arXiv:2301.02878v1 [cs.IT] 7 Jan 2023

摘要 霍夫曼编码与 PIFO 树嵌入

Keri D'Angelo* Dexter Kozen†

康奈尔大学计算机科学系，纽约州伊萨卡 14853-7501，美国

*kd349@cornell.edu †kozen@cs.cornell.edu

2023年1月10日

摘要

用于推导霍夫曼编码的算法，以及近期开发的将 PIFO 树编译为固定形状树的算法 [1] 彼此相似，但采用了不同的底层代数运算。本文利用前缀码的单子结构，构造了一种广义 Huffman 算法，并将这两种应用作为其特例。

1 引言

霍夫曼编码将固定字母表中的字母映射为d元码字，从而针对给定的字母频率分布实现最优压缩。已有一种广为人知的贪心算法，可根据给定的分布生成霍夫曼编码（见 [2]）。

一种名为*PIFO* 树（优先级先进先出）的新型数据结构，近期被提出用于在可编程网络路由器中实现多种数据包调度算法 [3, 4]。PIFO 树是一种由优先级队列构成的树。目前，大多数路由器仅支持少数几种调度算法，例如严格优先级或加权公平排队；这些算法都固化在硬件中。调度器可以在一定程度上进行配置，但通常无法实现更复杂的调度算法，因为这类算法需要对已经排队的数据包重新排序。而这正是 PIFO 树所能实现的功能。PIFO 很可能会在不久的将来得到网络设备的支持。

一些研究者已经开始探索如何在传统路由器上模拟 PIFO 抽象 [4]。在最近的一项工作中 [1], , 研究者展示了如何将针对任意形状 PIFO 树设计的算法，转换为使用固定形状 PIFO 树的算法；这种树可以是可能由硬件实现的完整d叉树，且性能下降可以忽略不计。

嵌入算法采用贪心策略，与霍夫曼算法非常相似，不同之处在于它基于不同的代数运算。对于霍夫曼编码，我们希望选择一个d叉前缀码C，使$\sum_{x\in C} |x| \cdot r(x)$的值最小，其中$r(x)$是分配给码字$x$的字母频率。这样可以使生成的编码的熵最小。对于 PIFO 树，我们希望最小化 $\max_{x\in C} |x| + r(x)$，其中$r(x)$是子树的高度。这样可以使生成的d叉树的高度最小，并决定嵌入是否可行。

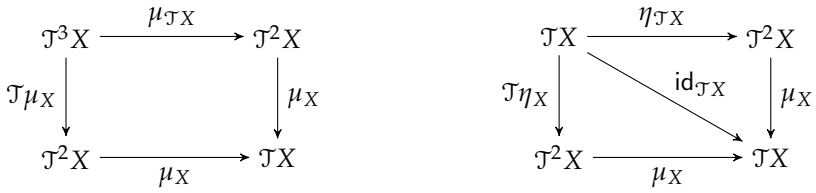
这种相似性促使我们寻求一种统一的公理化处理，它以代数运算为参数，并可通过实例化将两种应用都作为特例导出。我们的处理利用前缀码的单子结构，得到问题及其解法的抽象表述。我们确定了足以使抽象算法产生最优解的条件，其中最优 的含义也取决于具体实例化。

我们在§3中陈述保证最优性的公理。算法见§4，其正确性证明见§5。霍夫曼编码和 PIFO 树这两种应用将在§6中推导。

2 背景

我们假定读者熟悉范畴、函子和自然变换等基本的范畴论概念。本文的讲解基于单子 和艾伦伯格–摩尔代数的概念；此处简要回顾相关定义。更全面的介绍请参阅 [5–8]。

单子 在函数式编程中被广泛用于为计算建模，使其附加额外结构 [9–11]。形式上，范畴 C 上的单子 是一个三元组$(\mathcal{T}, \eta, \mu)$，其中 $\mathcal{T}\colon \mathsf{C} \to \mathsf{C}$ 是 C 上的自函子，而 $\eta : I \to \mathcal{T}$ 和 $\mu\colon \mathcal{T}^2 \to \mathcal{T}$ 是自然变换，分别称为单位和乘法；对所有对象X，下列图表都交换：



