Arithmetic Coding with Improved Solution for the Carry-over Problem

Xiaohui Xue and Wen Gao
(Dept. of Computer Science, Harbin Institute of Technology, Harbin, 150001, China)
{xhxue, wgao}@jdl.mcel.mot.com

The carry-over problem is inherent in arithmetic coding as a result of using finite precision arithmetic. As far as we know, the currently widely used solution for this problem is the bit-stuffing technique, which was proposed by Rissanen and Langdon[1]. However, this technique is not completely satisfactory. The stuffed-bit affects the coding efficiency slightly. The code stream inserted with several stuffed-bits loses its concept as a real number. This conflicts with the principle that arithmetic coding maps an input stream to an interval on the real line, which is neither perfect nor convenient for analysis. We present our solution for the carry-over problem, carry-trap technique, which works without the deliberately inserted stuffed-bit. We also present a concise termination method, named medium termination technique. Both are proved rigorously.

We use some notations of [1], and denote \( C'(s) = C(s) + A(s) \), \( I_n = [C(s), C'(s)] \). For the first time, we prove the following analytic properties of arithmetic coding, which play important roles in our analysis.

(The nested intervals property of arithmetic coding) \( \{I_n\} \) constructs a nest of intervals, namely, \( I_{n+1} \subset I_n \).

(The Convergence of Arithmetic Coding) As \( |s| \) increases, \( C(s) \) increases monotonically and is bounded from above, \( C'(s) \) decreases monotonically and is bounded from below, \( C(s) \) and \( C'(s) \) converge to the same limitation \( \lim_{|s| \to \infty} C(s) = \lim_{|s| \to \infty} C'(s) \).

Carry-trap technique initializes boolean variable \( SetTrap \) to be false and clear \( trap_n \). The process of carry-over is as follows: output one bit 1; if \( trap_n > 0 \), then output \( trap_n - 1 \)'s 0 and clear \( SetTrap \); if \( trap_n = 0 \), then set \( SetTrap \) to be false.

The process of shift is based on \( (SetTrap, trap_n) \) and the value of the shift bit. We prove the following theorems:

Theorem 1. If only \( SetTrap \) is true, the carry-trap technique can process correctly.

Theorem 2. When a carry-over occurs, \( SetTrap \) must be true.

Medium termination technique works as follows: \( C = C + \alpha/2 \), where \( \alpha = 0.100... \); if a carry-over occurs, output one bit 1 and \( trap_n + 2 \)'s 0, then exit; if \( SetTrap \) is true, then output one bit 0 and \( trap_n \)'s 1; output the first two most significant bits of \( C \).

With the carry-trap technique we can always get a pure code stream without stuffed-bits, and the decoder need not process carry-overs, yielding elegant structure. The concise medium termination technique is related to the concept of the trap-bit. The termination method of CACM87 is as concise as ours, which is related to its incremental transfer mechanism.

References


476 1068-0314/97$10.00 © 1997 IEEE