any adjacent 1's (this property is used by certain data compression methods; see Section 8.5.4). As an example, the integer 33 equals the sum $1+3+8+21$, so it is expressed in the Fibonacci base as the 7-bit number 1010101.

◇ **Exercise 2.10:** Show how the Fibonacci numbers can be used to construct a prefix code.

A nonnegative integer can be represented as a finite sum of binomial coefficients

$$n=\binom{a}{1}+\binom{b}{2}+\binom{c}{3}+\binom{d}{4}+\cdots, \quad \text{where } 0\le a<b<c<d\cdots$$

are integers and $\binom{i}{n}$ is the binomial coefficient $\frac{i!}{n!(i-n)!}$. This is the *binomial number system*.

*Actually, there is. Two is the smallest integer that can be a base for a number system. Ten is the number of our fingers.

2.4 The Golomb Code

The *Golomb code* for positive integers n [Golomb 66] can be an effective Huffman code. The code depends on the choice of a parameter b. The first step is to compute the three quantities

$$q=\left\lfloor\frac{n-1}{b}\right\rfloor, \quad r=n-qb-1, \text{ and } c=\lfloor\log_2 b\rfloor,$$

(where the notation $\lfloor x\rfloor$ implies truncation of x) following which the code is constructed in two parts; the first is the value of $q+1$, coded in unary (Exercise 2.5), and the second, the binary value of r. The latter is coded in a special way. The first c values of r are coded in c bits each (with a most-significant bit of 0) and the rest are coded in $c+1$ bits each (with a most-significant bit of 1).

Choosing $b=3$, for example, produces $c=1$ and the three remainders, 0, 1, and 2. The first one is coded in one bit 0, and the remaining two are coded in two bits each with a leading 1, thus 10, and 11. Selecting $b=5$ results in $c=2$ and produces the five remainders 0 through 4. The first two are coded in two bits each with a leading zero, and the rest are coded in three bits with a leading 1. Thus, 00, 01, 100, 101, and 110. Table 2.8 shows some examples of the Golomb code for $b=3$ and $b=5$.

n	1	2	3	4	5	6	7	8	9	10	11
$b=3$	0|0	0|10	0|11	10|0	10|10	10|11	110|0	110|10	110|11	1110|0	1110|10
$b=5$	0|00	0|01	0|100	0|101	0|110	10|00	10|01	10|100	10|101	10|110	110|00

Table 2.8: Some Golomb Codes for $b=3$ and $b=5$.

Imagine an input data stream that consists of positive integers where the probability of integer n appearing in the data is $P(n) = (1-p)^{n-1}p$, for some $0 \le p \le 1$. It can be shown that the Golomb code is an optimal code for this data if b is chosen such that

$$(1-p)^b + (1-p)^{b+1} \le 1 < (1-p)^{b-1} + (1-p)^b.$$

Given the right data, it is easy to generate the best variable-size codes without going through the Huffman algorithm.

Section 4.7.1 illustrates the use of the Golomb code for lossless image compression.

Bibliography

Golomb, S. W. (1966) "Run-Length Encodings," *IEEE Transactions on Information Theory* IT-12(3):399–401.

In addition to the codes, Solomon W. Golomb has his "own" Golomb constant 0.624329988543550870992936383100837244179642620180529286

2.5 The Kraft-MacMillan Inequality

This inequality is related to unambiguous variable-size codes. Its first part states that given an unambiguous variable-size code, with n codes of sizes L_i, then

$$\sum_{i=1}^{n} 2^{-L_i} \le 1. \tag{2.1}$$

The second part states the opposite, namely, given a set of n positive integers $(L_1, L_2, \ldots, L_n)$ that satisfy equation (2.1), there exists an unambiguous variable-size code such that the L_i are the sizes of its individual codes. Together, both parts say that a code is unambiguous if and only if it satisfies relation (2.1).

This inequality can be related to the entropy by observing that the lengths L_i can always be written as $L_i = -\log_2 P_i + E_i$, where E_i is simply the amount by which L_i is greater than the entropy (the extra length of code i).

This implies that

$$2^{-L_i} = 2^{(\log_2 P_i - E_i)} = 2^{\log_2 P_i}/2^{E_i} = P_i/2^{E_i}.$$

In the special case where all the extra lengths are the same ($E_i = E$), the Kraft inequality says that

$$1 \ge \sum_{i=1}^{n} P_i/2^E = \left(\sum_{i=1}^{n} P_i\right) \Big/ 2^E = 1/2^E \Longrightarrow 2^E \ge 1 \Longrightarrow E \ge 0.$$

An unambiguous code has non-negative extra length, meaning its length is greater than or equal to the length determined by its entropy.