

## 5. Übung zur Vorlesung "Internet-Suchmaschinen" im Sommersemester 2009

Prof. Dr. Gerd Stumme, Wi.Inf. Beate Krause

17. Juni 2009

### 1 Reguläre Ausdrücke (2)

Sie bekommen den Auftrag, ein Logfile eines Webservers zu untersuchen. Folgende Aufgaben sollen Sie mit Hilfe von regulären Ausdrücken lösen:

- Extrahieren Sie aus dem Logfile alle IP-Adressen.
- Finden Sie alle doppelten Wörter (z.B. das Das). Berücksichtigen Sie dabei, dass mehrere Leerzeichen zwischen den Wörtern stehen können und sowohl Groß- als auch Kleinschreibung zugelassen ist. Auch Wörter, die durch HTML Tags getrennt sind (Das Wetter ist **sehr** sehr schön), sollen gefunden werden.

### 2 String-Suche: Boyer-Moore

1. Finden Sie mit Hilfe des Boyer-Moore-Algorithmus den Substring "banana" in den folgenden Zeichenketten.

`http://www.buonissimo.org/ricette/131_ananasconcremadibanana.asp`  
`banabasanabanabanasabanananannasana`

`http://www.buonissimo.org/ricette/131_ananasconcremadibanana.asp`

- Bestimmen Sie dazu zuerst die Shifttabelle!
- Notieren Sie jeweils für die verschiedenen Strings, welche der beiden Strategien Sie wie oft weiterbringt. Nutzen Sie die Entscheidungsregel auf der Folie 33. Sie können wie folgt vorgehen:

```
http://www.buonissimo.org/ricette/131_ananasconcremadibanana.asp
banana.....
Mismatch bei j=6; D[6] = 1, 6 - last[/] = 6 0
.....banana.....
Mismatch bei j=6; D[6] = 1, 6 - last[b] = 5 0
.....banana.....
Mismatch bei j=6; D[6] = 1, 6 - last[s] = 6 0
```

usw.  $S$  steht für die Suffix-,  $O$  für die Occurrence-Heuristik.

### 3 Levenstein-Metrik

1. Zeigen Sie (durch Angeben einer Umwandlungsfolge der Form  $\text{test} \rightarrow \text{tost} \rightarrow \text{toast}$ ), dass  $\text{lev}(\text{"carsten"}, \text{"christina"}) \leq 6$ . (Schaffen Sie auch  $\leq 4$ ?)
2. Begründen Sie, warum  $\text{lev}(\text{"carsten"}, \text{"christina"}) \geq 2$ .

### 4 Praxisübung; Abgabe am 30.06.2009

Implementieren Sie einen Web-Spider. Er soll ausgehend von einer Menge von Start-URLs eine vorgegebene Anzahl weiterer Seiten aus dem Web sammeln.

#### Hinweise

- Sie brauchen keinen konkreten URL-Filter/Robots.txt-Mechanismus zu implementieren.
- Alle Datenstrukturen und die Dokumente können im Hauptspeicher gehalten werden.
- Das `HTMLDocument` aus der letzten Übung kann beim Extrahieren von Links nützlich sein; sie können auch ein Feld für die URL hinzufügen.
- `java.net.URL` weiß, wie man über HTTP Seiten herunterlädt.
- Der zweistellige Konstruktor von `URL` löst relative Links auf.
- Im Package `java.util.concurrent` gibt es viele Klassen, die beim Parallelisieren des Spiders helfen können.