

Klausur “Semistrukturierte Daten und XML”
Sommersemester 2007
Prof. Dr. Wolfgang May
19. Juli 2007, 13:30-15:30 Uhr
Bearbeitungszeit: 90 Minuten

Vorname:

Nachname:

Matrikelnummer:

Bei der Klausur sind **keine Hilfsmittel** (Skripten, Taschenrechner etc.) erlaubt. Handies müssen ausgeschaltet sein. Papier wird gestellt. Benutzen Sie nur die **ausgeteilten**, zusammengehefteten **Blätter** für Ihre Antworten. Schreiben Sie mit blauem/schwarzem Kugelschreiber, Füller etc.; Bleistift ist nicht erlaubt. Beantworten Sie die Fragen auf Deutsch oder Englisch.

Auf dem letzten Blatt finden Sie ein XML-Dokument, das in allen Aufgaben verwendet wird. Trennen Sie es ggf. zur Bearbeitung der Aufgaben ab.

Zum **Bestehen** der Klausur sind **45** Punkte hinreichend.

- meine Note soll mit Matrikelnummer so bald wie möglich auf der Vorlesungs-Webseite veröffentlicht werden.
- meine Note soll nicht veröffentlicht werden; ich erfahre sie dann vom Prüfungsamt (bzw. für nicht von einem Prüfungsamt verwaltete Teilnehmer: beim Abholen des Scheins).

	Max. Punkte	Schätzung für “4”
Aufgabe 1 (Allgemeines, DTD, Validierung)	34	19
Aufgabe 2 (XPath, XQuery)	26	12
Aufgabe 3 (XSLT)	30	15
Summe	90	46

Note:

Aufgabe 1 (Allgemeines, DTD, Validierung) [34 Punkte]

Diese Aufgabe verwendet die auf dem hintersten Blatt zu findende Ebay-Datenbasis.

1. Geben Sie eine DTD zu der Datenbasis an. Entscheiden Sie bei dem Entwurf, welche Attribute welcher Elemente Sie als ID auszeichnen würden (20 P). (Diese dürfen Sie im folgenden auch als IDs verwenden)
2. Geben Sie einen endlichen deterministischen (Teil-)Automaten (als Diagramm) zur DTD an, mit dem sich ein "auction"-Element zu dieser DTD validieren läßt. Sie dürfen dabei Attribute ignorieren (nehmen Sie an, dass diese bei dem Übergang, der den öffnenden Tag verarbeitet, überprüft werden) (8 P).
3. Geben Sie einen XPath-Ausdruck an, der den Namen des Verkäufers der Auktion mit der id "a004" angibt. (2 P)
4. Nehmen Sie an, das folgende auction-Element würde als letztes Subelement des Wurzelementes in das Dokument eingefügt:

```
<auction id="a004" seller="Music-CD"
        start="19.07.2007" end="29.07.2007">
  <object category="p045">
    <type>Mercedes 200D</type>
    <year>1982</year>
    <description>Wischerblaetter neu</description>
    <start-price>30.00</start-price>
  </object>
</auction>
```

Ist das dabei erhaltene Dokument bzgl. Ihrer DTD gültig (mit kurzer Begründung)? (2 P)

5. Geben Sie das Ergebnis der Auswertung des XPath-Ausdrucks aus Aufgabenteil (3) gegenüber dem erhaltenen Dokument an (mit kurzer Begründung). (2 P)