单子的典型示例包括

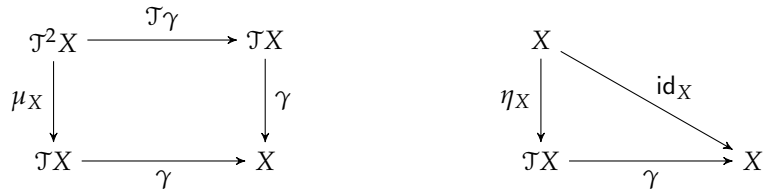
- 列表单子，其中 $\eta_X(a) = [a]$，是包含 a 的单元素列表，以及

  $$\mu_X([[a_{11},\ldots,a_{1k_1}],\ldots,[a_{n1},\ldots,a_{nk_n}]]) = [a_{11},\ldots,a_{1k_1},\ldots,a_{n1},\ldots,a_{nk_n}],$$

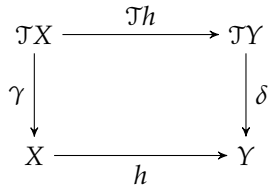
  列表扁平化操作；

- 幂集幂集单子，其中 $\eta_X(a) = \{a\}$ 表示包含 a 的单元素集合，而 $\mu_X(\mathcal{A}) = \bigcup \mathcal{A}$ 表示将 X 的子集的集合映射为其并集的操作。

给定范畴 C 上的一个单子$(\mathcal{T}, \eta, \mu)$，其上的一个***Eilenberg-Moore*** 代数 是一个有序对(X, γ)，其中X 是 C 的一个对象，而 $\gamma : \mathcal{T}X \to X$ 是 C 中的一个态射，称为该代数的结构映射，并且满足以下图交换：



Eilenberg-Moore 代数的态射 是 C 中与结构映射交换的态射。也就是说，若(X, γ) 和(Y, δ) 是两个代数，且$h : X \to Y$ 是 C 中的一个态射，那么当以下图交换时，h 就是代数之间的态射$h : (X, \gamma) \to (Y, \delta)$:



单子$(\mathcal{T}, \eta, \mu)$ 上的 Eilenberg-Moore 代数及其态射构成该单子 $\mathcal{T}$ 上的*Eilenberg-Moore*范畴。列表单子的 Eilenberg-Moore 范畴是由幺半群和幺半群同态构成的范畴。幂集单子的 Eilenberg-Moore 范畴是由完备上半格和半格同态构成的范畴。

在我们的应用中，我们将重点关注集合及集合函数范畴 Set 上的d 进制前缀码 单子。

3 公理

本节将陈述足以保证我们的广义 Huffman 算法最优性的公理。

回顾一下，在固定的d进制字母表 Σ 上，前缀码 是由Σ 上的有限长度词组成的集合，其中任意两个元素关于前缀关系均不可比较。若每个无限d进制字符串都以C中的某个元素为前缀，则前缀码C 称为完备的 。根据 K¨onig 引理，有限字母表上的每个完备前缀码都是有限的，但并非每个有限前缀码都是完备的。

设 $\mathcal{C} : \mathrm{Set} \to \mathrm{Set}$ 是一个自函子，其中

- $\mathcal{C}X$ 是满足以下条件的配对集合(C, r)： C 是某个任意但固定的d元字母表上的前缀码，且$r \geq 2$ 和$r : C \to X$，并且

- 对于 $h\colon X \to Y$，$\mathcal{C}h\colon \mathcal{C}X \to \mathcal{C}Y$，其中 $\mathcal{C}h(C, r) = (C, h \circ r)$。

函子 $\mathcal{C}$ 带有自然的单子结构，其单位为 $\eta : I \to \mathcal{C}$，乘法为$\mu\colon \mathcal{C}^2 \to \mathcal{C}$，定义如下：对于$a \in X$ 以及$(C,r) \in \mathcal{C}^2X$，其中$r(x) = (C_x,r_x)$，

$$\eta_X(a) = (\{\varepsilon\}, \varepsilon \mapsto a) \qquad \mu_X(C,r) = (\{xy \mid x \in C,\ y \in C_x\}, xy \mapsto r_x(y)).$$

映射 $xy \mapsto r_x(y)$ 定义良好，因为字符串 xy 可以唯一地拆分为 $x \in C$ 和$y \in C_x$，这是因为 C 是一个前缀码。

