Algorithmic counting of nonequivalent compact Huffman codes

UNIVERSITÄT KLAGENFURT

Outline

1. Various interpretations of the same numbers
   - Sum of unit fractions
   - Prefix-Codes
   - Canonical t-ary trees
   - Equivalence

2. Asymptotics

3. Generating Function

4. Counting

Sum of unit fractions

Fix integer base $t \geq 2$.

Definition

$f_t(r)$: number of (unordered) representations of 1 as a sum of r unit fractions whose denominators are powers of t, i.e.,

$$f_t(r) = \#\Bigl\{(x_1, \ldots, x_r) : 1 = \sum_{j=1}^{r} \frac{1}{t^{x_j}}, 0 \leq x_1 \leq x_2 \leq \cdots \leq x_r\Bigr\}.$$

Sum of unit fractions

Fix integer base $t \geq 2$.

Definition

$f_t(r)$: number of (unordered) representations of 1 as a sum of r unit fractions whose denominators are powers of t, i.e.,

$$f_t(r) = \#\Big\{(x_1, \dots, x_r) : 1 = \sum_{j=1}^{r} \frac{1}{t^{x_j}}, 0 \leq x_1 \leq x_2 \leq \cdots \leq x_r\Big\}.$$

E.g. $(t = 2,\ r = 5)$, $f_2(5) = 3$:

$$1 = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16},$$

Sum of unit fractions

Fix integer base $t \geq 2$.

Definition

$f_t(r)$: number of (unordered) representations of 1 as a sum of r unit fractions whose denominators are powers of t, i.e.,

$$f_t(r) = \#\Big\{(x_1, \ldots, x_r) : 1 = \sum_{j=1}^{r} \frac{1}{t^{x_j}}, 0 \leq x_1 \leq x_2 \leq \cdots \leq x_r\Big\}.$$

E.g. $(t = 2,\ r = 5)$, $f_2(5) = 3$:

$$\begin{aligned} 1 &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16}, \\ &= \frac{1}{2} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}, \end{aligned}$$

Sum of unit fractions

Fix integer base $t \geq 2$.

Definition

$f_t(r)$: number of (unordered) representations of 1 as a sum of r unit fractions whose denominators are powers of t, i.e.,

$$f_t(r) = \#\Big\{(x_1, \ldots, x_r) : 1 = \sum_{j=1}^{r} \frac{1}{t^{x_j}}, 0 \leq x_1 \leq x_2 \leq \cdots \leq x_r\Big\}.$$

E.g. $(t = 2,\ r = 5)$, $f_2(5) = 3$:

$$\begin{aligned} 1 &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \frac{1}{16}, \\ &= \frac{1}{2} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}, \\ &= \frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{8} + \frac{1}{8}. \end{aligned}$$

Prefix-Codes

$\mathcal{A}$: finite alphabet; $|\mathcal{A}| =: t$

Definition

prefix code: Finite set $C \subseteq \mathcal{A}^*$ such that:
no word in C is prefix of another word in C.

Prefix-Codes

$\mathcal{A}$: finite alphabet; $|\mathcal{A}| =: t$

Definition

prefix code: Finite set $C \subseteq \mathcal{A}^*$ such that:
no word in C is prefix of another word in C.

E.g., $\mathcal{A} = \{0, 1\}$:

$$C = \{0, 10, 110, 1110, 1111\}.$$

Other names: prefix-free code; Huffman code; instantaneous code

Prefix-Codes

$\mathcal{A}$: finite alphabet; $|\mathcal{A}| =: t$

Definition

prefix code: Finite set $C \subseteq \mathcal{A}^*$ such that:
no word in C is prefix of another word in C.