Lösung

1. die DTD:

```

<!ELEMENT ebay (person*, category+, auction*)>
  <!-- am Anfang sind keine Personen und keine
        Auktionen eingetragen -->
<!ELEMENT person (name, address)>
  <!ATTLIST person id ID #REQUIRED
                e-mail CDATA #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address EMPTY>
  <!ATTLIST address street CDATA #REQUIRED
                zip CDATA #REQUIRED
                city CDATA #REQUIRED>
<!ELEMENT category EMPTY>
  <!ATTLIST category name ID #REQUIRED
                has-subcategory IDREFS #IMPLIED
                subcategory-of IDREFS #IMPLIED>

<!ELEMENT auction (object, bid*)>
  <!-- bei einer neuen Auktion sind am Anfang keine
        bids eingetragen -->
  <!ATTLIST auction id ID #REQUIRED
                  seller IDREF #REQUIRED
                  start CDATA #REQUIRED
                  end CDATA #REQUIRED>
<!ELEMENT object (type, year, description?, start-price)>
  <!ATTLIST object category IDREF #REQUIRED>
  <!-- oder IDREFS falls man mehrere erlauben will -->
<!ELEMENT type (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT start-price (#PCDATA)>

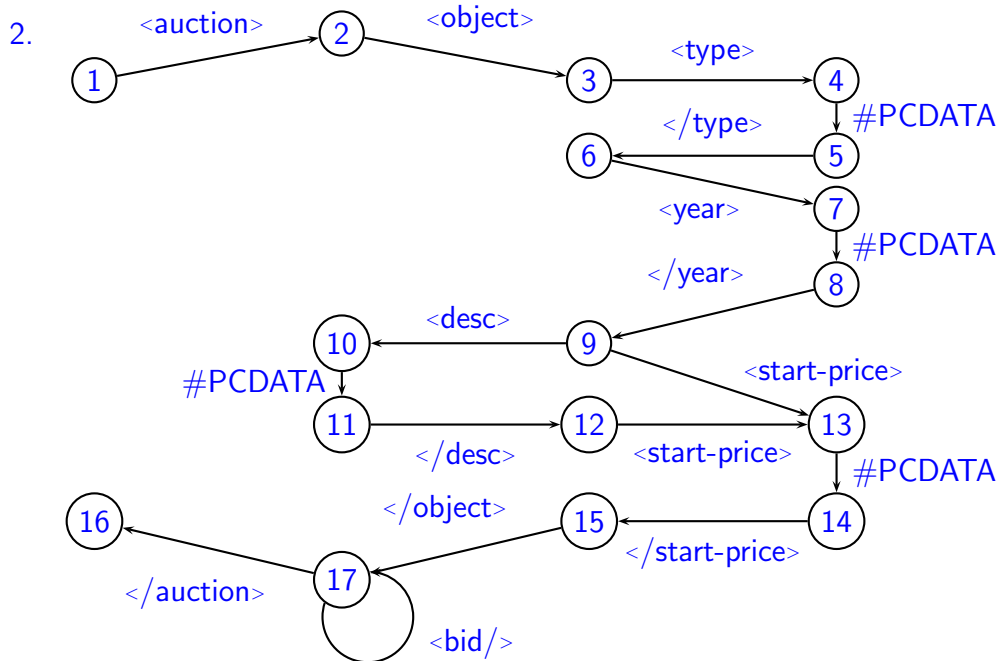
<!ELEMENT bid EMPTY>
  <!ATTLIST bid bidder IDREF #REQUIRED
                date CDATA #REQUIRED
                price CDATA #REQUIRED>

```

Bewertung:

- 1/2 pro relevantem Syntaxfehler,
- 1 für CDATA oder NMTOKEN(S) anstatt IDREF(S),
- 1 für weglassen von #IMPLIED,
- 1/2 für "*" statt "+" in ebay,
- 1/2 für fehlendes "?" bei description,
- 1/2 für IDREF statt IDREFS in category-Hierarchie.

Hinweis: e-mail nicht als ID geeignet, da mehrere Kunden dieselbe e-mail nutzen können.



Bewertung:

- 1 für nondeterministische Übergänge,
- 1 für Fehler bei bid*,
- 1/2 für fehlende #PCDATA-Kanten,
- 1/2 für Optionalität bei description vergessen (Folgefehler aus (a)).

3. mögliche Ausdrücke:

`//auction[@id="a004"]/id(@seller)/string(name)`

`id("a004")/id(@seller)/string(name)` (der zweite Ausdruck hat für eine reale Anwendung den Vorteil, dass man nicht einmal wissen muss, wie das ID-Attribut heisst, sondern einfach mit dem ID-Wert zugreifen kann).

4. Ja, es ist gültig. Die Struktur ist in Ordnung (wenn man mit "bid*" in auction berücksichtigt hat, dass eine Auktion am Anfang keine Gebote hat). Die IDREF-referenzierten Werte sind im Dokument vorhanden.

Hinweis: Die Werte von ID-Attributen müssen *dokumentweit* eindeutig sein. Die Werte von IDREF- (und IDREFS-) Attributen müssen als Werte von ID-Attributen im Dokument vorhanden sein. Es ist nicht möglich, dabei einzugrenzen, dass z.B. @seller auf eine Person verweist. Das Dokument ist daher korrekt. Eine Wertebereichsbedingung wäre in XML Schema mit key und keyref möglich.