例如，考虑二元字母表 $\{0,1\}$ 上的前缀码 $C = \{0,10,110,111\}$ 和 $C_0 = C_{10} = C_{110} = C_{111} = \{00,11\}$。代码 C 是完备的，而其他代码不是。令

$$\begin{array}{llll} r_0(00) = 2 & r_{10}(00) = 4 & r_{110}(00) = 6 & r_{111}(00) = 8 \\ r_0(11) = 3 & r_{10}(11) = 5 & r_{110}(11) = 7 & r_{111}(11) = 9 \\ r(0) = (C_0, r_0) & r(10) = (C_{10}, r_{10}) & r(110) = (C_{110}, r_{110}) & r(111) = (C_{111}, r_{111}). \end{array}$$

于是 $(\mathbf{C}_0, \mathbf{r}_0)$、$(\mathbf{C}_{10}, \mathbf{r}_{10})$、$(\mathbf{C}_{110}, \mathbf{r}_{110})$、$(\mathbf{C}_{111}, \mathbf{r}_{111}) \in \mathcal{C}\mathbb{N}$ 和 $(\mathbf{C},\mathbf{r}) \in \mathcal{C}^2\mathbb{N}$，以及 $\mu_{\mathbb{N}}(\mathbf{C},\mathbf{r}) = (\mathbf{C}', \mathbf{r}') \in \mathcal{C}\mathbb{N}$，其中

$$\begin{aligned} &C' = \{000, 011, 1000, 1011, 11000, 11011, 11100, 11111\} \\ &r'(000) = 2,\ r'(011) = 3,\ r'(1000) = 4,\ r'(1011) = 5, \\ &r'(11000) = 6,\ r'(11011) = 7,\ r'(11100) = 8,\ r'(11111) = 9. \end{aligned}$$

假设存在一个固定的 Eilenberg-Moore 代数(W, w)，其中 $w\colon \mathcal{C}W \to W$。我们称 W *weights* 中的元素为权重，并称(W, w) 为一个加权。若$(C, r) \in \mathcal{C}W$，则将C 的元素视为一棵树，映射$r\colon C \to W$ 为树的每个叶子赋予一个权重，而映射w 则说明如何根据叶子权重r为对象(C, r) 赋予权重。

为定义最优性的概念，我们假设W 在 $\leq$下是全预序的；也就是说，$\leq$ 具有自反性和传递性，并且对于所有x、$y \in W$，要么$x \leq y$，要么$y \leq x$（或二者皆成立）。在 $\leq$ 这一序下，W 的值越小，就被认为越好。如果$x \leq y$ 且$y \leq x$，我们记作$x \equiv y$。进一步假设我们在 $\mathcal{C}W$上有一个预序，同样记作 $\leq$，并满足以下性质。

(i) 如果$f : C \to D$ 是双射且长度非递减的，并且$r \leq s \circ f$ 逐点成立，那么$(C,r) \leq (D$、$s)$。这表示，更长的码字或更大的叶值不会导致序 $\leq$中的下降。

(ii)（交换性质）如果$r(x) \leq r(y)$，$|x| \leq |y|$，并且

$$s(z) = \begin{cases} r(x), & \text{if } z = y, \\ r(y), & \text{if } z = x, \\ r(z), & \text{if } z \in C \setminus \{x, y\}, \end{cases}$$

然后 $(C, s) \leq (C,r)$。也就是说，在树中将较大的元素向下移、与较高位置上的较小元素交换，绝不会有坏处。

(iii) 单子结构映射 $\eta_W : W \to \mathcal{C}W$ 和 $\mu_W : \mathcal{C}^2W \to \mathcal{C}W$ 关于 $\leq$ 是单调的，其中 $\leq$ 在 $\mathcal{C}^2W$ 上的定义如下：

$$(C, r) \leq (D, s) \Leftrightarrow \mathcal{C}w(C, r) \leq \mathcal{C}w(D, s).$$

（i）的一些特殊情形为

- 若 $f : C \to D$ 是双射且长度非递减，则 $(C, s \circ f) \leq (D, s)$。因此，延长码字不会导致 $\leq$ 减小。
- 若 $f : C \to D$ 是双射且长度保持，则 $(C, s \circ f) \equiv (D, s)$。这说明，树上的次序 $\leq$ 只取决于 C 中码字的长度，而与码字本身无关。
- 若 r、$s : C \to W$ 且 $r \leq s$ 逐点成立，则 $(C, r) \leq (C, s)$。因此，较大的叶值不会导致 $\leq$ 减小。

