

Datenkompression

Sommersemester 2019

Übungsblatt 2

Prof. Dr. Enno Ohlebusch

Institut für Theoretische Informatik

Uwe Baier

Ausgegeben am 11.06.2019

Besprechung am 18.06.2019

Aufgabe 2.1

- (a) Leiten Sie den ursprünglichen String S der Burrows-Wheeler-Transformierten $L = \text{yee}\$yaass$ her, d.h. invertieren Sie die Transformation.
Hinweis: S endet mit dem lexikographisch kleinsten Zeichen \$.
- (b) Bilden Sie die Move-To-Front-Transformierte des Strings $L = \text{yee}\$yaass$.
- (c) Sei S ein String und S_M die Move-To-Front-Transformierte von L . Seien Σ bzw. Σ_M die Mengen aller in S bzw. S_M vorkommenden Zeichen. Zeigen oder widerlegen Sie folgende Aussagen:
- (1) Es gilt immer $H(S) \leq H(S_M)$. (3) Es gilt immer $|\Sigma| \leq |\Sigma_M|$.
(2) Es gilt immer $H(S) \geq H(S_M)$. (4) Es gilt immer $|\Sigma| \geq |\Sigma_M|$.

Aufgabe 2.2

- (a) Kodieren Sie die nachfolgenden Zahlen mit den angegebenen Verfahren:
- 46 mit Elias-Epsilon-Code
 - 121 mit Rice-Code und $p = 16$
 - 18 mit Fibonacci-Code
- (b) Die folgende Bitsequenz besteht aus den konkatenierten Kodierungen von 5 Zahlen:

00000010110100010011000001101010000000110001011

Dekodieren Sie die 5 Zahlen, die mit den folgenden Verfahren kodiert wurden:

Zahl 1: Elias-Gamma Code

Zahl 4: Comma-Code mit $p = 2$

Zahl 2: Elias-Delta Code

Zahl 5: Fibonacci-Code

Zahl 3: Rice-Code mit $p = 16$

- (c) Warum wird für das Kodieren von Runs in einer BWT keiner der oben genannten Codes benutzt? Begründen Sie Ihre Antwort!

Aufgabe 2.3

Schreiben Sie ein Programm, das als Eingabe eine Datei mit der Move-To-Front-Transformierten eines Textes erhält und als Ausgabe die Move-To-Front Transformierte mit Lauflängenkodierung der Nullen ausgibt.

Laden Sie sich als Testdatei die Datei unter https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Lehre/SS19/bible.bwt.mtf herunter.¹ Bestimmen Sie die Größe der Huffmankodierung der Testdatei einmal mit und einmal ohne Lauflängenkodierung der Nullen.²

¹ Dies ist die Move-To-Front Transformierte der Burrows-Wheeler-Transformierten der Bibel, siehe <http://www.data-compression.info/Corpora/CanterburyCorpus/index.html>

²Die Huffmankodierung kann z.B. mit dem Programm `pack` berechnet werden, siehe <https://www.computerhope.com/unix/upack.htm>.