Memory Efficient Hierarchical Lookup Tables for Mass Arbitrary-Side Growing Huffman Trees Decoding

Sung-Wen Wang, Ja-Ling Wu, Shang-Chih Chuang, Chih-Chieh Hsiao, Yi-Shin Tung

APPENDIX

COMPLEXITY ANALYSIS

We compare the complexity of the proposed method with that of the brute-force top-down approach. For a tree with depth \( n \), let \( H_n \) be the number of hierarchical MHC partitions, \( B_n \) be the number of hierarchical Bpx partitions, and \( A_n \) be the number of HASGTs. Since the brute-force top-down approach considers all possible hierarchical partitions, \( A_n \) can, therefore, stand for the complexity of the brute-force approach. The proposed Viterbi-like method only searches \( op \)'s in the \( OP_i \) for each \( t_i \), hence the complexity of the proposed method is bounded by summing the complexity of all \( OP_i \)'s up. Let \( \hat{H}_d \), \( \hat{B}_d \) and \( \hat{A}_d \) be the numbers of \( OP_i \)'s which are respectively constrained by MHC, Bpx and HASGT for \( t_i \) with depth \( d \). Consequently, the complexity of the Viterbi-like method can be formulated as

\[
\hat{A}_n = \sum_d \hat{A}_d.
\] (A-1)

A. Perfect binary tree

A perfect binary tree (PBT) is a complete binary tree in which all leaves are with the same depth. It is trivial that only one MHC partition is applicable for a depth-1 PBT, that is

\[
H_1 = 1.
\] (A-2)

For a depth-2 PBT, two first-order MHC’s are considered: 1) MHC with length-1 and 2) MHC with length-2. For the first case, the number of hierarchical MHC partitions of two depth-1 PBTs decides the number of hierarchical MHC partitions of a depth-2 PBT. For the second case, there is only one MHC partition. Thus, the number of hierarchical MHC partitions of depth-2 PBT can be represented by

\[
H_2 = H_1^2 + 1.
\] (A-3)
Similarly, the number of hierarchical MHC partitions of depth-3 PBT is

\[ H_3 = H_2^2 + H_1^4 + 1, \]  

(A-4)

From Eqs. (A-2) to (A-4), we can derive the following recursive formula:

\[ H_n = \sum_{i=1}^{n-1} H_i^{2^{n-i}} + 1. \]  

(A-5)

From the relation between \( H_n \) and \( H_{n-1} \) one can be obtained the following inequality

\[ H_{n-1}^2 \leq H_n \leq (H_{n-2}^2 + H_{n-3}^2 + \cdots + H_1^{2^{n-2}})^2 = 2H_{n-1}^2. \]  

(A-6)

\( H_n \) can now be lower bounded by recursive substitution, and the result is

\[ 2^{n-2} \leq H_n. \]  

(A-7)

There are two Bpx partitions, \((0)_b\) and \((1)_b\) for a depth-1 PBT, that is

\[ B_1 = 2. \]  

(A-8)

Consider two kinds of first-order Bpx, length-1 bit-patterns (i.e. \((0)_b\) and \((1)_b\)) and length-2 bits patterns (i.e. \((00)_b\), \((01)_b\), \((10)_b\) and \((11)_b\)). According to these bits-patterns, The number of hierarchical Bpx partitions of depth-2 PBTs can be formulated as

\[ B_2 = 2^2 \times B_1 + 2^1 \times B_1 \times B_1 = 4B_1^2. \]  

(A-9)

Similarly, the number of partition methods for a depth-3 PBT can be obtained as

\[ B_3 = 2^3 \times B_2 \times B_1 + 2^2 \times B_2 \times B_1 \times B_1 + 2^1 \times B_2 \times B_2 = 4B_2^2 \]  

(A-10)

From Eqs. (A-8) to (A-10), we can derive the following recursive formulation:

\[ B_n = 4B_{n-1}^2. \]  

(A-11)

After recursive substitution for (A-11), we obtain

\[ B_n = 2^{2n-2} + 2^{n-1}. \]  

(A-12)

Consequently, for an HASGT, \( A_n \) can be formulated by \( H_n \) and \( B_n \) as

\[ A_n = \sum_{i=1}^{n} B_i(A_{n-i})^2 + \sum_{i=1}^{n} H_i(A_{n-i})^2 = \sum_{i=1}^{n} (B_i + H_i)(A_{n-i})^2. \]  