5. Das Ergebnis ist leer. Das IDREF-Attribut @seller referenziert auf die Kategorie, der Name ist dort aber nicht als Subelement, sondern als Attribut angelegt.

Aufgabe 2 (XPath, XQuery [26 Punkte])

Diese Aufgabe verwendet die auf dem hintersten Blatt zu findende Ebay-Datenbasis. Geben Sie für die Aufgabenteile (1)-(6) je eine XPath- oder XQuery-Anfrage an, die das Ergebnis in dem jeweils angegebenen Format ausgibt (falls kein Format angegeben, beliebig).

1. Geben Sie eine XPath- oder XQuery-Anfrage an, die die Namen (als Elemente) aller Personen ergibt, die ein Gebot für einen Laptop abgegeben haben (2 P).

Lösung

```
//person[.//auction[object/@category="Laptop"]  
          /bid/id(@bidder)]/name  
oder  
//auction[object/@category="Laptop"]/bid/id(@bidder)/name
```

Die XPath-Ausdrücke eliminieren automatisch Duplikate, da das Ergebnis in Dokumentordnung ausgegeben wird. Wenn XQuery verwendet wird, muss man darauf achten, mit "for \$p in person ..." zu arbeiten.

Hinweis: wenn man object/@category als IDREFS definiert hat, muss man id(...) machen, oder contains(@category, "Laptop") verwenden.

2. Geben Sie eine XPath- oder XQuery-Anfrage an, die die Namen aller Personen ergibt, die ein Gebot für eine Auktion abgegeben haben, bei der sie selber der Verkäufer sind (dies ist nicht erlaubt!) (3 P)

Lösung

```
//auction[@seller=../bid/@bidder]/id(@seller)/string(name)  
oder  
//bid[@bidder=../parent:auction/@seller]/id(@bidder)/string(name)  
oder  
for $p in //person  
where //auction[id(@seller)=$p and bid/id(@bidder)=$p]  
return $p/string(name)
```

(auf Duplikatvermeidung achten!)

3. Geben Sie eine XPath- oder XQuery-Anfrage an, die alle Namen aller Personen angibt, die aktuell nicht als Verkäufer agieren, aber ein (möglicherweise bereits überbotenes) Gebot abgegeben haben. (3 P)

Lösung Wichtig: für die Negation mengenwertige Semantik von Pfaden beachten!

```
//person[. = //bid/id(@bidder) and
not (. = //auction/id(@seller))]/name/text()
```

oder

```
for $p in //person
where not (//auction[id(@seller) = $p])
and //bid[id(@bidder) = $p]
return $p/name/text()
```

4. Geben Sie eine XPath- oder XQuery-Anfrage an, die alle Tupel (Verkäufer (V), Käufer (K), Objektbezeichnung (O), Preis (P)) ergibt, so dass K das derzeit höchste Gebot (zum Preis P) für eine Auktion des Verkäufers V mit der Objektbezeichnung O abgegeben hat. (4 P)

Ausgabeformat:

```
<answer v='verkäufer-name' k='käufer-name' o='objektbezeichnung' p='preis' />
```

Lösung

```
for $a in //auction[bid] (: nur wenn es ein bid gibt! :)
let $b := $a/bid[position()=last()]
return
<answer v='{ $a/id(@seller)/name }'
k='{ $b/id(@bidder)/name }'
o='{ $a/object/type }'
p='{ $b/@price }' />
```

5. Geben Sie eine XPath- oder XQuery-Anfrage an, die für jede Person, die als Verkäufer agiert, angibt, wieviel Geld sie bekommen würde, wenn ihre Auktionen jetzt beendet würden. (6 P)

Ausgabeformat:

```
<result>
<person name='name1' money='betrag1' />
<person name='name2' money='betrag2' />
:
</ result>
```

Lösung

```

<result>
{ for $p in //person[.=//auction/id(@seller)]
  let $myauctions := //auction[id(@seller) = $p]
  return
    <person name='{ $p/string(name) }'
      money='{fn:sum($myauctions/bid[position()=last()]/
        /number(@price))}' />
}
</result>

```

oder

```

<result>
{ for $p in //auction/id(@seller)
  let $p := //auction[id(@seller) = $p]/bid[position()=last()]/@price
  return
    <person name='{ $p/string(name) }'
      money='{fn:sum($p/number())}' />
}
</result>

```

wer das Risiko mit der Bindung des let an eine Sequenz scheut, kann es auch so machen:

```

<result>
{ for $p in //person[.=//auction/id(@seller)]
  let $s := fn:sum(//auction[id(@seller) = $p]
    /fn:max(bid/number(@price)))
  return
    <person name='{ $p/string(name) }' money='{ $s }' />
}
</result>

```

Für Details in der Benutzung von fn:sum und number() wurden keine Punkte abgezogen.

