

## Vorbereitung: Numerische Muster zu Zeichenketten

**Definition.** Sei  $\Sigma$  ein Alphabet. Seien  $a_1, \dots, a_q$  Zeichen aus  $\Sigma$ . Das zur Zeichenkette  $(a_1, \dots, a_q)$  gehörige **Muster**  $(n_1, \dots, n_q) \in \mathbf{N}_1^q$  ist die Liste von Zahlen, die so gebildet wird:

- $n_1 := 1$ .
- Für  $k = 2, \dots, q$ :  
Gibt es ein  $i$  mit  $1 \leq i < k$  und  $a_k = a_i$ , so  $n_k = n_i$ ;  
sonst  $n_k := 1 + \max\{n_i \mid 1 \leq i < k\}$ .

**Beispiel:** Zu REGENBOGEN gehört das Muster 1232456324.

### Bemerkungen.

1.  $n_i = n_j \Leftrightarrow a_i = a_j$  für  $1 \leq i \leq j \leq q$ .
2.  $\{n_1, \dots, n_q\} = [1..m]$  mit  $m = \#\{n_1, \dots, n_q\}$   
[= Anzahl der verschiedenen Zeichen in  $(a_1, \dots, a_q)$ ].

## Algorithmus

**Ziel:** Zu einer Zeichenkette das numerische Muster ermitteln.

**Eingabe:** Zeichenkette als Liste `string = (a1, ..., aq)`.

**Ausgabe:** Numerisches Muster als Liste `pattern = (n1, ..., nq)`.

Vorbesetzung: leere Liste.

### Hilfsvariablen:

- `n` = aktuelle Nummer; Vorbesetzung:  $n = 0$ .
- `assoc` = Liste der abgearbeiteten verschiedenen Zeichen;  
dem Index  $i$  entspricht das Zeichen `assoc[i]`;  
Vorbesetzung: leere Liste.

### Prozedur:

Schleife über die Zeichen von `string`; aktuelles Zeichen ist `x`:

Falls es  $i$  gibt mit `x = assoc[i]`

hänge  $i$  an `pattern` an;

sonst:

inkrementiere n,  
hänge n an pattern an;  
hänge x an assoc an.

Ein Perl-Programm, das dieses umsetzt, steht [hier](#). Es dient nur zum Verständnis des folgenden, komplizierteren Programm und wird sonst nicht weiter benötigt.

---

## Aufbau eines regulären Suchausdrucks für ein Muster

Die Mustersuche mit Perl funktioniert am knappsten mit sogenannten regulären Ausdrücken. Hier das Minimum zum Verständnis des entsprechenden Programms:

Suchausdruck	Wirkung
<code>/./</code>	passt auf das erste Zeichen (falls die Zeichenkette nicht leer ist).
<code>/(.)/</code>	passt auf das erste Zeichen und weist es der Variablen \$1 zu.
<code>/. ./</code>	passt auf die ersten beiden Zeichen.
<code>/(. .)/</code>	passt auf die ersten beiden Zeichen und weist sie als Zweierzeichenkette der Variablen \$1 zu.
<code>/(.)(.)/</code>	passt auf die ersten beiden Zeichen und weist sie einzeln den Variablen \$1 und \$2 zu.
<code>/(.)\1/</code>	passt bei beliebigem ersten Zeichen und mit ihm identischen zweiten Zeichen. [Das erste Zeichen wird der Variablen \$1 zugewiesen; diese wird innerhalb des Suchausdrucks mit \1 bezeichnet.]
<code>/(.)(?!\1)/</code>	passt auf die ersten beiden Zeichen, falls das zweite <i>ungleich</i> dem ersten ist, und weist sie einzeln den Variablen \$1 und \$2 zu.
<code>/(.)(.)\1(.)\1/</code>	Muster 12131, falls verschiedene Zahlen nicht notwendig verschiedene Zeichen bedeuten sollen.
<code>/(.)(?!\1)\1(?!\1 \1)\1/</code>	Muster 12131, falls verschiedene Zahlen verschiedene Zeichen bedeuten sollen.

---

## Mustersuchprogramm

Das fertige Programm im typischen, hochkondensierten Perl-Stil, ist [hier](#) zu finden.

---

Autor: Klaus Pommerening, 29. Oktober 1999; letzte Änderung: 20. Februar 2000.

[E-Mail](mailto:Pommerening@imsd.uni-mainz.de) an Pommerening@imsd.uni-mainz.de.