TECHNISCHE UNIVERSITÄT DRESDEN

ALGORITHMEN UND DATENSTRUKTUREN

ÜBUNG 9: SUCHEN & ERSETZEN

Eric Kunze

eric.kunze@mailbox.tu-dresden.de

TU Dresden, 08.01.2021

KMP-Algorithmus

Aufgabe 1

KMP-ALGORITHMUS

- Mustersuche in (großen) Texten
- Ziel: Verschiebung des Musters um mehr als eine Position bei Nichtübereinstimmung.
- Methode: Ermittlung einer Verschiebetabelle `Tab[]` in **Phase 1**
- Bedeutung des Eintrags `Tab[i]=j`:
  Bei Nichtübereinstimmung an Stelle i wird Position j des Musters an aktueller Vergleichsstelle angelegt.
- Suchprozess in **Phase 2**

j-algo: `http://j-algo.binaervarianz.de/`

KMP-ALGORITHMUS

Suche das Muster `aaabaaaa` im Text `aaabaaabaaacaaabaaaa`.

Position	0	1	2	3	4	5	6	7
Pattern	a	a	a	b	a	a	a	a
Tabelle	-1	-1	-1	2	-1	-1	-1	3

Erster Versuch:

a a a b a a a **b** a a a c a a a b a a a a
a a a b a a a **a**

Tabelleneintrag an Position 7 ist 3, d.h. `Tab[7]=3` — Lege Position 3 des Musters an aktueller Vergleichsposition an:

a a a b a a a b a a a **c** a a a b a a a a
a a a b a a a **a**

Gleicher Prozess noch einmal: Missmatch an Position 7 des Musters — verschiebe Muster auf Position 3.

KMP-ALGORITHMUS (FORTSETZUNG)

Suche das Muster `aaabaaaa` im Text `aaabaaabaaacaaabaaaa`.

Position	0	1	2	3	4	5	6	7
Pattern	a	a	a	b	a	a	a	a
Tabelle	-1	-1	-1	2	-1	-1	-1	3

Wir legen das Muster also wieder an Position 3 an:

a a a b a a a b a a a **c** a a a b a a a a
a a a **b** a a a a

Wegen `Tab[3]`=2, lege Muster an Position 2 an:

a a a b a a a b a a a **c** a a a b a a a a
a a **a** b a a a a

Wegen `Tab[2]`=−1, lege Muster an Position −1 an:

a a a b a a a b a a a c a a a b a a a a
a a a b a a a a ☺

KMP-ALGORITHMUS — DIE ZYKLENMETHODE

Zwei Phasen:

- **1. Phase:** Markieren der längsten Teilwörter im Pattern, die mit einem Präfix übereinstimmen
  - ein Zyklus beginnt an einer Patternposition `i` falls `i` $\neq$ `0` und `Pat[0] = Pat[i]`
  - ein Zyklus endet an der kleisten Patternposition `i+m`, sodass `Pat[m+1]` $\neq$ `Pat[i+m+1]`
- **2. Phase:** Bestimmung der Tabelleneinträge
  - `Tab[0] = -1`
  - Tabelleneinträge nach einem Zyklus:
    Länge des längsten dort endenden Zyklus
  - Tabelleneinträgen in einem Zyklus:
    Tabelleneintrag der derzeitigen Position im längsten laufenden Zyklus
  - verbleibende Einträge: `0`

AUFGABE 1

(a) Geben Sie zu dem Pattern `aabaaacaab` die mit Hilfe des KMP-Algorithmus (Knuth-Morris-Pratt) berechnete Verschiebetabelle an.

(b) Mit Hilfe des KMP-Algorithmus ist die unten stehende Verschiebetabelle berechnet worden: Vervollständigen Sie das aus den Symbolen `a`, `b` und `c` bestehende Pattern.

