Some properties of Fix – Free Codes

R.Ahlswede
Fakultät für Mathematik,
Universität Bielefeld,
Postfach 100131,
33501 Bielefeld,
Germany

B.Balkenhol*
Fakultät für Mathematik,
Universität Bielefeld,
Postfach 100131,
33501 Bielefeld,
Germany

L.Khachatrian†
Fakultät für Mathematik,
Universität Bielefeld,
Postfach 100131,
33501 Bielefeld,
Germany

Abstract

A (variable length) code is fix – free code if no codeword is a prefix or a suffix of any other. A database constructed by a fix – free code is instantaneously decodeable from both sides. We discuss the existence of fix – free codes, relations to the deBrujin Network and shadow problems. Particulary we draw attention to a remarkable **conjecture**: For numbers $l_1, \ldots, l_N$ satisfying $\sum_{i=1}^{N} 2^{-l_i} \leq \frac{3}{4}$ a fix–free code with lengths $l_1, \ldots l_N$ exists. If true, this bound is best possible.

*email:bernhard@mathematik.uni-bielefeld.de
†email:lk@mathematik.uni-bielefeld.de

1 Basic Definitions

For a finite set $\mathcal{X} = \{0, \dots, a-1\}$, called alphabet, we form $\mathcal{X}^n = \prod_1^n \mathcal{X}$, the words of length n, with letters from $\mathcal{X}$ and $\mathcal{X}^* = \bigcup_{n=0}^{\infty} \mathcal{X}^n$, the set of all finite length words including the empty word e from $\mathcal{X}^0 = \{e\}$, $\mathcal{X}^*$ is equipped with an associative operation, called concatenation, defined by

$$(x_1, \dots, x_n)(y_1, \dots, y_m) = (x_1, \dots, x_n, y_1, \dots, y_m).$$

We skip the brackets whenever this results in no confusion, in particular we write the letter x instead of (x). We also write $\mathcal{X}^+ = \mathcal{X}^* \setminus \{e\}$ for the set of non–empty words.

The **length** $| x^n |$ of the word $x^n = x_1 \dots x_n$ is the number n of letters in x^n.

A word $w \in \mathcal{X}^*$ is a **factor** of a word $x \in \mathcal{X}^*$ if there exist $u, v \in \mathcal{X}^*$ such that $x = uwv$. A factor w of x is **proper** if $w \neq x$.
For subsets $\mathcal{Y}, \mathcal{Z}$ of $\mathcal{X}^*$ and a word $w \in \mathcal{X}^*$, we define

$$\mathcal{Y}w = \{yw \in \mathcal{X}^* : y \in \mathcal{Y}\},$$

$$\mathcal{Y}\mathcal{Z} = \{yz \in \mathcal{X}^* : y \in \mathcal{Y}, z \in \mathcal{Z}\}$$

and

$$\mathcal{Y}w^{-1} = \{z \in \mathcal{X}^* : zw \in \mathcal{Y}\}.$$

A set of words $\mathcal{C} \subset \mathcal{X}^*$ is called a **code**.

Recall that a code is called prefix–free (resp. suffix–free), if no codeword is beginning (resp. ending) of another one.

Definition 1 *A code, which is simultaneously prefix–free and suffix–free, is called biprefix or fix–free. This can be expressed by the equations*

$$\mathcal{C}\mathcal{X}^+ \cap \mathcal{C} = \phi \text{ and } \mathcal{X}^+\mathcal{C} \cap \mathcal{C} = \phi.$$

Definition 2 *A code $\mathcal{C} = \{c_1, \dots, c_N\}$ over an a–letter alphabet $\mathcal{X}$ is said to be* **complete** *if it satisfies equality in Kraft's inequality, i.e. for $\ell_i = | c_i |$,*

$$\sum_{i=1}^{N} a^{-\ell_i} = 1.$$

Definition 3 *A fix–free code $\mathcal{C}$ is called* **saturated**, *if it is not possible to find a fix–free code $\mathcal{C}'$ containing $\mathcal{C}$ properly, that is, $| \mathcal{C}' | > | \mathcal{C} |$.*

2 The Existence

Lemma 1 *A finite fix–free code $\mathcal{C} = \{c_1, \ldots, c_N\}$ over $\mathcal{X} = \{0, \ldots, a-1\}$ is saturated iff $\mathcal{C}$ is complete.*

Proof:
Let $\ell_i =\mid c_i \mid$ for all $1 \leq i \leq N$.