E.g., $\mathcal{A} = \{0, 1\}$:

$$C = \{0, 10, 110, 1110, 1111\}.$$

Lemma (Kraft-McMillan inequality)

$$\sum_{c \in C} \frac{1}{t^{\operatorname{length}(c)}} \leq 1.$$

Other names: prefix-free code; Huffman code; instantaneous code

Compact Prefix Codes

Definition

"compact" prefix code: if equality holds in the Kraft-McMillan inequality:

$$\sum_{c\in C} \frac{1}{t^{\mathsf{length}(c)}} = 1.$$

Compact Prefix Codes

Definition

"compact" prefix code: if equality holds in the Kraft-McMillan inequality:

$$\sum_{c\in C} \frac{1}{t^{\mathsf{length}(c)}} = 1.$$

E.g., $C = \{0, 10, 110, 1110, 1111\}$ is compact, as

$$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} = 1.$$

Compact Prefix Codes

Definition

"compact" prefix code: if equality holds in the Kraft-McMillan inequality:

$$\sum_{c \in C} \frac{1}{t^{\text{length}(c)}} = 1.$$

E.g., $C = \{0, 10, 110, 1110, 1111\}$ is compact, as

$$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} = 1.$$

However: $C = \{0, 10, 1110\}$ not compact.

Compact Prefix Codes

Definition

"compact" prefix code: if equality holds in the Kraft-McMillan inequality:

$$\sum_{c\in C} \frac{1}{t^{\mathrm{length}(c)}} = 1.$$

E.g., $C = \{0, 10, 110, 1110, 1111\}$ is compact, as

$$\frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \frac{1}{2^4} = 1.$$

However: $C = \{0, 10, 1110\}$ not compact.

Observation

Code is compact $\iff$
If p is a proper prefix of some codeword in C, then pa is a prefix of a codeword in C for all $a \in \mathcal{A}$.

Canonical Prefix-Codes

Definition

Canonical code: Lexicographic ordering corresponds to nondecreasing ordering of the word lengths.

Canonical Prefix-Codes

Definition

Canonical code: Lexicographic ordering corresponds to nondecreasing ordering of the word lengths.

E.g., ($\mathcal{A} = \{0, 1\}$):
$C = \{0, 10, 110, 1110, 1111\}$: canonical,

Canonical Prefix-Codes

Definition

Canonical code: Lexicographic ordering corresponds to nondecreasing ordering of the word lengths.

E.g., ($\mathcal{A} = \{0, 1\}$):
$C = \{0, 10, 110, 1110, 1111\}$: canonical,
$C' = \{1, 01, 001, 0001, 0000\}$: not a canonical code (lexicographical ordering: 0000, 0001, 001, 01, 1).

Canonical Prefix-Codes

Definition

Canonical code: Lexicographic ordering corresponds to nondecreasing ordering of the word lengths.

E.g., ($\mathcal{A} = \{0, 1\}$):
$C = \{0, 10, 110, 1110, 1111\}$: canonical,
$C' = \{1, 01, 001, 0001, 0000\}$: not a canonical code (lexicographical ordering: 0000, 0001, 001, 01, 1).

Definition

$f_t(r)$: number of canonical compact codes of cardinality r over an alphabet of cardinality t.

Canonical Prefix-Codes

Definition

Canonical code: Lexicographic ordering corresponds to nondecreasing ordering of the word lengths.

E.g., ($\mathcal{A} = \{0, 1\}$):
$C = \{0, 10, 110, 1110, 1111\}$: canonical,
$C' = \{1, 01, 001, 0001, 0000\}$: not a canonical code (lexicographical ordering: 0000, 0001, 001, 01, 1).

Definition

$f_t(r)$: number of canonical compact codes of cardinality r over an alphabet of cardinality t.

$f_2(5) = 3$:

$\{0, 10, 110, 1110, 1111\}, \{0, 100, 101, 110, 111\}, \{00, 01, 10, 110, 111\}$.