我们假设下一节所述算法满足这些性质。

对于 (C, r)、$(D, s) \in \mathcal{C}W$，若两个对象所表示的权重多重集相同，则记作 $(C, r) \sim (D, s)$；也就是说，存在一个双射函数 $f : C \to D$，使得 $r = s \circ f$。若一棵树 $(C, r) \in \mathcal{C}W$ 在其权重多重集意义下是最优的，则称其为最优树；也就是说，(C, r) 在其 $\sim$-类中是 $\leq$ 最小的；亦即，对于所有满足 $(C, r) \sim (D, s)$ 的 (D, s)，都有 $(C, r) \leq (D, s)$。

我们将在第6节中给出两个详细示例。

4 算法

假设给定了 M 中取自 W 的权重多重集，$|M| \geq 2$。我们希望为这个权重多重集找到一棵最优树。下面给出一种用于寻找此类最优树的递归算法。

1. 假设 $M \geq 2$ 中有 n 个元素。令 $k \in \{2, \ldots d\}$，使得 $n \equiv \mod k \equiv (d - 1)$。令 $a_0, \ldots a_{k-1}$ 为权重最小的 k 个元素。构造对象

$$(\{0, 1, \ldots, k-1\}, i \mapsto a_i) \in \mathcal{C}W.$$

如果 M 中没有其他元素，则返回该对象。

2. 否则，令

$$M' = \{(\{0, 1, \ldots, k-1\}, i \mapsto a_i)\} \cup \{\eta_W(a) \mid a \in M \setminus \{a_0, \ldots, a_{k-1}\}\},$$

一个包含 $\mathcal{C}W$ 中 $n - k + 1 < n$ 个元素的多重集。

3. 在步骤 1 中递归调用算法，输入为 $M'' = \{w(E, t) \mid (E, t) \in M'\}$，即 W 的元素组成的多重集。算法返回一个类型为 $\mathcal{C}W$ 的树 (D, s)，它对于 M'' 是最优的。双射映射 $s : D \to M''$ 可分解为 $w \circ s'$，其中 $s' : D \to M'$ 是某个双射映射；并且 $(D, s') \in \mathcal{C}^2W$，其中 $\mathcal{C}w(D, s') = (D, w \circ s') = (D, s)$。将其展平为 $\mu_W(D, s') \in \mathcal{C}W$，并返回该值。

请注意，除第一次递归调用外，在步骤 1 中合并的项目数均为 d。这是因为在每一步中，如果 $k \in \{2,3,\dots,d\}$，那么该步骤之后剩余元素的数量将为 $(c(d-1)+k)-k+1 = c(d-1)+1$，它模 $d-1$ 同余于 d，因此下一步将取出 d 个元素。但从那时起，递归过程中剩余元素的数量始终保持为模$d-1$ 同余于 1，因为每一步都会移除 d 个元素并加回一个，使数量减少 $d-1$。

5 正确性

本节将借助以下引理证明该算法的正确性。

引理 1. 设 $k \in \{2,3,\dots,d\}$，且 $k \equiv |M| \bmod (d-1)$。设 $a_0,\dots,a_{k-1}$ 为 M 中权重最小的 k 个元素，并按权重非递减顺序排列。在 $\mathcal{C}W$ 中存在一棵最优树，其中 $a_0,\dots,a_{k-1}$ 是位于最深层的兄弟叶节点，且没有其他兄弟节点。

证明。设 $(C, r) \in \mathcal{C}W$ 为最优解。公理（i）允许我们对 (C, r) 进行变换，使除最深层外的任何层级都没有亏格节点（子节点少于 d 个的节点），且最深层只有一个亏格节点。因此，不失一般性，我们可以假设 C 中有 k 个长度为 n 的最长元素 $x_0,\dots,x_{k-1} \in C$，它们具有长度为 $n-1$的公共前缀，并且其他 $y \in C$ 均不具有该前缀。设 $x_0,\dots,x_{k-1}$ 按 $r(x_i)$ 的非递减顺序排列；也就是说，对所有 $0 \le i \le j \le k-1$，都有 $r(x_i) \le r(x_j)$。令 $y_0,\dots,y_{k-1} \in C$ 满足 $r(y_i) = a_i$。由于 a_i 是最小的，因此 $r(y_i) \le r(x_i)$。由于$|x_i|$ 的长度最长，$|y_i| \le |x_i|$。现在我们可以根据公理（ii）进行交换。令

$$s(z) = \begin{cases} r(x_i), & \text{if } z = y_i, \\ r(y_i), & \text{if } z = x_i, \\ r(z), & \text{otherwise.} \end{cases}$$

