Data Compression Techniques

Renewal/Separate Exam, 13 April 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. [*2+2+2+2+2 points*] Define the following concepts:

   (a) zeroth order empirical entropy

   (b) grammar compression

   (c) balanced parentheses sequence

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

   (d) LZW versus original LZ78

   (e) adaptive versus semiadaptive compression model

   A few lines for each part is sufficient.

2. [*5+5 points*] Consider the following prefix code:

symbol	a	b	c	d	e	f	g	h	i
code	1010	111	00101	000	01	1011	110	0011	00100

   (a) Show that the code is redundant, i.e., satisfies Kraft's inequality with strict inequality.

   (b) Modify the code by deleting some bits in the codewords so that the result is a complete prefix code, i.e., satisfies Kraft's inequality with equality. You may not add or change any bits, only delete them.

3. [*10 points*] Let $\{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$ be the alphabet with the probability distribution

symbol	a	b	c	d
probability	0.4	0.2	0.1	0.3

   Encode the string "bad" as a binary sequence using *exact* arithmetic coding. Give the intermediate steps in the encoding process. You may assume that the length of the string is known and does not need to be encoded.

4. [*10 points*] What are the properties of the Burrows–Wheeler transform that make it a useful tool for higher order text compression? Make your answer as complete as possible. Use examples to illustrate your answer.

5. [*10 points*] Let M be a $n \times n$ sparse matrix that contains m non-null entries. The non-null entries are integers from the interval $[0..\sigma)$. Design a compressed representation for M. The representation should support the following operations:

   - access(i, j) returns the value at $M[i, j]$ (which may be null). The time complexity should be constant.
   - row(i) returns all non-null values on the row i. The time complexity should be $\mathcal{O}(k + 1)$, where k is the number of values returned.
   - column(j) returns all non-null values on the column j. The time complexity should be $\mathcal{O}(k + 1)$, where k is the number of values returned.

   The space complexity should be as small as possible. You may use any of the compressed data structures described on the lectures.