Canonical Trees

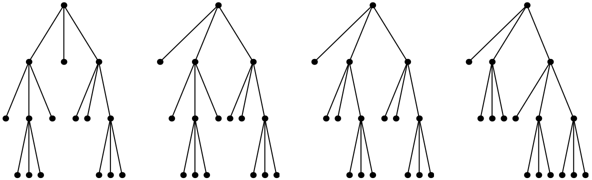
Definition

Canonical t-ary tree: longer paths as far to the right as possible:

Canonical Trees

Definition

Canonical t-ary tree: longer paths as far to the right as possible:



no no no yes

Number of canonical trees

Definition

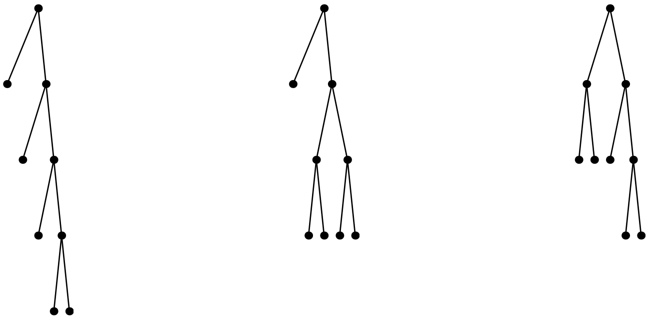
$f_t(r)$: number of canonical t-ary trees with r leaves.

Number of canonical trees

Definition

$f_t(r)$: number of canonical t-ary trees with r leaves.

$f_2(5) = 3$:

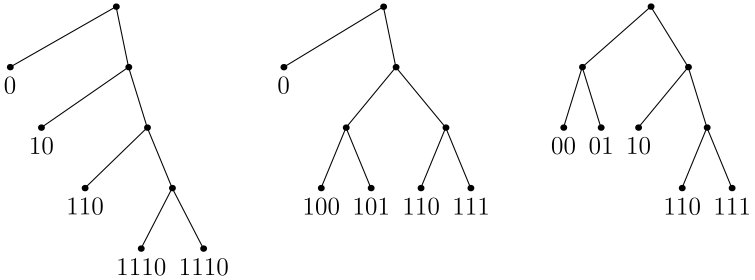


All three definitions of $f_t(r)$ are equivalent

Observation

All three definitions of $f_t(r)$ are equivalent.

$f_2(5) = 3$:



$$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{16}=\frac{1}{2}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}+\frac{1}{8}=\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{8}+\frac{1}{8}=1.$$

Not all r are relevant

n: number of internal vertices (non-leaves).
r: number of leaves
t-ary tree.
Then:

$$r = 1 + n(t-1).$$

Not all r are relevant

n: number of internal vertices (non-leaves).
r: number of leaves
t-ary tree.
Then:

$$r = 1 + n(t-1).$$

$$g_t(n) := f_t(1 + n(t-1)).$$

Asymptotics?

AMERICAN MATHEMATICAL SOCIETY
MathSciNet® *Mathematical Reviews on the Web*

Previous | Up | Next Article

Citations
From References: 0 From Reviews: 0

MR0239986 (39 #1340) 05.10 (10.00)

Bende, Sándor

The Diophantine equation $\sum_{i=1}^{n} 1/2^{x_i} = 1$ and its connection with graph theory and information theory. (Hungarian)

Mat. Lapok **18** 1967 323–327

Denote by $\tau(n)$ the number of solutions of $1 = \sum_{i=1}^{n} 1/2^{x_i}$, $x_1 \leq \cdots \leq x_n$, where the x_i are positive integers. The author gives a graph theoretic interpretation of $\tau(n)$ and obtains a recursion formula for it; further, he discusses some connection with information theory. It would be desirable to obtain an asymptotic formula for $\tau(n)$.