那么 $(C, s) \le (C, r)$。但是，由于 (C, r) 是最优的，$(C,r) \equiv (C, s)$，而且 (C, s) 也是最优的。□

定理 2. *§4* 中的算法生成一棵最优树。

证明。 对 n 使用归纳法。基础情形为 $n \le d$，此时结论显然成立。

假设我们有一个由 W 中 n 个 $> d$ 元素组成的多重集 M。设 (C, r) 是 M 的一棵最优树。设 $k \in \{2,3,\dots,d\}$ 与 $|M|$ 模 $d-1$ 同余。设 $a_0,\dots,a_{k-1}$ 是 M 中最小的 k 个元素。根据引理 1，不失一般性地，我们可以假设 $a_0,\dots,a_{k-1}$ 互为兄弟，且位于 (C, r) 的最大深度处，因此存在长度最大的字符串 $x\,0, x\,1, \dots, x\,(k-1) \in C$，它们有公共前缀 x，且 $r(x\,i) = a_i$。从 C 中移除字符串 $x\,i$，并用 x 替换它们。将所得集合记为 C'。对于 $z \in C'$，令

$$r'(z) = \begin{cases} (\{0,1,\dots,k-1\}, i \mapsto a_i), & \text{if } z = x, \\ \eta_W(r(z)), & \text{otherwise.} \end{cases}$$

那么$(C', r') \in \mathcal{C}^2W$，且$(C, r) = \mu_W(C', r')$。$r'$的值所构成的多重集，正是算法第 2 步中的$M'$。

该算法将构造多重集

$$M'' = \{w(E, t) \mid (E, t) \in M'\} = \{w(r'(z)) \mid z \in C'\}$$

并使用这些权重递归调用。根据归纳假设，返回值将是一个树 $(\mathrm{D}, \mathrm{s}) \in \mathcal{C}\mathrm{W}$，它对于 M'' 是最优的，因此 $(\mathrm{D}, \mathrm{s}) \leq (\mathrm{C}', \mathrm{w} \circ \mathrm{r}')$，且双射映射 $\mathrm{s} : \mathrm{D} \to \mathrm{M}''$ 可分解为 $\mathrm{s} = \mathrm{w} \circ \mathrm{r}' \circ \mathrm{f}$，其中 $\mathrm{f} : \mathrm{D} \to \mathrm{C}'$ 是双射。令 $\mathrm{s}' = \mathrm{r}' \circ \mathrm{f}$。根据公理 (iii)，

$$\mathcal{C}w(D, s') = (D, w \circ s') = (D, s) \leq (C', w \circ r') = \mathcal{C}w(C', r'),$$

因此 $(D, s') \leq (C', r')$，并且由于 μ_W 是单调的，

$$\mu_W(D, s') \leq \mu_W(C', r') = (C, r).$$

由于 (C, r) 是最优的，因此 $\mu_W(D, s')$也是最优的，而这正是算法返回的值。 □

6 应用

通过选择两种特定的权重 (W, w) 并适当定义序关系 $\leq$，我们可以得到该算法的两个特殊情形。

6.1 霍夫曼编码

我们的第一个应用是霍夫曼编码。在这里，给定待编码字母的频率，我们希望最小化变长码字的期望长度。对于这一应用，我们取$W = \mathbb{R}_+ = \{a \in \mathbb{R} \mid a \geq 0\}$ 并进行加权

$$w(C, r) = \sum_{x \in C} r(x).$$

回顾一下，对于$a \in W$和$(C、r) \in \mathcal{C}^2W$，其中$r(x) = (C_x、r_x)$，

$$\eta_W(a) = (\{\varepsilon\}, \varepsilon \mapsto a) \qquad \mu_W(C, r) = (\{xy \mid x \in C,\ y \in C_x\}, xy \mapsto r_x(y)).$$

那么$(W、w)$是单子$(\mathcal{C}、\ \mu、\ \eta)$的 Eilenberg-Moore 代数，其中

$$\begin{aligned} w(\eta_W(a)) &= w(\{\varepsilon\}, \varepsilon \mapsto a) = \sum_{x \in \{\varepsilon\}} (\varepsilon \mapsto a)(x) = a, \\ w(\mu_W(C, r)) &= \sum_{x \in C} \sum_{y \in C_x} r_x(y) = \sum_{x \in C} w(C_x, r_x) \\ &= \sum_{x \in C} w(r(x)) = w(C, w \circ r) = w(\mathcal{C}w(C, r)). \end{aligned}$$

