Data Compression Techniques

Separate Exam, 16 November 2012 at 16-20 Lecturer: Juha Kärkkäinen

Please write on each sheet: your name, student number or identity number, signature, course name, exam date and sheet number. You can answer in English, Finnish or Swedish.

1. [*3+3+3+3 points*] Define the following concepts:

   (a) run-length encoding

   (b) LF-mapping

   What is the *main difference* between the following pairs:

   (c) Kraft's inequality versus McMillan's inequality

   (d) grammar compression versus Lempel–Ziv compression

   A few lines for each part is sufficient.

2. [*12 points*] Let $\{\texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}\}$ be the alphabet with the probability distribution

symbol	a	b	c	d
probability	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{8}$	$\frac{1}{8}$

   A string T of length 3 has been encoded using exact arithmetic coding and the resulting code is `100010`. What is T? Give the intermediate steps in the decoding process.

3. [*12 points*] Describe the basic principles of adaptive compression models. What are the strengths and weaknesses compared to semiadaptive models? Be as complete as possible.

4. [*6+6 points*] Let $L =$ `oydbbbbaaaad$` be the Burrows–Wheeler transform for a text T. The order of the symbols is `$` $<$ `a` $<$ `b` $<$ `d` $<$ `o` $<$ `y`. The last character of T is `$`.

   (a) What is T?

   (b) Give a Huffman wavelet tree of L.

   Give enough intermediate steps to show how you arrived at the solutions.

5. [*12 points*] Let S be a set of n integers in the range $U = [0..u)$. We want to store S so that the following operations are supported:

   - predecessor(i) is the largest element of S that is smaller than i. If S contains no elements smaller than i, the result is the smallest element in S.
   - successor(i) is the smallest element of S that is larger than i. If S contains no elements larger than i, the result is the largest element in S.

   For example, if $U = [0..100)$ and $S = \{7, 19, 20, 56, 65\}$, then predecessor(5) = 7, predecessor(20) = 19, successor(5) = 7 and successor(20) = 56.

   Describe how to store S in $u + o(u)$ bits so that predecessor and successor operations are supported in constant time. Give a pseudocode for the operations.