Datenkompression

Sommersemester 2016 — **Übungsblatt 3** — Prof. Dr. E. Ohlebusch

Institut für Theoretische Informatik — J. Lorenz

Ausgegeben am 19.05.2016 — Besprechung am 24.05.2016

Aufgabe 3.1

Die Zeichenfolge $S := \texttt{MISS MISSION MISSISSIPPI}$ über dem Quellenalphabet $\Sigma := \{\texttt{M}, \texttt{I}, \texttt{S}, \texttt{O}, \texttt{P}, \texttt{N}\}$ soll komprimiert werden. Die Leerzeichen dienen dabei nur der Formatierung und sollen nicht mitkodiert werden.

a) Verwenden Sie das LZ77-Verfahren (Suchpuffer 7 Zeichen und Codierpuffer 3 Zeichen lang). Welche Ausgabesymbole könnte man weglassen, um die Kompressionsrate zu verbessern?

b) 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?

c) Welches der beiden Verfahren komprimiert S besser?

Aufgabe 3.2

Diese und die folgenden Aufgaben beziehen sich auf den Skriptteil „Lempel-Ziv Factorization: LZ77 without Window“, der auf der Vorlesungshomepage verfügbar ist. Geben Sie für die folgenden Probleme jeweils einen Algorithmus an, der einen Stack benutzt und dessen Laufzeit linear ist.

1. Einen Algorithmus, der NSV_{lex} bestimmt.
2. Einen Algorithmus, der PSV_{lex} bestimmt.
3. Einen Algorithmus, der sowohl NSV_{lex} als auch PSV_{lex} in einem Schleifendurchlauf bestimmt.

Aufgabe 3.3

Wo liegen mögliche Vor- bzw. Nachteile von Algorithmus 7 im Vergleich mit Algorithmus 5?

Aufgabe 3.4

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

a) Berechnen Sie mit Algorithmus 4 die Arrays $\mathsf{PSV}_{\text{lex}}$ und $\mathsf{NSV}_{\text{lex}}$ für S.

b) Bestimmen Sie $\mathsf{PSV}_{\text{text}}$ und $\mathsf{NSV}_{\text{text}}$.

c) Berechnen Sie mit Algorithmus 5 die LZ-Faktorisierung von S. Wie viele Zeichenvergleiche werden benötigt?

d) Zeigen Sie, dass die Laufzeit von Algorithmus 5 in $O(n)$ liegt. Es wurde bereits gezeigt, dass die Arrays SA, ISA, $\mathsf{PSV}_{\text{lex}}$ und $\mathsf{NSV}_{\text{lex}}$ in linearer Zeit berechnet werden können.

i	SA	ISA	$S_{\mathsf{SA[i]}}$
0	0		ϵ
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	ABABDCCCCAABABDCAABA$
8	13	17	ABDCAABA$
9	3	16	ABDCCCCAABABDCAABA$
10	19	4	BA$
11	12	6	BABDCAABA$
12	2	11	BABDCCCCAABABDCAABA$
13	14	8	BDCAABA$
14	4	13	BDCCCCAABABDCAABA$
15	16	20	CAABA$
16	9	15	CAABABDCAABA$
17	8	3	CCAABABDCAABA$
18	7	5	CCCAABABDCAABA$
19	6	10	CCCCAABABDCAABA$
20	15	2	DCAABA$
21	5	1	DCCCCAABABDCAABA$

Abbildung 1: Suffix Array und Inversers Suffix Array des Strings ABABDCCCCAABABDCAABA$