Efficient Generation of Binary Words of Given Weight

Nicolas Sendrier
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay CEDEX, FRANCE

1 Introduction

In some cryptographic systems [2, 3], it is necessary to put some information in binary words of given weight, say $t$, and given length, say $n$.

It is possible to produce an explicit bijection between the set $W_{n,t}$ of these words and the set of integer $\{1, 2, \ldots, \binom{n}{t}\}$. As far as we know, this cannot be achieved efficiently.

We propose here a procedure that converts any binary sequence into elements of $W_{n,t}$ in linear time both for encoding and decoding. The solution we propose is an approximation: all the words of $W_{n,t}$ will not be reached and the words of $W_{n,t}$ are not obtained with uniform probability.

2 Representing Words of Given Weight

Let $x$ be an element of $W_{n,t}$, and let $k_0 < \ldots < k_{t-1}$ denote the positions of the “1”s, the positions are numbered from 1 to $n$. Let $l_0 = k_0$, and for all $i$, $0 < i < t$, let $l_i = k_i - k_{i-1}$. The word $x$ will be represented by the $t$-tuple of integers $(l_0, \ldots, l_{t-1})$.

Reciprocally, any $t$-tuple of strictly positive integers $(l_0, \ldots, l_{t-1})$ such that $\sum_{i=0}^{t-1} l_i \leq n$ will represent an element of $W_{n,t}$.

Our goal is to encode binary information into such $t$-tuples.

3 Huffman Code

We consider the memoryless source $X = \{1, \ldots, K\}$ with the law probability $P_X(i) = \lambda \binom{n-i}{t-1}/\binom{n}{t}$, where $\lambda$ is such that $\sum_{i=1}^{K} P_X(i) = 1$.

When $K = n - t + 1$, then $\lambda = 1$, and the probability $P_X(i)$ is the probability to have a sequence of $i - 1$ “0” between two “1” when we consider the set $W_{n,t}$ with a uniform distribution. We will see that taking values of $K$ smaller than $n - k + 1$ will reduce the amount the memory for the encoder and the decoder, but will not significantly reduce the performance.

Let $h_K$ be a Huffman code of source $X$ [1, Ch. 3]. We have

$$h_K : \{1, \ldots, K\} \rightarrow \{0, 1\}^* = \cup_{i>0}\{0, 1\}^i$$

$$l_i \mapsto h_K(l_i)$$
To $h_K$ we can associate the encoder of infinite sequences:

$$H_K : \{1, \ldots, K\}^\mathbb{N} \rightarrow \{0, 1\}^\mathbb{N}$$

$$(l_i)_{i \geq 0} \mapsto (h_K(l_i))_{i \geq 0}$$

Since $h_K$ is a Huffman code, $H_K$ is a bijection. By extension we will also use $H_K$ to denote the image of finite sequences.

4 Encoding a Binary Sequence

Let $a = (a_i)_{i \geq 0}$ be a binary sequence to be encoded, and let

$$l = (l_i)_{i \geq 0} = H_K^{-1}(a).$$

The encoding of the $t$ first letters of $l$ will give

$$H_K(l_0, \ldots, l_{t-1}) = (a_0, \ldots, a_{m-1})$$

for some $m$. If $\sum_{i=0}^{t-1} l_i \leq n$, then the $t$-tuple $(l_0, \ldots, l_{t-1})$ represents a word of $W_{n,t}$ “containing” $m$ bits of information.

The probability that $\sum_{i=0}^{t-1} l_i \leq n$ can be computed by using generating function techniques. We have

$$F(Y, Z) = \left( \sum_{i=1}^{K} Y^i Z^{r_i} \right)^t = \sum_{u,v} A_{u,v} Y^u Z^v$$

where $r_i$ is the length of $h_K(i)$, and $A_{u,v}$ is the number of sequences of $X^t$ which adds to $u$, and whose image by $H_K$ has a binary length of $v$.

The probability of the event “$u \leq n$”, when the sequence $a$ is produced by a uniform-ally distributed binary source, is equal to

$$P_{n,t}(K) = \sum_{u \leq n} \frac{A_{u,v}}{2^v} = \sum_{u \leq n} [Y^u] F(Y, \frac{1}{2})$$

and the average value of $v$ given that $u \leq n$ is equal to

$$\mathcal{L}_{n,t}(K) = \frac{1}{P_{n,t}(K)} \sum_{u \leq n} \frac{v A_{u,v}}{2^v} = \frac{1}{P_{n,t}(K)} \sum_{u \leq n} [Y^u] \frac{1}{2} \frac{\partial F}{\partial Z}(Y, \frac{1}{2})$$

This number $\mathcal{L}_{n,t}(K)$ is the average number of information bits contained in one word of $W_{n,t}$ obtained that way.

For $n = 1024$, $t = 50$ and $K = 77$, we have $P_{n,t}(K) = 0.747$ and $\mathcal{L}_{n,t}(K) = 276.0$ while we have $\log_2 \binom{n}{t} = 284.0$. Such a result is practically not acceptable since on average 25% of the binary sequences cannot be encoded.