Reviewed by *P. Erdös*

Flajolet/Prodinger 1987

Discrete Mathematics 65 (1987) 149–156
North-Holland 149

LEVEL NUMBER SEQUENCES FOR TREES

Philippe FLAJOLET
INRIA, Rocquencourt, 78153-Le Chesnay, France

Helmut PRODINGER
Technische Universität Wien, A-1040 Wien, Austria

We give explicit asymptotic expressions for the number of "level number sequences" (l.n.s.) associated to binary trees. The level number sequences describe the number of nodes present at each level of a tree.

Asymptotic Expression — Result

Boyd (1975), Komlos, Moser and Nemetz (1984), Flajolet and Prodinger (1987), and others; ...

Theorem (Elsholtz, H. and Prodinger 2011)

For $t \geq 16$:

$$f_t(1+(t-1)n) = g_t(n) = R\rho^{n+1} + R_2\rho_2^{n+1} + 5t^4 r_3^n \varepsilon_1(t,n),$$

where (with $D = \log 2$)

$$\rho = 2 - \frac{1}{2^{t+1}} - \frac{t+3}{2^{2t+3}} - \frac{3t^2+19t+24}{2^{3t+6}} + \frac{0.28t^3}{2^{4t}}\varepsilon_2(t),$$

$$\rho_2 = 1 + \frac{D}{t} - \frac{D-D^2}{2t^2} + \frac{4D^3+3D^2+6\log 2}{24t^3} + \frac{0.04}{t^4}\varepsilon_3(t)$$

$$r_3 = 1 + \frac{D}{t} - \frac{D-D^2}{2t^2}, \qquad R = \frac{1}{8} + \frac{t-2}{2^{t+5}} + \frac{0.001t^3}{2^{2t}}\varepsilon_4(t),$$

$$R_2 = \frac{1}{4t} - \frac{4D+1}{8t^2} + \frac{0.77}{t^3}\varepsilon_5(t) \qquad \forall j\, \forall t : |\varepsilon_j(t)| \leq 1.$$

Values of the Parameters for $t \leq 10$

t	ρ	ρ_2	r_3	R	R_2
2	1.7941471875	1.2795491347	1.13	0.141	0.061
3	1.9207125384$*$	1.2114793781	1.1	0.133$*$	0.050
4	1.9646247578$*$	1.1651583745	1.1	0.130$*$	0.042
5	1.9832939867$*$	1.1344596984	1.13437555$*$	0.128$*$	0.036
6	1.9918971757$*$	1.1130198498	1.11257044$*$	0.127$*$	0.031
7	1.9960151077$*$	1.0973240755	1.09685068$*$	0.126$*$	0.028
8	1.9980255446$*$	1.0853892421	1.08498172$*$	0.125$*$	0.025
9	1.9990176639$*$	1.0760324885	1.07570343$*$	0.125$*$	0.022
10	1.9995101615$*$	1.0685114109	1.06825125$*$	0.125$*$	0.021

Starred ($*$) entries correspond to values satisfying the asymptotic estimates of the Theorem.

Generating Function

$\mathcal{T}$: set of t-ary rooted canonical trees.
For $T \in \mathcal{T}$: $n(T) :=$ number of inner vertices of T.

$$F(q) := \sum_{n \geq 0} g_t(n) q^n = \sum_{T \in \mathcal{T}} q^{n(T)}.$$

Generating Function

$\mathcal{T}$: set of t-ary rooted canonical trees.
For $T \in \mathcal{T}$: $n(T) :=$ number of inner vertices of T.

$$F(q) := \sum_{n\geq 0} g_t(n) q^n = \sum_{T\in\mathcal{T}} q^{n(T)}.$$

Theorem

Setting $[k] := 1 + t + t^2 + \cdots + t^{k-1}$, *we have*

$$F(q) = \frac{\displaystyle\sum_{j=0}^{\infty} (-1)^j q^{[j]} \prod_{i=1}^{j} \frac{q^{[i]}}{1-q^{[i]}}}{\displaystyle\sum_{j=0}^{\infty} (-1)^j \prod_{i=1}^{j} \frac{q^{[i]}}{1-q^{[i]}}}.$$

Auxiliary Generating Function

For $T \in \mathcal{T}$: $m(T) :=$ number of leaves of maximum height.

$$G(q,u) := \sum_{T \in \mathcal{T}} q^{n(T)} u^{m(T)}.$$

By definition: $F(q) = G(q,1)$.