- Geben Sie eine XPath- oder XQuery-Anfrage an, die die Namen aller Kunden ergibt, die für alle Auktionen, in denen ein Auto versteigert wird, ein Gebot abgegeben haben. (8 P).

Lösung Dies ist eine relationale Division. Erste Version: Vorgehensweise wie bei "not exists where not in" von SQL:

```

let $non-bidding :=
  (for $a in //auction[object/@category="Car"], $p in //person
    where not ($p = $a/bid/id(@bidder))
    return $p)
for $p in //person where not ($p = $non-bidding)
return $p/name

```

Mit every geht es deutlich einfacher:

```
for $p in //person
where every $a in //auction[object/@category="Car"]
    satisfies $a/bid/id(@bidder) = $p
return $p/name
```

oder in XPath 2.0:

```
//person[every $a in //auction[object/@category="Car"]
    satisfies $a/bid/id(@bidder) = $p]/name
```


Aufgabe 3 (XSLT [30 Punkte])

Diese Aufgabe verwendet ebenfalls die auf dem hintersten Blatt zu findende Ebay-Datenbasis.

Schreiben Sie ein XSLT-Stylesheet, das alle Auktionen, nach Kategorien und Unterkategorien gruppiert und innerhalb der Kategorien nach dem aktuellen Preis aufsteigend geordnet in einer HTML-Tabelle wie unten gezeigt listet.

- Auktionen, die mehreren Kategorien zugeordnet sind, werden mehrfach aufgeführt.
- der Eintrag für jede Auktion soll aus der Angabe des Artikeltyps, des aktuellen Preises sowie des Zeitpunktes des Auktionsendes bestehen.
- die Begrenzungslinien der Tabellen und Tabellenzellen dürfen Sie weglassen.

Name der 1. Kategorie
Name der 1. Unterkategorie von Kategorie 1
Name der 1. Unterkategorie der 1. Unterkategorie von Kategorie 1
<i>billigste Auktion dieser Kategorie</i>
<i>zweitbilligste Auktion dieser Kategorie</i>
Name der 2. Unterkategorie der 1. Unterkategorie von Kategorie 1
<i>Unterstrukturen wie oben</i>
Name der 2. Unterkategorie von Kategorie 1
<i>Unterstrukturen wie oben</i>
:
Name der 2. Kategorie
<i>Unterkategorien von Kategorie 2</i>
:

Lösung

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fn="http://www.w3.org/2005/xpath-functions"
                version="2.0">
<xsl:output method="html" indent="yes"/>
<xsl:template match="ebay">
  <html><body><table>
    <xsl:apply-templates
      select="category[not (@subcategory-of)]"/>
  </table></body></html>
</xsl:template>

<xsl:template match="category">
  <tr>
    <td>
      <table>
        <tr>
          <td colspan="3" align="center">
            <xsl:value-of select="@name"/>
          </td>
        </tr>
        <!-- all subcategories -->
        <xsl:apply-templates select="id(@has-subcategory)"/>
        <!-- all auctions of that category -->
        <xsl:variable name="cat" select="string(@name)"/>
        <xsl:apply-templates
          select="//auction[object/id(@category)/string(@name)=$cat]">
          <xsl:sort select="fn:max(bid/number(@price))"
            data-type="number" order="ascending"/>
        </xsl:apply-templates>
      </table>
    </td>
  </tr>
</xsl:template>