1. If $\sum\limits_{i=1}^{N} a^{-\ell_i} = 1$, then $\mathcal{C}$ is saturated, because otherwise we get a contradiction to Kraft's inequality.

2. Now we show that in case $\sum\limits_{i=1}^{N} a^{-\ell_i} < 1$, $\ell_1 \leq \ldots \leq \ell_N$, we can add another codeword to $\mathcal{C}$.

   Indeed, by the proof of Kraft's inequality there exists a word $x^{\ell_N} \in \mathcal{X}^{\ell_N}$ such that no codeword of $\mathcal{C}$ is prefix of x^{ℓ_N}. Similarly, there exists a word $y^{\ell_N} \in \mathcal{X}^{\ell_N}$ such that no codeword of $\mathcal{C}$ is suffix of y^{ℓ_N}. Define now the new codeword

$$c_{N+1} = x^{\ell_N} y^{\ell_N}.$$

Definition 4 *We define the* **shadow** *of a word $w \in \mathcal{X}^*$ in* **level** *l as*

$$\begin{aligned} \delta_l(w) &= \{x^l \in \mathcal{X}^l : w \textit{ is prefix or suffix of } x^l\}. \\ &= w^{-1}\mathcal{X}^l \cup \mathcal{X}^l w^{-1}. \end{aligned}$$

For a set $\mathcal{Z}$ this notation is extended to

$$\delta_l(\mathcal{Z}) = \bigcup_{z \in \mathcal{Z}} \delta_l(z).$$

We are next looking for Kraft–type inequalities.

Lemma 2 $\sum\limits_{i=1}^{N} a^{-\ell_i} \leq \frac{1}{2}$ *implies that there exists a fix–free code $\mathcal{C}$ over $\mathcal{X} = \{0, \ldots, a-1\}$ with $\ell_1 \leq \ldots \leq \ell_N$ as lengths of codewords.*

Proof: We proceed by induction in the number of codewords. The case $N = 1$ being obvious we assume that we have found a fix–free code for $N - 1$ codewords. We present these words as vertices of a tree, where a word of length ℓ corresponds to a certain vertex on the ℓ–th level (in the usual way).

We count now all leaves of this tree in the ℓ_N'th level, which have one of the codewords as a prefix or as a suffix. (The shadow of the code in the ℓ_N's level.)

For each codeword c_i of length ℓ_i we thus count $a^{\ell_N-\ell_i}$ leaves, which have c_i as a prefix and also $a^{\ell_N-\ell_i}$ leaves, which have c_i a suffix. These sets need not be distinct. However, their total number does not exceed $2\sum_{i=1}^{N-1} a^{\ell_N-\ell_i}$. By our assumption this is smaller than a^{ℓ_N} and there is a leaf on the ℓ_N's level, which was not counted. The corresponding word can serve as our N–th codeword.

We define now γ as the largest constant such that for every integral tuple $(\ell_1, \ell_2, \dots, \ell_N)$ $\sum_{i=1}^{N} 2^{-\ell_i} < \gamma$ implies the existence of a binary fix–free code with lengths $\ell_1, \ell_2, \dots, \ell_N$.

Lemma 3 $\gamma \le \frac{3}{4}$.

Proof: For any $\gamma = \frac{3}{4} + \varepsilon, \varepsilon > 0$, choose k such that $2^{-k} < \varepsilon$. For the vector $(\ell_1, \dots, \ell_N) = (1, k, \dots, k)$ with $N = 2^{k-2} + 2$ we have

$$\sum_{i=1}^{N} 2^{-\ell_i} = \frac{1}{2} + 2^{-k}(2^{k-2} + 1) = \frac{3}{4} + 2^{-k} < \frac{3}{4} + \varepsilon.$$

However, there are exactly 2^{k-2} words of length k without a codeword c_1 as a prefix and a suffix and, since $1 + 2^{k-2} < N$, we have shown the nonexistence of a code with the desired parameters.

There is some evidence for the

Conjecture: $\gamma = \frac{3}{4}$.

For instance we have the following observation.