此外，我们定义 $\alpha :\ \mathcal{C}W \to W$ 为

$$\alpha(C, r) = \sum_{x \in C} |x| \cdot r(x).$$

引理 3。

$$\alpha(\eta_W(a)) = 0 \qquad\qquad \alpha(\mu_W(C,r)) = \alpha(C, w\circ r) + w(C, \alpha\circ r).$$

证明。

$$\begin{aligned}
\alpha(\eta_W(a)) &= \alpha(\{\varepsilon\}, \varepsilon\mapsto a) = \sum_{x\in\{\varepsilon\}} |x|\cdot(\varepsilon\mapsto a)(x) = |\varepsilon|\cdot a = 0,\\
\alpha(\mu_W(C,r)) &= \alpha(\{xy \mid x\in C,\ y\in C_x\}, xy\mapsto r_x(y))\\
&= \sum_{x\in C}\sum_{y\in C_x} |xy|\cdot r_x(y) = \sum_{x\in C}|x|\sum_{y\in C_x} r_x(y) + \sum_{x\in C}\sum_{y\in C_x}|y|\cdot r_x(y)\\
&= \sum_{x\in C}|x|\cdot w(C_x, r_x) + \sum_{x\in C}\alpha(C_x, r_x) = \sum_{x\in C}|x|\cdot w(r(x)) + \sum_{x\in C}\alpha(r(x))\\
&= \alpha(C, w\circ r) + w(C, \alpha\circ r).
\end{aligned}$$

□

注意，α 和w 在深度为一的树上是一致的：

$$w(\{0,1,\ldots,k-1\}, i\mapsto a_i) = \sum_{i=0}^{k-1} a_i,$$

$$\alpha(\{0,1,\ldots,k-1\}, i\mapsto a_i) = \sum_{i=0}^{k-1}|i|\cdot a_i = \sum_{i=0}^{k-1} a_i,$$

其中，$|i|$ 表示i 作为字符串的长度，在本例中为 1。

映射 α 与香农熵H有关。若$r(x) = d^{-|x|}$表示在d元字母表的均匀分布下，d元码字x 的概率，则

$$H(C,r) = \sum_{x\in C} -d^{-|x|}\log d^{-|x|} = \sum_{x\in C}|x|\cdot d^{-|x|}\log d = \alpha(C,r)\log d,$$

所以 $\alpha(C,r) = H(C,r)/\log d$.

要使用第 §4 节中的算法，我们需要在 $\mathcal{C}W$上定义一个序 $\le$。若$(C,r) \le (D,s)$ 当$(C,r) \sim (D,s)$时，也就是说，存在一个双射映射$f\colon C\to D$，使得$r = s\circ f$，并且

$$\alpha(C,r) \le \alpha(D,s).$$

注意，若 $(C,r) \le (D,s)$，则

$$w(C,r) = \sum_{x\in C} r(x) = \sum_{x\in C} s(f(x)) = \sum_{y\in D} s(y) = w(D,s).$$

根据公理 (iii)，对于 (C,r)、$(D,s) \in \mathcal{C}^2W$，

$$\begin{aligned}
(C,r)\le(D,s) &\Leftrightarrow \mathcal{C}w(C,r)\le\mathcal{C}w(D,s)\\
&\Leftrightarrow \alpha(\mathcal{C}w(C,r))\le\alpha(\mathcal{C}w(D,s))\\
&\Leftrightarrow \alpha(C, w\circ r)\le\alpha(D, w\circ s). \qquad (1)
\end{aligned}$$

此外，如果 $(C,r) \le (D,s)$ 属于 $\mathcal{C}^2W$，那么

$$w(C,\alpha\circ r) = \sum_{x\in C}\alpha(r(x)) = \sum_{x\in C}\alpha(s(f(x))) = \sum_{y\in D}\alpha(s(y)) = w(D,\alpha\circ s). \qquad (2)$$

引理 4. μ_W： $\mathcal{C}^2W \to \mathcal{C}W$ 和 η_W： $W \to \mathcal{C}W$ 关于 $\leq$ 是单调的。

证明。 对于 η_W，设 $a,b \in W$ 且 $a \leq b$。根据引理 3，

$$\alpha(\eta_W(a)) = 0 = \alpha(\eta_W(b)) \qquad w(\eta_W(a)) = a \leq b = w(\eta_W(b)).$$