<xsl:template match="auction">
  <tr>
    <td>
      <xsl:value-of select="object/type"/>
    </td>
    <!-- alternatively, one of the following can be used -->
    <td>
      <xsl:value-of select="bid[position()=last()]/@price"/>
    </td>
    <td>
      <xsl:value-of select="bid[@price = fn:max(..bid/@price)]/@price"/>
    </td>
    <td>
      <xsl:value-of select="@end"/>
    </td>
  </tr>
</xsl:template>
</xsl:stylesheet>

```

Deutsch. Für Aufgaben 1-3 (DTDs, XQuery und XSLT) sei das folgende XML-Dokument, das die Datenbank einer Internet-Auktionsplattform (wie z.B. ebay) beschreibt, gegeben (siehe Rückseite): Es sind viele Personen registriert, deren Name, e-mail-Adresse und Adresse bekannt ist. Personen können Objekte verkaufen, indem sie eine Auktion starten, die 10 Tage läuft. Jedes Objekt ist einer oder mehreren vorgegebenen Kategorien und Unterkategorien (z.B. Elektronik, Computer, PC) zugeordnet. Eine Auktion beginnt mit einem Anfangspreis. Während der Laufzeit einer Auktion können andere Personen Gebote (bids) abgeben, indem sie sagen, dass sie einen Preis angeben, den sie für ein Objekt bezahlen würden, der höher als der Startpreis und die bisherigen Gebote sein muss. Die Gebote werden in chronologischer, und damit logischerweise in preislich aufsteigender Ordnung, abgelegt. Wenn eine Auktion endet, erwirbt derjenige Bieter, der das höchste (=letzte) Gebot abgegeben hat, das Objekt zum dem von ihm angegebenen Preis.

Englisch.

For Exercises 1-3 (DTDs, XQuery and XSLT), the following XML document that describes the database of an Internet Auction Platform (like e.g., ebay), is given (see on the back of that sheet): A number of persons is registered whose name, e-mail-address, and address are known. Persons can sell objects by starting an auction, which runs then for 10 days. Each object is assigned to one or more categories and subcategories (e.g. electronics, computer, PC). An auction starts with an initial price. During the running time of the auction, other persons can place bids by indicating a price that they are willing to pay for the object, that must be higher than the initial price and hogher than all previous bids for that object. The bids are stored in chronological, and thus also ascending by price, order. When an auction ends, the bidder who places the hightest (=last) bid purchases the object for the price he has bidden.

```
<?xml version="1.0"?>
<!DOCTYPE ebay SYSTEM "ebay.dtd">
<ebay>
  <person id="p001" e-mail="kalle@vermessung.ugoe.de">
    <name>Carl Friedrich Gauss</name>
    <address street="Bunsenstrasse 3" zip="37075" city="Goettingen"/>
  </person>
  <person id="p043" e-mail="gottel@bergwerk.tuc.de">
    <name>Gottlob Wilhelm Leibniz</name>
    <address street="Hauptstrasse 1" zip="38678" city="Clausthal"/>
  </person>
  <person id="p045" e-mail="hannes@rheinbegradigung.uka.de">
    <name>Johann Gottfried Tulla</name>
    <address street="Tullastrasse 42" zip="76131" city="Karlsruhe"/>
  </person>

  <category name="Electronics" has-subcategory="Computer Digicam"/>
  <category name="Computer" subcategory-of="Electronics"
    has-subcategory="PC Laptop"/>
  <category name="Laptop" subcategory-of="Computer"/>
  <category name="PC" subcategory-of="Computer"/>
  <category name="Entertainment" has-subcategory="Book CD"/>
  <category name="Camera" has-subcategory="Digicam"/>
  <category name="Digicam" subcategory-of="Camera Electronics"/>
  <category name="Book" subcategory-of="Entertainment"/>
  <category name="CD" subcategory-of="Entertainment"
    has-subcategory="Music-CD Game-CD"/>
  <category name="Music-CD" subcategory-of="CD"/>
  <category name="Game-CD" subcategory-of="CD Computer"/>
  <category name="Car"/>

  <auction id="a001" seller="p043" start="10.07.2007" end="20.07.2007">
    <object category="Laptop">
      <type>Dell Latitude X200</type>
      <year>2003</year>
      <description>the "e" button is broken</description>
      <start-price>200.00</start-price>
    </object>
    <bid bidder="p001" date="18.07.2007 12:34" price="200.00"/>
    <bid bidder="p045" date="19.07.2007 08:00" price="210.00"/>
    <bid bidder="p001" date="19.07.2007 08:03" price="215.00"/>
  </auction>
  <auction id="a002" seller="p001" start="12.07.2007" end="22.07.2007">
    <object category="Music-CD">
      <type>ABBA Gold</type>
      <year>2000</year>
      <start-price>1.00</start-price>
    </object>
    <bid bidder="p045" date="15.07.2007 22:30" price="1.16"/>
    <bid bidder="p001" date="15.07.2007 22:30" price="100.00"/>
    <bid bidder="p045" date="15.07.2007 22:30" price="110.00"/>
  </auction>
</ebay>
```