Lemma 4 *Suppose that*

$$\text{either } \ell_i = \ell_{i+1} \text{ or } 2\ell_i \le \ell_{i+1} \text{ for all } 1 \le i \le N. \tag{2.1}$$

Then $\sum_{i=1}^{N} 2^{-\ell_i} \le \frac{3}{4}$ *implies the existence of a binary fix–free code with these codeword lengths.*

Proof: We go by induction on the number n of different lengths occurring in

$$\ell_1 \le \ell_2 \le \ldots \le \ell_N.$$

Obviously the result is true, if there is only one length, that is, $n = 1$.

Assuming that we can construct a code with $n - 1$ different codeword lengths we show that we can construct a code with n different codeword lengths. Let M be the largest index i with $\ell_i < \ell_N$. Thus $\sum_{i=1}^{M} 2^{-\ell_i} \le \frac{3}{4}$

and by induction hypothesis we have a fix–free code $\mathcal{C}'$ with the lengths $\ell_1, \ldots, \ell_M$. We estimate now the shadow $\delta_{\ell_N}(\mathcal{C}')$. Actually, by 2.1 we get an exact formula:

$$|\delta_{\ell_N}(\mathcal{C}')| = 2\sum_{i=1}^{M} 2^{\ell_N - \ell_i} - \sum_{i=1}^{M} 2^{\ell_N - 2\ell_i} - 2 \sum_{1 \le i < j \le M} 2^{\ell_N - (\ell_i + \ell_j)}. \quad (2.2)$$

A code with lengths $\ell_1, \ldots, \ell_N$ is constructable exactly if

$$|\delta_{\ell_N}(\mathcal{C}')| \le 2^{\ell_N} - (N - M). \quad (2.3)$$

Writing $K = N - M$ and $\alpha = \sum_{i=1}^{M} 2^{-\ell_i}$ we get after division by 2^{ℓ_N} from (2.2) and (2.3) that sufficient for constructability is

$$2\alpha - \alpha^2 \le 1 - \frac{K}{2^{\ell_N}}.$$

With the abbreviations $\beta = \sum_{i=1}^{N} 2^{-\ell_i} = \alpha + \frac{K}{2^{\ell_N}}$ and $\delta = \frac{K}{2^{\ell_N}}$ we get the equivalent inequality

$$\beta \le 1 + \delta - \sqrt{\delta}.$$

This is satisfied for $\beta \le \frac{3}{4}$, because $1 + \delta - \sqrt{\delta}$ has the minimal value $\frac{3}{4}$ (at $\delta = \frac{1}{4}$).

2.1 Minimal Average Codeword Lengths

The aim of data compression in Noiseless Coding Theory is to minimize the average length of the codewords (see [2, 5]).

Theorem 1 *For each probability distribution $P = (P(1), \ldots, P(N))$ there exists a binary fix – free code $\mathcal{C}$ where the average length of the codewords satisfies*

$$H(P) \le \overline{L}(\mathcal{C}) < H(P) + 2.$$

Proof: The left–hand side of the theorem is clearly true, because each fix – free code is a prefix code and for each prefix code the left–hand side of the theorem follows from the Noiseless Coding Theorem. It is also clear, that this lower bound is reached for $N = 2^m$ $(m \in \mathbb{N})$ and $P(i) = 2^{-m}$ for all $1 \le i \le 2^m$.

The proof of the right–hand side of the Theorem is the same as the proof for alphabetic codes, which can be found in [1]:

We define $\ell_i \triangleq \lceil -\log(P(i)) \rceil + 1$. It follows that

$$\sum_{i=1}^{N} 2^{-\ell_i} \leq \frac{1}{2} \sum_{i=1}^{N} 2^{\log(P(i))} = \frac{1}{2} \sum_{i=1}^{N} P(i) = \frac{1}{2}.$$

By Lemma 2 there exists a fix – free code $\mathcal{C}$ with the codeword lengths $\ell_1, \ldots, \ell_N$.

The average length of this code is

$$\begin{aligned} \overline{L}(\mathcal{C}) &= \sum_{i=1}^{N} P(i)\ell_i < \sum_{i=1}^{N} P(i)(-\log(P(i)) + 2) \\ &= H(P) + 2\sum_{i=1}^{N} P(i) = H(P) + 2, \end{aligned}$$

where the logarithm is taken to the base 2. For an arbitrary alphabet the proof follows the same lines.

