

FH Aargau: Vorlesung XML Sommersemester 2004

Test XML Schemasprachen (DTD und XML Schema)

17.5.2004, 13¹⁵ - 14¹⁵

Name: _____

Bitte schreiben Sie *sorgfältig* und *gut lesbar*! Danke!

Für diesen Test dürfen Sie Ihre *schriftlichen Unterlagen* verwenden, jedoch keine elektronischen Hilfsmittel wie Laptop oder Organizer!

Die folgenden Aufgaben lösen Sie am besten, indem Sie für Schemas (*DTD* und *XML Schema*) die jeweilige Syntax verwenden. Syntaxfehler werden nicht bewertet, solange sie nicht die Interpretation der Schemas beeinträchtigen. Für XML Schema können Sie als Alternative eine graphische Notation verwenden, die sich z.B. an der graphischen Sicht des XML Spy orientieren könnte. Bedenken Sie aber dabei, dass die graphische Sicht (je nach Aufgabe) *Elemente*, *Attribute*, und *Typen* sowie deren *Beziehungen* wiedergeben muss!

- 1) Sie sollen Software schreiben, die aus einem XML Schema eine möglichst gute DTD macht, also möglichst wenig Modellinformation verliert. Dazu sollen Sie die XML Schema Informationen in DTD Syntax mit *Parameter Entities* umwandeln, und die Informationen, die sich nicht abbilden lassen, in *Kommentaren* in der DTD unterbringen. Beschreiben Sie in einigen Worten und Beispielen Ihre generelle Implementierungsstrategie, und nehmen Sie dabei zur Abbildung der folgenden Konstrukte explizit Stellung:

1. Simple Types (für Elemente und Attribute, anonym und benannt)
2. Simple Type Restrictions
3. Complex Types (anonym und benannt)
4. Complex Type Restrictions
5. Complex Type Extensions
6. Elementdeklarationen (lokal und global)
7. Attributdeklarationen (lokal und global)
8. Identity Constraints

Beschreiben Sie die Abbildungen so *kurz* und *prägnant* wie möglich, so dass die Idee Ihrer Abbildungsvorschrift deutlich hervortritt.

- 2) Zeichen in *Mixed Content* sind (neben Kommentaren und Processing Instructions) der einzige typfreie Inhalt in XML Schema. Beschreiben Sie kurz
 - warum das so ist (was das Problem wäre, Mixed Content im Typmodell von XML Schema einzuordnen), sowie
 - die Konsequenzen dieser Tatsache.

Hier geht es um konzeptionelle Fragen, nicht um technische Details!

- 3) Bei der *Erweiterung (Derivation by Extension)* von *Complex Types* kann man *Element Only Types* nur zu *Element Only Types* erweitern und *Mixed Types* nur zu *Mixed Types*. Warum? (der Grund liegt in XML selber...)
- 4) Gegeben sein ein XML Schema, das Elemente und Attribute enthält, die der folgenden DTD entsprechen:

```

<! ELEMENT dokument (kunden, bestellungen, lieferanten) >
<! ELEMENT kunden (kunde+) >
<! ELEMENT bestellungen (bestellung+) >
<! ELEMENT lieferanten (lieferant+) >
<! ELEMENT kunde (#PCDATA) >
<! ATTLIST kunde
      id ID #REQUIRED >
<! ELEMENT bestellung (#PCDATA) >
<! ATTLIST bestellung
      bestellung ID #REQUIRED
      kunde IDREF #REQUIRED
      lieferant IDREF #REQUIRED >
<! ELEMENT lieferant (#PCDATA) >
<! ATTLIST lieferant
      id ID #REQUIRED >

```

Dabei enthalten verschiedene Elemente Informationen in PCDATA, die hier nicht interessieren, es geht in dieser Aufgabe nur um Referenzen. Geben Sie für die folgenden Anforderungen jeweils Identity Constraints an:

1. IDs von Kunden und Lieferanten müssen eindeutig sein (es darf also kein Kunde die ID eines Lieferanten haben und umgekehrt).
2. IDs von Bestellungen müssen eindeutig sein.
3. Der Kunde in einer Bestellung muss auf einen Kunden verweisen.
4. Der Lieferant in einer Bestellung muss auf einen Lieferanten verweisen.
5. Es darf nicht zwei Bestellungen von einem Kunden geben.

Bedenken Sie, dass *drei* Aspekte wichtig sind: der *Kontext*, der *Selector*, und das oder die *Fields*. Beachten Sie die Benennung der Elemente und Attribute!