ISS-P-060

可逆ハフマン符号化のゴミ出力量の最適化

田島 嘉人 †　　横山 哲郎 ††
† 南山大学大学院理工学研究科　　†† 南山大学理工学部

1　はじめに

消費エネルギー最適化や量子コンピュータの実現に用いられる可逆計算法 [2] が研究されている．可逆計算法は計算の全過程の単射化により実現される．ハフマン符号化法と漸近的に同じ時間及び空間計算量でゴミ出力量 $\Theta(n \lg n)$ の可逆アルゴリズム [1] が報告されている．ゴミ出力は，計算を単射化するために出力される情報であり，問題の本質的な解決に関わらない．本稿では，整列された頻度表を入力とするハフマン木構築法 [3] を基礎とする，ゴミ出力がなく 1 パスの可逆ハフマン木構築法を提案する．

2　可逆で情報損失がないハフマン木構築法

本稿では入力文字列中の文字頻度表 C が整列されている場合を考える．C を入力としてハフマン木を出力とする関数は単射である．この関数を可逆プログラミング言語 ROOPL++ で記述した可逆なプログラム 1 を示す．ここで C は実行後の値が nil であり，root は初期値が nil で実行後の値がハフマン木の根への参照である．

プログラム 1　可逆ハフマン木構築法

```
 1 method huffmanTree(Queue C, Node root)
 2     local int n = C.length // 要素数
 3     local Queue Q = nil // 節点を格納する待ち行列
 4     new Queue Q
 5     local int i = 0
 6
 7     from i = 0 loop
 8         local Node node = nil // ハフマン木に追加する節点
 9         new Node node
10         if C.head.value.freq < Q.head.value.freq then
11             call C::dequeue(node.left) // 左の子
12         else
13             call Q::dequeue(node.left) // 左の子
14         fi node.left.left = nil && node.left.right = nil
15         if C.head.value.freq < Q.head.value.freq then
16             call C::dequeue(node.right) // 右の子
17         else
18             call Q::dequeue(node.right) // 右の子
19         fi node.right.left = nil && node.right.right =
               nil
20         node.freq ^= node.left.freq + node.right.freq
21         call Q::enqueue(node) // キューに追加
22         delocal Node node = nil
23         i += 1
24     until i = n − 1
25
26     delocal int i = n − 1
27     call Q::dequeue(root)
28     delete Queue Q
29     delocal Queue Q = nil
30     uncall root::leaf(n) // ゼロクリア
31     delocal int n = 0
32     delete Queue C // ゼロクリア
```

C は待ち行列により実現され先頭の要素への参照 head と要素数 length をもつ．また，C の各要素は Node 型の値 value をもつハフマン木の節点を一時的に格納するために待ち行列 Q を用いる．プログラム 7–27 行目のループでは，ハフマン木の新節点 node が作成され，Q に追加される．11–15 行目で，C の先頭の要素と Q の先頭の要素のより小さい要素がデキューされ，新節点の左の子となる．同様の処理を 17–21 行目において右の子について行う．ここで，節点に追加した子が葉であるかどうかの結合条件により制御流の結合後にどちらの分岐を辿ったか一意に定まることに注意して欲しい．ループ終了後，Q の要素はただ一つであり，その値はハフマン木の根への参照である．33 行目の leaf は，ハフマン木の葉の個数を n の値から引く．leaf は，時間計算量 $\Theta(n)$，入力と出力を除く空間計算量 $\mathrm{O}(n)$，ゴミ出力ゼロで実行される．プログラム 1 は，時間計算量 $\Theta(n)$，入力と出力を除く空間計算量 $\mathrm{O}(n)$，ゴミ出力ゼロで動作する．

プログラム 1 において，可逆性制約を満たすことが自明でないキューの更新について考える．入力される文字の頻度表が整列済であるという仮定から $y_i \leq y_{i+1}$ $(1 \leq i \leq n)$ は常に成り立つ．キュー Q の末尾の根の値はキュー C の先頭の 2 つの値の和以下であるという条件 (*) が常に成り立つ．なぜなら，キューの更新には x_1, x_2, y_1, y_2 のうち 1 番目と 2 番目に小さいもので次の 3 つの場合に分けられキューの更新後にも条件 (*) が成り立つからである:

x_1, x_2 の場合　Q の末尾の根の値は $x_1 + x_2$, C の先頭 2 要素は y_1 と y_2 であり，x_1 と x_2 が 1 番目と 2 番目に小さいので $x_1 + x_2 \leq y_1 + y_2$

y_1, y_2 の場合　Q の末尾の根の値は $y_1 + y_2$, C の先頭 2 要素は y_3 と y_4 であり，整列済の条件より，$y_1 + y_2 \leq y_3 + y_4$

x_1, y_1 の場合　$x_1 \leq y_2 \leq y_3$ より $x_1 + y_2 \leq y_2 + y_3$

3　おわりに

本稿では，入力文字頻度表が整列済の場合，時間計算量 $\Theta(n)$，入力と出力を除く空間計算量 $\mathrm{O}(n)$，ゴミ出力なしでハフマン木を構築する方法を考案した．この方法の正当性検証は今後の課題である．

謝辞　JSPS 科研費 18K11250 及び 2021 年度南山大学パッヘ研究奨励金 I-A-2 の助成を受けた．

参考文献

[1] Hay-Schmidt, L.: Investigating the Reversibilization of Irreversible Algorithms, Master's thesis, University of Copenhagen (2021).

[2] Perumalla, K.S.: *Introduction to reversible computing*, CRC Press (2013).

[3] Van Leeuwen, J.: On the Construction of Huffman Trees, ICALP, pp.382–410 (1976).