Discrete Mathematics 340 (2017) 51–57

Contents lists available at ScienceDirect

Discrete Mathematics

journal homepage: www.elsevier.com/locate/disc

On the set of uniquely decodable codes with a given sequence of code word lengths

Adam Woryna

Silesian University of Technology, Institute of Mathematics, ul. Kaszubska 23, 44-100 Gliwice, Poland

ARTICLE INFO

Article history:
Received 19 January 2016
Received in revised form 14 July 2016
Accepted 9 August 2016

Keywords:
Uniquely decodable code
Prefix code
Code with finite delay
Kraft's procedure
Sardinas–Patterson algorithm

ABSTRACT

For every natural number $n \geq 2$ and every finite sequence L of natural numbers, we consider the set $UD_n(L)$ of all uniquely decodable codes over an n-letter alphabet with the sequence L as the sequence of code word lengths, as well as its subsets $PR_n(L)$ and $FD_n(L)$ consisting of, respectively, the prefix codes and the codes with finite delay. We derive the estimation for the quotient $|UD_n(L)|/|PR_n(L)|$, which allows to characterize those sequences L for which the equality $PR_n(L) = UD_n(L)$ holds. We also characterize those sequences L for which the equality $FD_n(L) = UD_n(L)$ holds.

1. Preliminaries and the statement of the results

Let X be an alphabet with $n := |X| \geq 2$ letters. We refer to a finite sequence

$$C = (v_1, \dots, v_m), \quad m \geq 1$$

of words over X as *a code* and to the words $v_i \in X^*$ $(1 \leq i \leq m)$ as the *code words*. In particular, our convention differs a bit from the more usual one, where codes are considered as sets of words rather than sequences of words. The code C is called *uniquely decodable* if for all $l, l' \geq 1$ the equality $v_{i_1} v_{i_2} \dots v_{i_l} = v_{j_1} v_{j_2} \dots v_{j_{l'}}$ with $1 \leq i_t, j_{t'} \leq m$ $(1 \leq t \leq l, 1 \leq t' \leq l')$ implies $l = l'$ and $i_t = j_t$ for every $1 \leq t \leq l$. Thus every uniquely decodable code must be an injective sequence of non-empty words. In the algebraic language, one could say that the code C is uniquely decodable if and only if the monoid generated by the set $\{v_1, \dots, v_m\}$ (with concatenation of words as the monoid operation) is a free monoid of rank m freely generated by this set, or that this set is an m-element basis for this monoid. If for all $1 \leq i, j \leq m$ the condition: *v_i is a prefix (initial segment) of v_j* implies $i = j$, then C is called a *prefix code*.

The prefix codes are the most useful examples of uniquely decodable codes and, in a sense, they are universal for all uniquely decodable codes. Namely, according to the Kraft–McMillan theorem [5], for every finite sequence $L = (a_1, \dots, a_m)$ of natural numbers the following three statements are equivalent: (1) there exists a uniquely decodable code $C = (v_1, \dots, v_m)$ with the sequence L as the sequence of code word lengths, i.e. $|v_i| = a_i$ for every $1 \leq i \leq m$; (2) there exists a prefix code $C' = (v'_1, \dots, v'_m)$ with the sequence L as the sequence of code word lengths; (3) the inequality $\sum_{i=1}^{m} n^{-a_i} \leq 1$ holds.

Uniquely decodable codes of length $m \leq 2$ are exceptional, as every such a code has finite delay [2]. Recall that a code C has finite delay if there is a number t with the following property: picking up the consecutive letters of an arbitrary word $u \in X^*$ which can be factorized into the code words, it is enough to pick up at most t first letters of u to be sure which code word begins u (see also [1]). The smallest number t with this property is called the *delay* of the code C. If such a number

E-mail address: adam.woryna@polsl.pl.

http://dx.doi.org/10.1016/j.disc.2016.08.013
0012-365X/

does not exist, then we say that the code has *infinite delay*. Obviously, every prefix code has finite delay (which is not greater that the maximum length of a code word) and every code with finite delay must be uniquely decodable. It turns out (see Section 6.1.2 in [4] and Proposition 6.1.9 therein) that a code $C = (v_1, \ldots, v_m)$ has infinite delay if and only if there is an infinite word $u \in X^\omega$ and two factorizations

$$u = v_{i_1} v_{i_2} v_{i_3} \ldots,$$
$$u = v_{j_1} v_{j_2} v_{j_3} \ldots$$

into code words such that $v_{i_1} \neq v_{j_1}$. If $m \geq 3$, then there are uniquely decodable codes of length m which have infinite delay.

