

# Datenkompression

Sommersemester 2015

## Übungsblatt 3

Prof. Dr. E. Ohlebusch

Institut für Theoretische Informatik

T. Beller

Ausgegeben am 26.05.2015

Besprechung am 02.06.2015

### Aufgabe 3.1

Die Zeichenfolge  $S := \text{ABABDCCCCAABABDCAABA}$  über dem Quellenalphabet  $\Sigma := \{\text{A, B, C, D}\}$  soll komprimiert werden.

- Verwenden Sie das LZ77-Verfahren (Suchpuffer 7 Zeichen und Codierpuffer 3 Zeichen lang). Welche Ausgabesymbole könnte man weglassen, um die Kompressionsrate zu verbessern?
- Verwenden Sie das LZSS-Verfahren (Suchpuffer 7 Zeichen und Codierpuffer 3 Zeichen lang). Ab welcher Länge sollte man hier (anstatt die Zeichen selbst zu codieren) das Format mit Offset und Länge verwenden?
- Verwenden Sie das LZ78-Verfahren.
- Welches der drei Verfahren komprimiert  $S$  am besten, welches am schlechtesten?

### Aufgabe 3.2

Konstruieren Sie eine Zeichenkette, die sich mit LZ77 besser komprimieren lässt als mit LZ78.

### Aufgabe 3.3

Wir betrachten einen LZW-Decodierer mit folgendem Anfangswörterbuch:

Index	Eintrag
1	a
2	b
3	c

- Decodieren Sie die Folge 1,2,3,6,4,8 mit diesem LZW-Decodierer.
- Werden in dem Trie eines LZW-Decodierers Zeiger auf Kindknoten benötigt? Werden Zeiger auf Elternknoten benötigt?
- Beschreiben Sie einen LZW-Decodierer, dessen Wörterbuch mit Hilfe einer Tabelle anstelle eines Tries implementiert ist. Welche Vorteile sehen Sie in solch einem Verfahren?

*Tipp:* Für die Folge 1,2,2,1,3,4,9,3 könnte die Tabelle wie folgt aussehen:

1	2	3	4	5	6	7	8	9	10
0	0	0	1	2	2	1	3	4	9
a	b	c	b	b	a	c	a	a	c

### Aufgabe 3.4

Diese Aufgabe bezieht sich auf den Skriptteil „Lempel-Ziv Factorization: LZ77 without Window“, der auf der Vorlesungshomepage und im Skriptdrucksystem verfügbar ist.

Das Suffix-Array (und die Suffixe) des Strings  $S := \text{ABABDCCCAABABDCAABA}\$$  ist in Abbildung 1 dargestellt.

- Berechnen Sie mit Algorithmus 4 die Arrays  $\text{PSV}_{\text{lex}}$  und  $\text{NSV}_{\text{lex}}$  für  $S$ .
- Berechnen Sie mit Algorithmus 5 die LZ-Faktorisierung von  $S$ . Wie viele Zeichenvergleiche werde benötigt?
- Zeigen Sie, dass die Laufzeit von Algorithmus 5 in  $O(n)$  liegt. Es wurde bereits gezeigt, dass die Arrays SA, ISA,  $\text{PSV}_{\text{lex}}$  und  $\text{NSV}_{\text{lex}}$  in linearer Zeit berechnet werden können.

i	SA	ISA	$S_{\text{SA}[i]}$
0	0		$\epsilon$
1	21	7	\$
2	20	12	A\$
3	17	9	AABA\$
4	10	14	AABABDCAABA\$
5	18	21	ABA\$
6	11	19	ABABDCAABA\$
7	1	18	ABABDCCCAABABDCAABA\$
8	13	17	ABDCAABA\$
9	3	16	ABDCCCAABABDCAABA\$
10	19	4	BA\$
11	12	6	BABDCAABA\$
12	2	11	BABDCCCAABABDCAABA\$
13	14	8	BDCAABA\$
14	4	13	BDCCCAABABDCAABA\$
15	16	20	CAABA\$
16	9	15	CAABABDCAABA\$
17	8	3	CCAABABDCAABA\$
18	7	5	CCCAABABDCAABA\$
19	6	10	CCCAABABDCAABA\$
20	15	2	DCAABA\$
21	5	1	DCCCAABABDCAABA\$

Abbildung 1: Suffix Array und Inversers Suffix Array des Strings  $\text{ABABDCCCAABABDCAABA}\$$