

**Webbasierte Informationssysteme**  
**Wintersemester 10**  
 Besprechung 19.01.2011

## 8. Übungsblatt: DOM SAX DTD

### Übung 1 (Darstellung einer XML-Datei mittels DOM und SAX Parser)

- a) Gegeben ist ein XML Dokument *amazon.xml* (steht auf der Homepage als Download zur Verfügung). Geben Sie Program mit DOM-Parser an, so dass die spezielle Buchinformationen in *amazon.xml* als HTML Datei formatiert wie in Abbildung 1 ausgibt.

BILD	ASIN	TITEL	AMAZON PREIS
	059600625X	Excel Hacks: 100 Industrial-Strength Tips and Tools	\$16.47
	0596007795	Mind Hacks (Hacks)	\$16.47

Abbildung 1: Darstellung der Amazon XML Informationen in einem HTML Dokument.

- b) Erweitern Sie die JavaBean-Klasse `Book.java` und die beiden Parser Klassen `SAXBookHandler.java`, `DOMBookParser.java` so, dass sie von dem entsprechenden (SAX/DOM)-Parser mit Buchinformationen aus der Eingabedatei *amazon.xml* gefüllt wird (die Java-Templates und XML-Dateien stehen auf der Homepage als Download zur Verfügung). Ergänzen Sie darauhin die Methode `toHTMLFile()`, die spezielle Buchinformationen als HTML Datei formatiert wie in Abbildung 1 ausgibt.

### Übung 2 (DTDs)

Gegeben ist folgende DTD  $D_1$  von den XQuery Use Cases.

```

<!ELEMENT bib (book*)>
<!ELEMENT book (title, (author+ | editor+ ), publisher, price)>
<!ATTLIST book year CDATA #REQUIRED >
<!ELEMENT author (last, first)>
<!ELEMENT editor (last, first, affiliation)>
<!ELEMENT title (#PCDATA)> <!ELEMENT last (#PCDATA)>
<!ELEMENT first (#PCDATA)> <!ELEMENT affiliation (#PCDATA)>
<!ELEMENT publisher (#PCDATA)> <!ELEMENT price (#PCDATA)>
    
```

- a) Geben Sie jeweils das kleinste XML Dokument an, das valide bezüglich dieser DTD ist und das mindestens Knoten mit den Bezeichnungen (1) `bib`, (2) `book`, und (3) sowohl `bib`, `author` als auch `editor` enthält. Wir messen hier die Grösse von XML Dokumenten in der Anzahl von Zeichen *ohne* Whitespaces.

Überprüfen Sie Ihr Ergebnis, z.B. mit dem Tool `xmllint`<sup>1</sup>. Um zu testen, ob eine XML Datei *x.xml* wohlgeformt ist, geben Sie ein `“xmllint -format x.xml”`. Mit `“xmllint -dtdvalid d.dtd x.xml”` validieren Sie die Datei gegen die DTD *d.dtd*.

<sup>1</sup><http://xmlsoft.org/>

- b) Wie muss die DTD geändert werden um sicherzustellen, dass `bib` immer als `root`-Tag verwendet werden muss?<sup>2</sup>
- c) In einer Webapplikation wird die Produktion für `book`-Knoten wie folgt umdefiniert.

```
<!ELEMENT book ( (title, author+, publisher, price)
                  | (title, editor+, price, publisher) )>
```

Doch `xmllint` kann nun nicht mehr alle XML Dokumente für diese DTD validieren. Was ist das Problem, und wie kann man es beheben?

- d) Die Produktion für `book`-Knoten wird nun umdefiniert. Wir erhalten die DTD  $D_2$  mit

```
<!ELEMENT book (title, author*, price, publisher, editor*)>.
```

Geben Sie eine XPath Anfrage an, mit der man Dokumente unterscheiden kann, die jeweils valide bezüglich der DTDs  $D_1$  und  $D_2$  sind und die mindestens einen `book`-Knoten enthalten. Das heisst, die XPath Anfrage soll ein leeres Ergebnis für Dokumente aus der einen Gruppe liefern, und ein nicht-leeres Ergebnis für die Dokumente aus der anderen Gruppe.

### Übung 3 (DTD und endliche Automaten)

- a) Geben Sie zu jeder Elementdefinition in der DTD einen DFA (deterministic finite automaton) an, der das jeweilige Inhaltsmodell akzeptiert. Hierbei sollen gerichtete Kanten mit den entsprechenden Elementnamen beschriftet werden, die das entsprechende Element vom öffnenden bis hin zum schließenden Tag konsumieren. Beispielsweise kann die Elementdefinition `<!ELEMENT a (b,c)>` durch den Automaten  $\bullet \xrightarrow{<b>} \bullet \xrightarrow{<c>} \bullet$  repräsentiert werden. Spezifizieren Sie auch die Start- und Endzustände.

```
<!DOCTYPE Mondial[
  <!ELEMENT Mondial (Land*)>
  <!ELEMENT Land (LName, (Provinz+|Stadt+), Lage, Mitglied?)>
  <!ELEMENT LName (#PCDATA)>
  <!ELEMENT Provinz (PName, Flaeche, Stadt*)>
  <!ELEMENT PName (#PCDATA)>
  <!ELEMENT Flaeche (#PCDATA)>
  <!ELEMENT Stadt (SName, Einwohner)>
  <!ELEMENT SName (#PCDATA)>
  <!ELEMENT Einwohner (#PCDATA)>
  <!ELEMENT Lage (Kontinent, Prozent)>
  <!ELEMENT Kontinent (#PCDATA)>
  <!ELEMENT Prozent (#PCDATA)>
  <!ELEMENT Mitglied EMPTY>
]>
```

- b) Geben Sie ein minimales XML Dokument an, das valide bezüglich der obigen DTD ist und alle Elemente enthält. Überprüfen Sie ihre Automaten mit Hilfe dieses Dokuments.

---

<sup>2</sup>Achtung: `xmllint` unterstützt diese Erweiterung nicht!