Auxiliary Generating Function

For $T \in \mathcal{T}$: $m(T) :=$ number of leaves of maximum height.

$$G(q,u) := \sum_{T \in \mathcal{T}} q^{n(T)} u^{m(T)}.$$

By definition: $F(q) = G(q,1)$.

Partition $\mathcal{T}$ w.r.t. height:

$$G_k(q,u) = \sum_{\substack{T \in \mathcal{T} \\ \text{height}(T)=k}} q^{n(T)} u^{m(T)}.$$

Obviously:

$$G(q,u) = \sum_{k \geq 0} G_k(q,u).$$

Functional Equation for $G_k(q, u)$

Observation

$T \in \mathcal{T}$ of height k: $\longleftrightarrow$ *$m(T)$ trees $T'_1, \ldots, T'_{m(T)}$ of height $k+1$:*
T'_j: Replace replacing j of the $m(T)$ leaves of maximum height by vertices with t attached leaves.

Functional Equation for $G_k(q, u)$

Observation

$T \in \mathcal{T}$ of height k: $\longleftrightarrow$ $m(T)$ *trees* $T'_1, \ldots, T'_{m(T)}$ *of height* $k+1$*:*
T'_j*: Replace replacing* j *of the* $m(T)$ *leaves of maximum height by vertices with* t *attached leaves.*

Thus

$$G_{k+1}(q, u)$$

Functional Equation for $G_k(q, u)$

Observation

$T \in \mathcal{T}$ of height k: $\longleftrightarrow$ $m(T)$ trees $T'_1, \ldots, T'_{m(T)}$ of height $k+1$:
T'_j: Replace replacing j of the $m(T)$ leaves of maximum height by vertices with t attached leaves.

Thus

$$G_{k+1}(q, u) = \sum_{\substack{T \in \mathcal{T} \\ \text{height}(T)=k}} \sum_{j=1}^{m(T)} q^{n(T)+j} u^{jt}$$

Functional Equation for $G_k(q, u)$

Observation

$T \in \mathcal{T}$ of height k: $\longleftrightarrow$ $m(T)$ trees $T'_1, \ldots, T'_{m(T)}$ of height $k+1$:
T'_j: Replace replacing j of the $m(T)$ leaves of maximum height by vertices with t attached leaves.

Thus

$$G_{k+1}(q, u) = \sum_{\substack{T \in \mathcal{T} \\ \text{height}(T)=k}} \sum_{j=1}^{m(T)} q^{n(T)+j} u^{jt}$$

$$= \sum_{\substack{T \in \mathcal{T} \\ \text{height}(T)=k}} q^{n(T)} \cdot qu^t \cdot \frac{1 - (qu^t)^{m(T)}}{1 - qu^t}$$

Functional Equation for $G_k(q, u)$

Observation

$T \in \mathcal{T}$ of height k: $\longleftrightarrow$ $m(T)$ trees $T'_1, \ldots, T'_{m(T)}$ of height $k+1$:
T'_j: Replace replacing j of the $m(T)$ leaves of maximum height by vertices with t attached leaves.