For μ_W，设 (C,r)、$(D,s) \in \mathcal{C}^2W$ 且 $(C,r) \leq (D,s)$。根据引理 3，（1）和（2），

$$\begin{aligned}\alpha(\mu_W(C,r)) &= \alpha(C, w \circ r) + w(C, \alpha \circ r)\\ &\leq \alpha(D, w \circ s) + w(D, \alpha \circ s) = \alpha(\mu_W(D,s)).\end{aligned}$$

□

定理 5. *§4* 中的代数 $(\mathbb{R}_+,w)$ 及由 α 定义的序关系 $\leq$ 下的算法等价于霍夫曼算法，并能为给定的权重多重集生成最优霍夫曼编码。

证明。 取 $X \subset \mathbb{R}_+$ 为有限多重集，并按升序对集合 X 排序。对于霍夫曼编码的二元情形（d元版本的证明方法相同），我们总是选择$k = 2$。第一步，令 $a_0,a_1 \in X$ 为列表中最小的两个元素。构造对象 $(\{0,1\},i \mapsto a_i) \in \mathcal{C}X$。当$n = 2$时，这就是列表中唯一剩下的对象。否则，我们将它们合并为一个元素，其权重为 a_0 和 a_1的权重之和，这与霍夫曼编码的做法完全一致。

对于 $n > 2$ 的情况，集合 X 中还剩下一些元素。取所有剩余的 $a \in X\backslash\{a_0,a_1\}$，并将 a 替换为 $\eta_X(a) \in \mathcal{C}X$。这样还剩下 $n-1$ 个类型为 $\mathcal{C}X$ 的元素。如果我们递归调用步骤 1 中的算法，就会不断地将剩余集合中权重最小的两个元素与权重为 w 的元素组合起来。注意，对于加权函数 w，$w(\eta_X(a)) = a$；而对于 $\mathcal{C}X$ 中的元素，w 取 r(x)′s 的和，这与霍夫曼编码完全相同。最终，我们得到一棵属于 $\mathcal{C}^2X$ 的树，其叶节点的权重形式为 $\eta_X(a_i)$。将这棵树记为 (D, s)。取 μ_X(D, S)，即可得到我们所需的 $\mathcal{C}X$ 中的树。 □

6.2 PIFO 树

PIFO 树在 [3] 中被引入，作为可编程分组调度器的模型。在 [1]，的近期工作中，研究者进一步研究了 PIFO 树，并赋予其一种语义，使其支持某些嵌入算法。定义了同态嵌入这一概念，用于确定一个 PIFO 树能否由另一个 PIFO 树表示，并在可以表示时找到相应的嵌入。我们所考虑的嵌入算法将任意 PIFO 树嵌入一棵d 叉树。这是 §4 中算法的一个特例：我们选择加权 (W,w) 中的w，以使源树能够嵌入其中的目标d 叉树的高度最小化。

对于此应用，我们采用加权后的 $W = \mathbb{N}$

$$w(C,r) = \max_{x \in C} |x| + r(x).$$

这为单子$(\mathcal{C},\mu,\eta)$给出了一个 Eilenberg-Moore 代数(W,w) 。对于 $\in W$ 中的$(C,r) \in \mathcal{C}^2 W$，且$r(x) = (C_x, r_x)$，与之前一样，我们有

$$\eta_W(a) = (\{\varepsilon\}, \varepsilon \mapsto a) \qquad \mu_W(C,r) = (\{xy \mid x \in C,\ y \in C_x\}, xy \mapsto r_x(y)),$$

so

$$\begin{aligned}
w(\eta_W(a)) &= w(\{\varepsilon\}, \varepsilon \mapsto a) = \max_{x\in\{\varepsilon\}} |x| + (\varepsilon \mapsto a)(x) = |\varepsilon| + a = a,\\
w(\mu_W(C,r)) &= w(\{xy \mid x \in C,\ y \in C_x\}, xy \mapsto r_x(y)) = \max_{x\in C}\max_{y\in C_x} |xy| + r_x(y)\\
&= \max_{x\in C}\max_{y\in C_x} |x| + |y| + r_x(y) = \max_{x\in C} |x| + \max_{y\in C_x} |y| + r_x(y)\\
&= \max_{x\in C} |x| + w(C_x, r_x) = \max_{x\in C} |x| + w(r(x))\\
&= w(C, w\circ r) = w(\mathcal{C}w(C,r)).
\end{aligned}$$