3 On Complete Fix–Free–Codes

3.1 Auxiliary Results

In Chapter 3 of [3] the structure of complete fix–free codes is studied and methods for constructing finite codes are presented. To each complete fix – free code two basic parameters are associated: its *kernel* and its *degree*. The kernel is the set of codewords which are proper factors of some codeword. The degree d is a positive integer which is defined as follows:
It is well known (see [3]) that for each finite complete fix – free code $\mathcal{C} = \{c_1, \ldots, c_N\}$ and for each $w \in \mathcal{X}^+$, there exists a positive integer $m < \max_{1 \leq i \leq N} \mid c_i \mid$ such that $\underbrace{w \ldots w}_{m} \in \mathcal{C}^*$. Now we define

$$d \triangleq \max_{w \in \mathcal{X}^+} \min_{m \in \mathbb{N}} \{m : \underbrace{w \ldots w}_{m} \in \mathcal{C}^*\}.$$

We need the following results of [3]:

Proposition 1 *Let $\mathcal{C}$ be a finite complete fix – free code over a finite alphabet $\mathcal{X}$ and let d be its degree. Then we have the properties:*

(i) For each letter $x \in \mathcal{X}$,

$$\underbrace{x \ldots x}_{d} \in \mathcal{C}.$$

(ii) There is only a finite number of finite complete fix–free codes over $\mathcal{X}$ with degree d.

(iii) If the length of the shortest codeword is d, then the length of every codeword is d as well.

Lemma 5 *For each finite complete fix–free code* $\mathcal{C} = \{c_1, \ldots, c_N)$ *over* $\mathcal{X} = \{0, \ldots, a-1\}$, a^2 *divides the number of codewords of maximal length.*

Proof : From the definition of complete fix–free codes it follows that with every codeword $c \in \mathcal{C}$ of maximal length, there are also $a^2 - 1$ other codewords which differ from c only in the first and/or last components. Hence the set of codewords of maximal length is a disjoint union of equivalent classes each of cardinality a^2. □

Lemma 6 *For each binary complete fix – free code* $\mathcal{C}$ *there is at most one codeword of length* 2 *or all codewords have length* 2.

Proof : By (i) in Proposition 1 we know that $\mathcal{C}$ contains no codeword of length one. If $\mathcal{C}$ contains a codeword c with $| c |> 2$ then by (iii) of Proposition 1 the degree of $\mathcal{C}$ is greater than 2, and by (i) of Proposition 1 $00 \notin \mathcal{C}$ and $11 \notin \mathcal{C}$. Hence if we have two codewords of length 2 then these two codewords are 01 and 10. However, there is a codeword of maximal length starting with 01 or 10 (see Lemma 5). □

3.2 Only Three Different Levels

Let $\mathcal{C}$ be a finite binary complete fix–free code and let $\mathcal{C}_i \triangleq \{c \in \mathcal{C} :| c |= i\}$. Let $bin^{-1}(c)$ be the natural number which corresponds to the binary representation of c (Note that the length of c is not fixed so that $bin^{-1}(c) = bin^{-1}(0c)$).

Lemma 7 *Let* $\mathcal{C} = (c_1, \ldots, c_N)$ *be a finite binary complete fix–free code with codeword lengths* $\ell_1, \ldots, \ell_N$ *satisfying* $\ell_i \in \{k, k+1, k+2\}$ *for all* $1 \le i \le N$ *and some* k. *Then for every* $\mathcal{E} \subset \mathcal{C}_k$
$| \delta_{k+1}(\mathcal{E}) | \ge 2 | \mathcal{E} |$ *and equality holds exactly if* $| \mathcal{E} |= 2^k$.

Proof : The union of the sets $\mathcal{E}0$ and $\mathcal{E}1$ contains $2 | \mathcal{E} |$ elements. Hence always $| \delta_{k+1}(\mathcal{E}) | \ge 2 | \mathcal{E} |$, if $| \mathcal{E} |< 2^k$ then by (i) and (iii) of Proposition 1, $(0, \ldots, 0) \notin \mathcal{E}$.
Let c be the element in $\mathcal{E}$ with smallest $bin^{-1}(c)$. We consider $0c \in \delta_{k+1}(\mathcal{E})$ and let us show that $0c \notin \mathcal{E}0 \cup \mathcal{E}1$. Assume in the opposite $0c = c'0$ or $0c = c'1$ for some $c' \in \mathcal{E}$. However $bin^{-1}(0c) = bin^{-1}(c) < 2bin^{-1}(c') = bin^{-1}(c'0)$ and $bin^{-1}(0c) < 1 + 2bin^{-1}(c') = bin^{-1}(c'1)$ hold, since c is the element of $\mathcal{E}$ with smallest $bin^{-1}(c)$. Hence $| \delta_{k+1}(\mathcal{E}) | \ge 2 | \mathcal{E} | +1$ if $| \mathcal{E} |< 2^k$. □

