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.

  Lösungsvorschlag:

  Einfache Möglichkeit ohne Berücksichtigung der Zahlenränge: `\b(?:\d{1,3}\.){3}\d{1,3}\b`

  Mit Berücksichtigung der Zahlenränge:

  `\b(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\b`

- 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.

  Lösungsvorschlag:

  Siehe Abbildung 1.

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`

   `banabasananabanabanasabananananannasana`

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

   - Bestimmen Sie dazu zuerst die Shifttabelle!

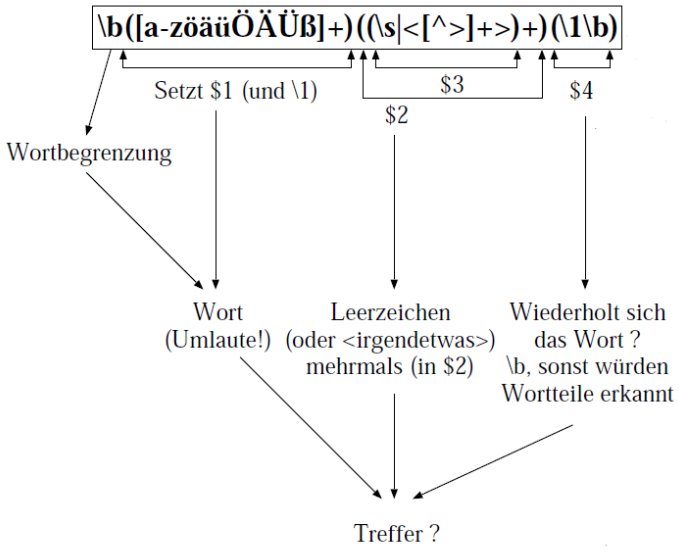

Figure 1: Regulärer Ausdruck zum Erkennen von doppelten Wörtern

Lösungsvorschlag:

Die Shifttabelle ist 662641.

- 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 O
......banana.....................................................
Mismatch bei j=6; D[6] = 1, 6 - last[b] = 5 O
...........banana.....................................................
Mismatch bei j=6; D[6] = 1, 6 - last[s] = 6 O
```

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

Lösungsvorschlag:

(Weiterführung des ersten Beispiels:)

```
................banana.....................................
Mismatch bei j=6; D[6] = 1, 6 - last[o] = 6 O
......................banana.....................................
Mismatch bei j=6; D[6] = 1, 6 - last[c] = 6 O
..........................banana.....................................
```

```
Mismatch bei j=6; D[6] = 1, 6 - last[1] = 6 O
.................................banana.....................
Mismatch bei j=3: D[3] = 2, 3 - last[_] = 3 O
.....................................banana..................
Mismatch bei j=6; D[6] = 1, 6 - last[s] = 6 O
.........................................banana.............
Mismatch bei j=6; D[6] = 1, 6 - last[e] = 6 O
................................................banana........
Mismatch bei j=5: D[5] = 4, 5 - last[b] = 4 SO
.............................................................banana....
Treffer!
```

Zweites Beispiel:

```
banabasananabanabanasabananananannasana
banana
Mismatch bei j=5; D[5] = 4, 5 - last[b] = 5 - 1 = 4 SO
....banana.........................
Mismatch bei j=3; D[3] = 2, 3 - last[s] = 3 - 0 = 3 O
.......banana.....................
Mismatch bei j=6; D[6] = 1, 6 - last[b] = 5 O
...........banana.................
Mismatch bei j=5; D[5] = 4, 5 - last[b] = 5 - 1 = 4 SO
.............banana.............
Mismatch bei j=5; D[5] = 4, 5 - last[s] = 5 - 0 = 5 O
.................banana.........
Mismatch bei j=6; D[6] = 1, 6 - last[n] = 6 - 5 = 1 SO
...................banana.........
Treffer!
```

Drittes Beispiel:

```
http://www.bibsonomy.org/tag/ananas+banana+mango
banana
Mismatch bei j = 6; D[6] = 1, 6 - last[/] = 6 O
......banana..............................
Mismatch bei j = 6; D[6] = 1, 6 - last[b] = 5 O
..........banana..........................
Mismatch bei j = 6; D[6] = 1, 6 - last[n] = 1 SO
...........banana..........................
Mismatch bei j = 6; D[6] = 1, 6 - last[o] = 6 O
...............banana......................
Mismatch bei j = 6; D[6] = 1, 6 - last[g] = 6 O
....................banana................
Mismatch bei j = 5; D[5] = 4, 5 - last[/] = 5 O
........................banana............
Mismatch bei j = 6; D[6] = 1, 6 - last[s] = 6 O
..............................banana.......
```

```
Mismatch bei j = 6; D[6] = 1, 6 - last[n] = 1 SO
.................................banana......
Treffer!
```

3 Levenstein-Metrik

1. Zeigen Sie (durch Angeben einer Umwandlungsfolge der Form test → tost → toast), dass lev("carsten", "christina") ≤ 6. (Schaffen Sie auch ≤ 4?)

   Lösungsvorschlag:

   Eine kürzeste Umwandlungsfolge wäre: carsten → chrsten → christen → christin → christina

2. Begünden Sie, warum lev("carsten", "christina") ≥ 2.

   Lösungsvorschlag:

   Weil "Christina" um zwei Zeichen länger ist als "Carsten". Es müssen also mindestens zwei Zeichenhinzufügungen durchgeführt werden.

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.