Position	0	1	2	3	4	5
Pattern	c	b				a
Tabelle	-1	0	-1	1	0	2

KMP-ALGORITHMUS — DIE ZWEI-FINGER-METHODE

Die Methode beruht auf der Gleichung

$$\texttt{Tab[i]} = \max\{-1\} \cup \left\{ m \;\middle|\; \begin{array}{rcl} 0 \leq m & \leq & i-1 \\ b_0 \dots b_{m-i} & = & b_{i-m} \dots b_{i-1} \\ b_m & \neq & b_j \end{array} \right\} \qquad (\star)$$

Daraus ergibt sich nach Initialisierung von $\texttt{Tab[0]} = -1$ für jeden folgenden Eintrag `Tab[i]` folgendes Verfahren:

- *linker Finger*: wähle $m < i$ in absteigender Reihenfolge (also $i-1$, $i-2$, ...), sodass `Pat[i]` $\neq$ `Pat[m]`
- *Parallelverschiebung beider Finger bis zum linken Rand*: wenn `Pat[0...m-1]` = `Pat[i-m...i-1]`, dann fülle `Tab[i]` = m.
- wenn keine passende Position m gefunden werden kann, dann fülle `Tab[i]` = -1.

Levenshtein-Distanz

Aufgabe 2

LEVENSHTEIN-DISTANZ

Kosten zur Überführung eines Wortes $w = w_1 \dots w_n$ in ein Wort $v = v_1 \dots v_k$; schreibe $d(w_1 \dots w_j, v_1 \dots v_i) = d(j, i)$.

$$d(0, i) = i$$

$$d(j, 0) = j$$

$$d(j, i) = \min\{d(j, i-1) + 1, d(j-1, i) + 1, d(j-1, i-1) + \delta_{j,i}\}$$

für alle $1 \leq j \leq n$ und alle $1 \leq i \leq k$ wobei

$$\delta_{j,i} = \begin{cases} 1 & \text{wenn } w_j \neq v_i \\ 0 & \text{sonst} \end{cases}$$

Anschaulich: Überlagerung durch Pattern → Pfeile zeigen ”Ursprung“ des Minimums an

$w_j \neq v_i$:

+1	+1
+1	?

$w_j = v_i$:

+0	+1
+1	?

AUFGABE 2

Gegeben seien die Wörter $w = \texttt{espen}$ und $v = \texttt{beispiele}$.

(a) Berechnen Sie die Levenshtein-Distanz $d(w, v)$. Geben Sie dazu die Berechnungsmatrix an. Tragen Sie alle Zelleneinträge zusammen mit den dazugehörigen Pfeilen ein.

(b) Geben Sie die Levenshtein-Distanz $d(\texttt{espe}, \texttt{beispiel})$ an. Beachten Sie, dass `espe` und `beispiel` Präfixe von `espen` bzw. `beispiele` sind.

(c) Geben Sie zwei Alignments zwischen `espen` und `beispiele` an, die zu den minimalen Kosten führen. Dabei sollen die Alignments die jeweils angewendeten Editieroperation enthalten.

(d) Wieviele Alignments enthält die in Aufgabe (a) angegebene Berechnungsmatrix?

Weitere Aufgaben aus der Aufgabensammlung
mit Lösungen

AUFGABE 7.1.13 (AGS)

(a) Bestimmen Sie die mit Hilfe des KMP-Algorithmus berechnete Verschiebetabelle für das Pattern `abbabbaa`.

(b) Mit Hilfe des KMP-Algorithmus ist unten stehende Verschiebetabelle berechnet worden. Die mit einem „?“ markierten Einträge sind unbekannt. Vervollständigen Sie das aus den Symbolen `a`, `b` und `c` bestehende Pattern.

Position	0	1	2	3	4	5
Pattern	*b*					*c*
Tabelle	-1	?	?	0	?	3

AUFGABE 7.2.1 (AGS)

Gegeben seien die Wörter $w = \texttt{Dinstas}$ und $v = \texttt{Distanz}$.

(a) Berechnen Sie die Levenshtein-Distanz $d(w, v)$ zwischen w und v. Geben Sie die Berechnungsmatrix vollständig an.

(b) Geben Sie alle Alignments mit minimaler Levenshtein-Distanz zwischen w und v an.

AUFGABE 7.2.2 (AGS)

(a) Berechnen Sie die Levenshtein-Distanz $d(\texttt{bürste}, \texttt{schürze})$. Geben Sie die Berechnungsmatrix vollständig an. Wieviele Backtraces enthält die Berechnungsmatrix?

(b) Geben Sie zwei Alignments mit minimaler Levenshtein-Distanz zwischen den Wörtern `bürst` und `sch` an.