对于(C,r)、$(D,s) \in \mathcal{C}W$，如果存在双射$f\colon C \to D$，使得$r = s\circ f$，则定义$(C,r) \le (D,s)$，并且

$$w(C,r) \le w(D,s).$$

引理 6. *$\mu_W\colon \mathcal{C}^2W \to \mathcal{C}W$ 和 $\eta_W\colon W \to \mathcal{C}W$ 关于 $\le$是单调的。*

证明。对于 η_W，若 $a \le b$，则 $w(\eta_W(a)) = a \le b = w(\eta_W(b))$。

对于 μ_W，设$(C,\ r)$，$(D,\ s) \in \mathcal{C}^2W$且$(C,\ r) \le (D,\ s)$。根据公理 (iii),

$$\begin{aligned}
(C,r) \le (D,s) &\Leftrightarrow \mathcal{C}w(C,r) \le \mathcal{C}w(D,s)\\
&\Leftrightarrow w(\mathcal{C}w(C,r)) \le w(\mathcal{C}w(D,s)).
\end{aligned}$$

Then

$$w(\mu_W(C,r)) = w(\mathcal{C}w(C,r)) \le w(\mathcal{C}w(D,s)) = w(\mu_W(D,s)).$$ □

定理 7. *§4 中的代数$(\mathbb{N},w)$ 及由 w 定义的序关系 $\le$ 所对应的算法，等价于判定是否存在将 PIFO 树嵌入有界 d 叉树的方法，并在存在时求出该嵌入。*

7 结论

我们提出了一种广义 Huffman 算法，并证明两种已知算法——霍夫曼编码和 PIFO 树嵌入——都可以作为特例推导出来。文献 [1] 中介绍了 PIFO 嵌入算法，并指出它与求最优霍夫曼编码的常用组合算法非常相似，尽管两者基于不同的代数结构。这启发了本文提出的统一推广。

我们提出的广义算法利用了前缀码的单子结构，因此，相较于通常的组合方法，可以对霍夫曼算法进行更具代数性的处理。通过为两个应用分别选择特定的 Eilenberg-Moore 代数，它们便能自然地纳入范畴论框架。其他贪心算法也可能适用于这一框架。

参考文献

[1]Anshuman Mohan、Yunhe Liu、Nate Foster、Tobias Kapp´e 和 Dexter Kozen，《数据包调度的形式化抽象》，技术报告，http://arxiv.org/abs/2211.11659，康奈尔大学，2022年11月。[2] Thomas M. Cover 和 Joy A. Thomas，《信息论基础》，Wiley，第二版，2006年。[3] Anirudh Sivaraman、Suvinay Subramanian、Mohammad Alizadeh、Sharad Chole、Shang-Tse Chuang、Anurag Agrawal、Hari Balakrishnan、Tom Edsall、Sachin Katti 和 Nick McKeown，《线速可编程数据包调度》，发表于 SIGCOMM，2016年。[4] Albert Gran Alcoz、Alexander Dietm¨uller 和 Laurent Vanbever，《SP-PIFO：使用严格优先级队列近似推入式先进先出行为》，发表于 NSDI，2020年。[5] Andrea Asperti 和 Giuseppe Longo，《范畴、类型与结构：面向计算机科学家的范畴论导论》，《计算基础》。麻省理工学院出版社，1991年。[6] Michael Barr 和 Charles Wells，《拓扑斯、三元组与理论》，《数学科学基础丛书》第278卷，Springer，2013年。[7] Michael Barr 和 Charles Wells，《计算机科学范畴论》，Prentice Hall，1990年。[8] Ji˘r´ı Ad´amek、Horst Herrlich 和 George E. Strecker，《抽象范畴与具体范畴》，多佛出版社，2009年。[9] Eugenio Moggi，《计算概念与单子》，《信息与计算》，第93卷，第1期，第55–92页，1991年。

[10] Philip Wadler，“理解单子”，《计算机科学中的数学结构》，第 2 卷，第 461–493 页，1992年。

[11] Philip Wadler，“函数式编程中的单子”，载于《高级函数式编程：高级函数式编程技术国际学校第 *1* 届》，Johan Jeuring 和 Erik Meijer 编，第 925 卷，计算机科学讲义，第 24–52 页。Springer-Verlag，1995年。