Theorem 2 *Let* $\mathcal{C}$ *be a finite binary complete fix – free code with codeword lengths:* $k = \ell_1 \le \ell_2 \le \ldots \le \ell_N = k+2$. *Then* *(i)* $xcy \in \mathcal{C}_{k+2}$, $x, y \in \{0,1\}$ *if and only if* $c \in \mathcal{C}_k$ *and*

(ii) $|\ \delta_{k+1}(\mathcal{C}_k)\ | = 4\ |\ \mathcal{C}_k\ |$.

Proof :

(i) Let $\mathcal{C}_k^0 = \big\{c \in \{0,1\}^k \setminus \mathcal{C}_k : xcy \in \mathcal{C}_{k+2}, x, y \in \{0,1\}\big\}$, $\mathcal{C}_{k+2}^0 = \big\{xcy \in \mathcal{C}_{k+2}, x, y \in \{0,1\} : c \in \mathcal{C}_k^0\big\}$ and let $\mathcal{D} = \delta_{k+1}(\mathcal{C}_k^0) = \big\{c0, c1, 0c, 1c \in \{0,1\}^{k+1} : c \in \mathcal{C}_k^0\big\}$. From Lemma 5 we know that $|\ \mathcal{C}_{k+2}^0\ | = 4\ |\ \mathcal{C}_k^0\ |$. We consider new codes $\mathcal{C}_1' = (\mathcal{C} \setminus \mathcal{C}_{k+2}^0) \cup \mathcal{C}_k^0$ and $\mathcal{C}_2' = (\mathcal{C} \setminus \mathcal{C}_{k+2}^0) \cup \mathcal{D}$. It can be easely verified, that both $\mathcal{C}_1'$ and $\mathcal{C}_2'$ are fix–free codes. Moreover, $\mathcal{C}_1'$ is complete, since $\mathcal{C}$ is complete. Therefore we can apply Lemma 7 with respect to $\mathcal{E} = \mathcal{C}_k^0, |\ \mathcal{C}_k^0\ | < 2^k$, to get $|\ \delta_{k+1}(\mathcal{C}_k^0)\ | = |\ \mathcal{D}\ | > 2\ |\ \mathcal{C}_k^0\ |$. However this leads to the contradiction, because $\mathcal{C}_2'$ is a fix-free code, but

$$\begin{aligned}\sum_{c \in \mathcal{C}_2'} 2^{-|c|} &= \sum_{c \in (\mathcal{C} \setminus \mathcal{C}_{k+2}^0)} 2^{-|c|} + \sum_{c \in \mathcal{D}} 2^{-|c|} \\ &> \sum_{c \in (\mathcal{C} \setminus \mathcal{C}_{k+2}^0)} 2^{-|c|} + \sum_{c \in \mathcal{C}_{k+2}^0} 2^{-|c|} \\ &= \sum_{c \in \mathcal{C}} 2^{-|c|} = 1.\end{aligned}$$

(ii) We consider teh lower shadow of $\mathcal{C}_{k+2}$:

$$\delta_{k+1}^{-}(\mathcal{C}_{k+2}) \triangleq \Big\{c \in \{0,1\}^{k+1} : \delta_{k+2}(c) \cap \mathcal{C}_{k+2} \neq \emptyset\Big\}.$$

