

Web-Informationssysteme, WS 2009/10

Seminar-Übung 2: Information-Retrieval, CSS, Extraktion von Web-Tabellen

Besprechung am Mi 04.11.2009

— SÜ-2.1 —

Was Ihr wissen müßt über “Information Retrieval (Vektorraum-Modell)”

Wiederholung

- Was ist die **Aufgabe** einer (klassischen) Suchmaschine?
- Wie wird die **Qualität** von Suchmaschinen gemessen?
- Warum ist dieses Qualitätsmaß für das **Web** zumindest problematisch.
- Was ist die Grundidee des **Vektorraum-Modells**?
- Ihr solltet **binäre, einfache, normalisierte und TF/IDF** Varianten des Vektorraummodells anwenden können.
- Ihr solltet zumindest das **Cosinus-Ähnlichkeitsmaß** kennen und abschätzen können.

— SÜ-2.2 —

Auswertung von CSS Selektoren

Programmierung

Implementieren Sie einen einfachen Auswerter für CSS Selektoren. Gegeben ein CSS Selektor und ein HTML (oder XML) Dokument soll der Auswerter alle Elemente des Dokuments ausgeben, die mit dem Selektor matchen.

Jedes Element soll identifiziert werden durch seine *hierarchische* Position, z.B. 1.2.14.7 identifiziert das 7. Kind des 14. Kinds des 2. Kinds des 1. Kinds der (virtuellen) Dokumentwurzel, also des Knotens vom Typ `org.w3c.dom.Document` in der gegebenen DOM Repräsentation. Dazu soll das jeweilige Label des Elements ausgegeben werden.

Beispielsweise auf der Webseite der Vorlesung sollte der CSS Selektor `table.blaetter td > a` beispielsweise die folgende Ausgabe liefern:

```

CSS Engine with input wis-index.html for selector table.blaetter td > a!
2  table.blaetter td > a

4  =====
6  There are 29 total answers for table.blaetter td > a on wis-index.html:
8  D.1.2.30.2.3.1--a: Lageplan, Erdgeschoss
   D.1.2.35.9.2.1--a: Übungsblatt 08 (.pdf)
10 D.1.2.35.12.2.1--a: Übungsblatt 10 (.pdf)
   D.1.2.42.6.2.1--a: Themenblatt 04 (.pdf)
12 D.1.2.42.3.2.1--a: Themenblatt 01 (.pdf)
   D.1.2.30.3.3.1--a: Lageplan, Erdgeschoss
   D.1.2.42.9.2.1--a: Themenblatt 07 (.pdf)
   D.1.2.35.10.2.1--a: Übungsblatt 09 (.pdf)
14 D.1.2.35.8.2.1--a: Übungsblatt 07 (.pdf)
   D.1.2.35.15.2.1--a: Wiederholungsblatt 02 (.pdf)
16 D.1.2.42.10.2.1--a: Themenblatt 08 (.pdf)
   ...

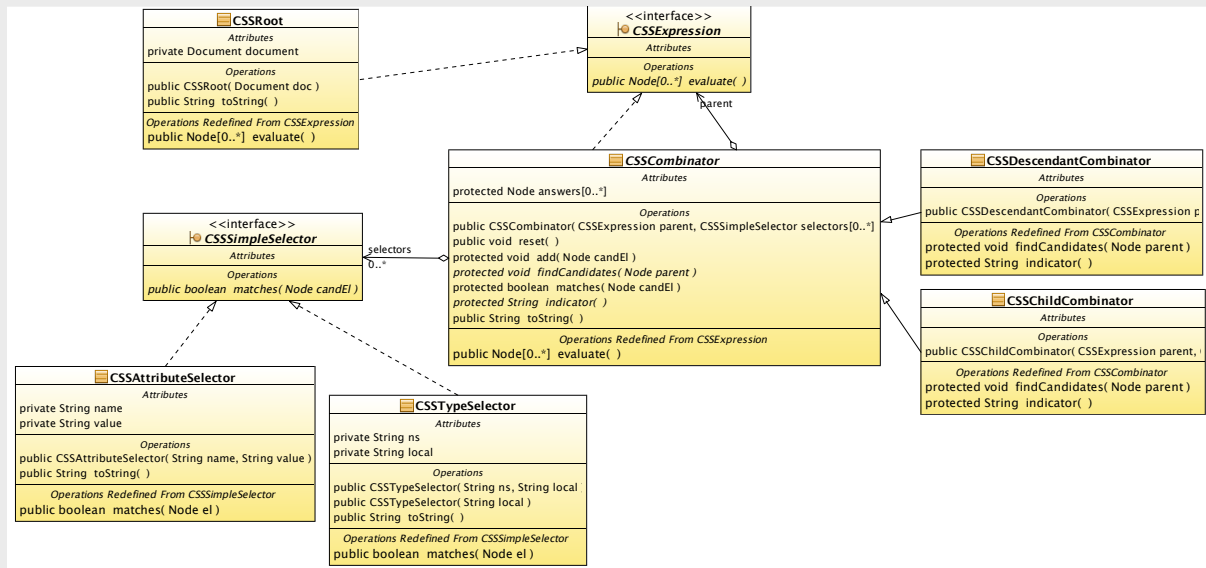
```

Der Auswerter muß nicht alle CSS3 Selektoren unterstützen. Er sollte aber zumindest die folgenden einfachen Selektoren sowie die Kombinatoren *descendant* (" ") und *child* (">") unterstützen.

- Universeller Selektor `*`.
- Typ-Selektoren wie `h1` (ohne Namensraum-Behandlung).
- ID-Selektor (wie `#nav`).
- Class-Selektor (wie `.menu`).

Wir verzichten also auf Pseudo-Klassen, Pseudo-Elemente, und Namensräume. Es ist auch nicht nötig Abkürzungen wie `#nav` für `*#nav` zu realisieren.

Hinweis: Wenn Ihr nicht alleine zu recht kommt (und nur dann): Als Hilfestellung findet Ihr in den Anlagen ([t02CLIDriver.java](#), [t02/css](#) Ordner) das Skelett eines Lösungsvorschlages. Das folgende UML-Diagramm zeigt den groben Aufbau des Lösungsvorschlages:



- Die Implementierung ist so aufgebaut, dass ein CSSCombinator ausgehend von den Antworten für einen übergeordneten Teil-Ausdruck (parent im Konstruktor) mögliche Antworten für seinen Kombinator-Typ (z.B. alle Kinder von solchen Antworten) berechnet und für jeden prüft, ob die zugeordneten *simple selectors* (CSSSimpleSelector) alle erfüllt sind.
- Der Parser ist etwas fragil. In der Praxis ist CSS bereits komplex genug, um den Einsatz eines Parser-generators wie ANTLR oder JavaCC zu rechtfertigen. Für den Lösungsvorschlag haben wir Euch davor bewahren wollen.

In den Anlagen-Dateien sind noch zu implementierende Stellen deutlich mit @TODO markiert.