Thus

$$\begin{aligned} G_{k+1}(q,u) &= \sum_{\substack{T\in\mathcal{T}\\ \text{height}(T)=k}} \sum_{j=1}^{m(T)} q^{n(T)+j} u^{jt} \\ &= \sum_{\substack{T\in\mathcal{T}\\ \text{height}(T)=k}} q^{n(T)} \cdot qu^t \cdot \frac{1-(qu^t)^{m(T)}}{1-qu^t} \\ &= \frac{qu^t}{1-qu^t}\left(G_k(q,1) - G_k(q,qu^t)\right). \end{aligned}$$

Functional Equation for $G(q, u)$

Recall:

$$G_{k+1}(q,u) = \frac{qu^t}{1-qu^t}\left(G_k(q,1) - G_k(q,qu^t)\right);$$

Functional Equation for $G(q, u)$

Recall:

$$G_{k+1}(q, u) = \frac{qu^t}{1 - qu^t} \left(G_k(q, 1) - G_k(q, qu^t) \right);$$

Summation over k:

$$G(q, u) = u + \frac{qu^t}{1 - qu^t} \left(G(q, 1) - G(q, qu^t) \right).$$

Functional Equation for $G(q, u)$

Recall:

$$G_{k+1}(q, u) = \frac{qu^t}{1 - qu^t} \left(G_k(q, 1) - G_k(q, qu^t) \right);$$

Summation over k:

$$G(q, u) = u + \frac{qu^t}{1 - qu^t} \left(G(q, 1) - G(q, qu^t) \right).$$

Iteration

Computing $g_t(N)$ in 1972

Theorem (Even and Lempel 1972)

$g_2(N)$ can be computed with $O(N^3)$ additions of integers of bit size $O(n)$, i.e., $O(n^4)$ bit operations.

Computing $g_t(N)$ in 2019

Theorem (Elsholtz, Krenn and H. 2019)

Calculating the first N terms of $g_t(n)$ can be done with

$$\mathbf{D}+\bigl(\log_t N+O(1)\bigr)\mathbf{M}+2\bigl(\log_t N+O(1)\bigr)\mathbf{A}+O(\log N)\mathbf{S}+O(\log N)\mathbf{O}$$

power series operations, ...

Computing $g_t(N)$ in 2019

Theorem (Elsholtz, Krenn and H. 2019)

Calculating the first N terms of $g_t(n)$ can be done with

$$\mathbf{D} + \bigl(\log_t N + O(1)\bigr)\mathbf{M} + 2\bigl(\log_t N + O(1)\bigr)\mathbf{A} + O(\log N)\mathbf{S} + O(\log N)\mathbf{O}$$

power series operations, . . .

- **D**: division,
- **M**: multiplication,
- **A**: addition,
- **S**: series manipulation (e.g. initialization),
- **O**: other operations (e.g. index manipulation)

of power series of precision N.

Computing $g_t(N)$ in 2019 (contd.)

Theorem (Elsholtz, Krenn and H. 2019, contd.)

Calculating the first N terms of $g_t(n)$ can be done with ...

$$N(\log N)^2\, 2^{O(\log^* N)}$$

operations in the ring of integers,

$\log^*(N)$: iterated logarithm (number of iterations of log to get to value ≤ 1)

Computing $g_t(N)$ in 2019 (contd.)

Theorem (Elsholtz, Krenn and H. 2019, contd.)

Calculating the first N terms of $g_t(n)$ can be done with ...

$$N(\log N)^2 \, 2^{O(\log^* N)}$$

operations in the ring of integers, and with

$$N^2(\log N)^4 \, 2^{O(\log^* N)}$$

bit operations.

$\log^*(N)$: iterated logarithm (number of iterations of log to get to value ≤ 1)

Computing $g_t(N)$ in 2019 (contd.)

Theorem (Elsholtz, Krenn and H. 2019, contd.)

Calculating the first N terms of $g_t(n)$ can be done with ...

$$N(\log N)^2\, 2^{O(\log^* N)}$$

operations in the ring of integers, and with

$$N^2(\log N)^4\, 2^{O(\log^* N)}$$

bit operations.

$\log^*(N)$: iterated logarithm (number of iterations of log to get to value ≤ 1)
According to preprint by Harvey and Van Der Hoeven on multiplication: remove factor $2^{O(\log^* N)}$.