Example 1. The code $C = (10, 100, 000)$ has infinite delay because of the following two factorizations of the infinite word $u = 10^\infty$ into the code words:

$$10 - 000 - 000 - 000 - \cdots,$$
$$100 - 000 - 000 - 000 - \cdots.$$

The code C is also uniquely decodable, as its *reverse* $C^R = (01, 001, 000)$ is a prefix code (we use the well known fact that a code is uniquely decodable if and only if its reverse is uniquely decodable).

For every finite sequence L of natural numbers we denote by $UD_n(L)$ the set of all uniquely decodable codes over the alphabet X with the sequence L as the sequence of code word lengths. We also consider the subset $PR_n(L) \subseteq UD_n(L)$ of all prefix codes and the subset $FD_n(L) \subseteq UD_n(L)$ of all codes with finite delay. Thus, we have the inclusions $PR_n(L) \subseteq FD_n(L) \subseteq UD_n(L)$ and the set $UD_n(L)$ is non-empty if and only if the set $PR_n(L)$ is non-empty. If L is constant, then each code in $UD_n(L)$ is a block code and we obviously have in this case: $PR_n(L) = UD_n(L)$. As we mentioned above, if the length of L is 1 or 2, then $FD_n(L) = UD_n(L)$.

The aim of this work is to characterize those sequences L for which the equality $PR_n(L) = UD_n(L)$ holds, as well as those sequences L for which $FD_n(L) = UD_n(L)$. For the first characterization, we modify the Kraft's procedure [3] describing the construction of an arbitrary prefix code $C \in PR_n(L)$. This allows us to obtain the following estimation for the quotient $|UD_n(L)|/|PR_n(L)|$ in the case when L is non-constant.

Theorem 1. *Let L be a non-constant sequence such that the set $UD_n(L)$ is non-empty. Then we have*

$$\frac{|UD_n(L)|}{|PR_n(L)|} \geq 1 + \frac{r_a r_b}{|PR_n((a, b))|},$$

where a and b are arbitrary two different values of L and r_a (resp. r_b) is the number of those elements in L which are equal to a (resp. to b).

As a direct consequence of the above inequality, we obtain the following result.

Theorem 2. *If the set $UD_n(L)$ is non-empty, then the statements are equivalent:*

(i) *$UD_n(L) = PR_n(L)$,*
(ii) *L is constant.*

For the second characterization, we involve the Sardinas–Patterson algorithm [6] and obtain the following theorem.

Theorem 3. *If the set $UD_n(L)$ is non-empty, then the statements are equivalent:*

(i) *$FD_n(L) = UD_n(L)$,*
(ii) *the length of L is not greater than 2 or, after reordering the elements of L, we have $L = (a, a, \ldots, a, b)$, where $a \mid b$.*

2. The Kraft's procedure for prefix codes

Let L be a finite sequence of natural numbers. We now present the Kraft's method for the construction of an arbitrary code $C \in PR_n(L)$ [3], which can be used in deriving the formula for the number of elements in the set $PR_n(L)$.

Let $\widetilde{L} := \{\nu_1, \nu_2, \ldots, \nu_l\}$ be the set of values of the sequence L ordered from the smallest to the largest, i.e. $\nu_1 < \nu_2 < \cdots < \nu_l$ and let r_{ν_i} $(1 \leq i \leq l)$ be the number of those elements in L which are equal to ν_i.

To construct an arbitrary code $C \in PR_n(L)$ we proceed as follows. As the code words of length ν_1, we choose arbitrarily r_{ν_1} words among all the words of length ν_1. This can be done in $\binom{n^{\nu_1}}{r_{\nu_1}}$ ways. Next, we must arrange the chosen words in r_{ν_1} available positions of the sequence C, which can be done in $r_{\nu_1}!$ ways. For the construction of the code words of length $\nu_2 > \nu_1$, we can use the remaining $n^{\nu_1} - r_{\nu_1}$ available words of length ν_1 as possible prefixes; for the final segments, we can take arbitrary words of length $\nu_2 - \nu_1$. Consequently, the number of ways to construct the code words of length ν_2 is equal to

$$\binom{n^{\nu_2 - \nu_1} \cdot (n^{\nu_1} - r_{\nu_1})}{r_{\nu_2}}.$$

Finally, as before, we arrange the chosen words in the sequence C, which can be done in $r_{\nu_2}!$ ways.