(A-13)

Substituting (A-7) and (A-12) into (A-13), we obtain
\[ A_n \geq \sum_{i=1}^{n} (2^{2^i-2+2^{i-1}} + 2^{2^{i-1}})(A_{n-i})2^i \geq 2^{2^n-2+2^{n-1}} + 2^{n-2}. \] (A-14)

Subsequently, we analyze the complexity of the proposed method. Let \( \|OP\|_d \) be the number of operations of a tree with depth \( d \). By definition, \( \|OP\|_d \) is the sum of the number of MHC partitions methods, denoted as \( \|MHC\|_d \), and the number of Bpx partitions methods, denoted as \( \|Bpx\|_d \). That is

\[ \|OP\|_d = \|MHC\|_d + \|Bpx\|_d, \] (A-15)

in which

\[ \|MHC\|_d = d \] (A-16)

and

\[ \|Bpx\|_d = 2^d + (2^d - 1) - 1 = 2^{d+1} - 2. \] (A-17)

By summing up all possible operations of internal subtrees, we obtain

\[
\hat{A}_n = \sum_{i=1}^{n} 2^{n-i} \cdot \|OP\|_i = 2^n \sum_{i=1}^{n} \frac{2^{i+1}-2^i}{2^i} = 2^n \sum_{i=1}^{n} \left( 2 - \frac{1}{2^{i-1}} + \frac{1}{2^i} \right)
= 2^n \sum_{i=1}^{n} \left( \frac{1}{2} \right) + 2n \cdot 2^n - 2^{n+1} + 2. \] (A-18)

To compute the term \( 2^n \sum_{i=1}^{n} \left( \frac{1}{2^i} \right) \) in Eq. (A-18), we introduce the following reduction formula:

\[ 1 \cdot 2^{n-1} + 2 \cdot 2^{n-2} + 3 \cdot 2^{n-3} + \cdots + n \cdot 2^0. \] (A-19)

Eq. (A-19) can be generated by the multiplication of the generating functions \( \frac{1}{(1-Z)^2} \) and \( \frac{1}{(1-2Z)} \), that is, Eq. (A-19) can be calculated by taking the inverse z-transform of the term

\[ [z^{n-1}] \frac{1}{(1-Z)^2} \times \frac{1}{(1-2Z)}. \] (A-20)

Consequently, the complexity of the proposed method becomes

\[
\hat{A}_n = -(n-1) - 3 + 4 \cdot 2^{n-1} + 2n \cdot 2^n - 2^{n+1} + 2
= 2n \cdot 2^n - n. \] (A-21)

From Eqs. (A-18) and (A-21), we proved that

\[ \hat{A}_n << A_n. \] (A-22)
B. SGH-Tree

Following similar derivation processes presented above, for an SGH-Tree the relationship between $H_n$ and $H_{n-1}$ can be written as

$$H_n = 2H_{n-1}, \quad (A-23)$$

and the relationship between $B_n$ and $B_{n-1}$ becomes

$$B_n = 3B_{n-1}. \quad (A-24)$$

By recursive substitution, we obtain

$$H_n = 2^{n-1}, \quad (A-25)$$

and

$$B_n = 3^{n-1} \cdot B_1 = 2 \cdot 3^{n-1}. \quad (A-26)$$

Consequently, we obtain

$$A_n = \sum_{i=1}^{n} (2 \cdot 3^{i-1} + 2^{i-1})(A_{n-i})^{2^i} \geq 2 \cdot 3^{n-1} + 2^n. \quad (A-27)$$

In sequel, $\|MHC\|_d$ and $\|Bpx\|_d$ in Eq. (A-15) respectively are

$$\|MHC\|_d = d \quad (A-28)$$

and

$$\|Bpx\|_d = (d + 1) + (d - 1) = 2d. \quad (A-29)$$

Thus

$$\hat{A}_n = \sum_{i=1}^{n} \|OP\|_i = \sum_{i=1}^{n} 3i = \frac{3n(n + 1)}{2}. \quad (A-31)$$

Finally, from Eqs. (A-27) and (A-31), we proved again that

$$\hat{A}_n \ll A_n. \quad (A-32)$$