML-Schema unterstützt, im Gegensatz zu DTDs, keine Schlüsselattribute. 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 **forum.xml** (siehe Anhang).

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

Betrachten Sie dazu folgendes Beispiel:

```
//nachricht
```

```
m1 m2 m3 m4 m5 m6
```

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

```
//nachricht[@user='u1'][2]
```

```
m3
```

```
//nachricht[2][@user='u1']
```

```
leere Ausgabe
```

```
//nachricht[*]
```

```
m2 m3 m5 m6
```

```
//nachricht[a/@nachricht=@id]
```

```
m5
```

```
//nachricht[//a/@nachricht=@id]
```

```
m1 m2 m5
```

## Aufgabe 6:

(8)

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

```
for $u in //user
return element {string($u/@kuerzel)} {
  let $n := //nachricht[@user=$u/@kuerzel]
  let $fn := //nachricht[@id=$n/a/@nachricht]
  let $fu := //user[@kuerzel=$fn/@user]/@name
  return $fu[1]
}
```

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

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

```
<u1 name="user2"/>
<u2 name="user1"/>
<u3 name="user2"/>
<u4 name="user4"/>
```

Erstellen Sie ein XSLT-Stylesheet **forum.xsl**, das angewandt auf Dokumente der Gestalt **forum.xml** jedem **a** Element mit **nachricht** Attribut, ein Attribut **user** hinzufügt (sofern nicht schon vorhanden). Inhalt dieses Attributs **user** soll das Kürzel des Users sein, der die durch das Attribut **nachricht** spezifizierte Nachricht erstellt hat. Der Rest des Dokuments soll unverändert bleiben. Beachten Sie dazu folgendes Beispiel:

Ist im Dokument folgendes Element vorhanden

```
<a nachricht="m5"/>
```

soll es ersetzt werden durch

```
<a nachricht="m5" user="u4"/>
```

Vervollständigen Sie hier das XSLT-Stylesheet **forum.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. Sie dürfen auch XSLT 2.0 verwenden, dies ist aber für die Lösung nicht erforderlich.

Datei **forum.xsl**:

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

  <xsl:template match="@*|node()">
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

  <xsl:template match="a[@nachricht][not(@user)]">
    <a nachricht="{@nachricht}"
      user="{//nachricht[@id=current()/@nachricht]/@user}">
      <xsl:apply-templates/>
    </a>
  </xsl:template>
</xsl:stylesheet>
```

Vervollständigen Sie die folgende Methode `process`, die ein DOM Document erhält, und angewandt auf Dokumente der Gestalt `forum.xml` jedem `a` Element mit `nachricht` Attribut, ein Attribut `user` hinzufügt (sofern nicht schon vorhanden). Inhalt dieses Attributs `user` soll das Kürzel des Users sein, der die durch das Attribut `nachricht` spezifizierte Nachricht erstellt hat. Der Rest des Dokuments soll unverändert bleiben. Diese Aufgabenstellung entspricht also jener des XSLT-Beispiels. Beachten Sie dazu folgendes Beispiel:

Ist im Dokument folgendes Element vorhanden

```
<a nachricht="m5"/>
```

soll es ersetzt werden durch

```
<a nachricht="m5" user="u4"/>
```

Um Fehlerbehandlung müssen Sie sich nicht kümmern.

```
public void process(Document doc) {
    Map<String,String> nachrichtUser = new HashMap<String,String>();

    NodeList nachrichten = doc.getElementsByTagName("nachricht");
    for (int i = 0; i < nachrichten.getLength(); i++) {
        Element nachricht = (Element) nachrichten.item(i);
        nachrichtUser.put(nachricht.getAttribute("id"), nachricht.getAttribute("user"));
    }

    NodeList references = doc.getElementsByTagName("a");
    for (int i = 0; i < references.getLength(); i++) {
        Element reference = (Element) references.item(i);
        if (reference.hasAttribute("nachricht") && !reference.hasAttribute("user")) {
            reference.setAttribute("user", nachrichtUser.get(reference.getAttribute("nachricht")));
        }
    }
}
```



Sie können diese Seite abtrennen!

Datei forum.xml:

```
<forum>
  <benutzer>
    <user kuerzel="u1" geschlecht="m" name="user1"/>
    <user kuerzel="u2" geschlecht="f" name="user2"/>
    <user kuerzel="u3" geschlecht="f" name="user3"/>
    <user kuerzel="u4" geschlecht="m" name="user4"/>
  </benutzer>
  <nachrichten>
    <nachricht id="m1" user="u1" datum="2012-10-12" titel="Titel1">
      Inhalt1.
    </nachricht>
    <nachricht id="m2" user="u2" datum="2012-10-12" titel="Titel2">
      Bezieht sich auf die <a nachricht="m1">Nachricht</a> von <a user="u1">User</a>.
    </nachricht>
    <nachricht id="m3" user="u1" datum="2012-10-13" titel="Titel3">
      Antwort auf <a nachricht="m2">Nachricht</a>.
    </nachricht>
    <nachricht id="m4" user="u3" datum="2012-10-13" titel="Titel4">
      Inhalt4.
    </nachricht>
    <nachricht id="m5" user="u4" datum="2012-10-13" titel="Titel5">
      Das ist eine eigenartige <a nachricht="m5">Nachricht</a>.
    </nachricht>
    <nachricht id="m6" user="u3" datum="2012-10-14" titel="Titel6">
      Antwort auf <a nachricht="m2">Nachricht</a>.
    </nachricht>
  </nachrichten>
</forum>
```

Gesamtpunkte: 75