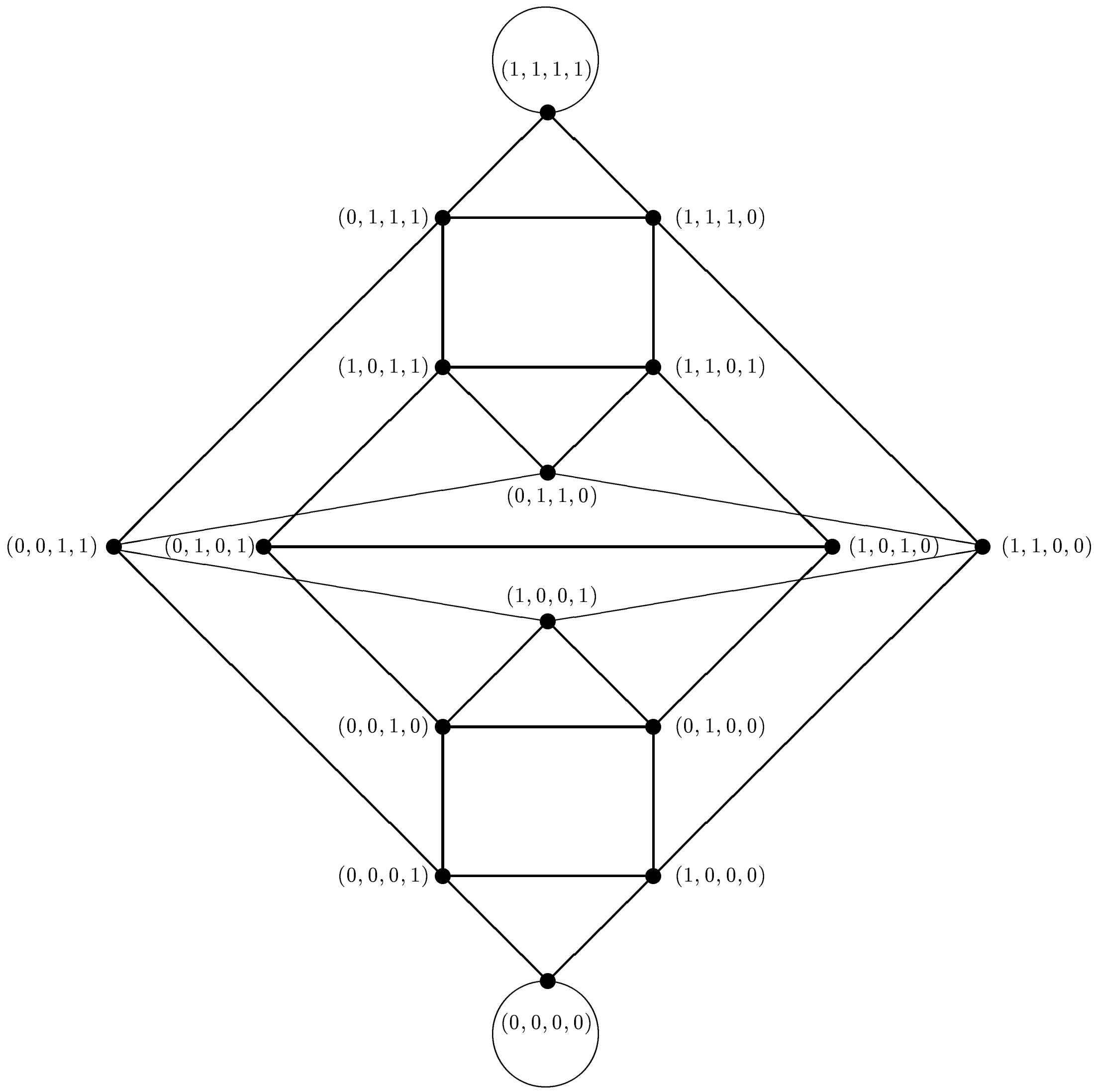
By (i) we have $\delta_{k+1}^{-}(\mathcal{C}_{k+2}) = \delta_{k+1}(\mathcal{C}_k)$.
Therefore $\mathcal{C}_{k+1} = \{0,1\}^{k+1} \setminus \delta_{k+1}(\mathcal{C}_k)$, since $\mathcal{C}$ is complete.
Now $|\ \delta_{k+1}(\mathcal{C}_k)\ | < 4\ |\ \mathcal{C}_k\ |$ would imply $\sum_{c \in \mathcal{C}} 2^{-|c|} > 1$. □

3.3 Relations to the deBruijn Network

The **binary deBruijn Network** of order n is an undirected graph $\mathcal{B}^n = (\mathcal{V}^n, \mathcal{E}^n)$, where $\mathcal{V}^n = \mathcal{X}^n$ is the set of vertices and $(u^n, v^n) \in \mathcal{E}^n$ is an edge iff

$$u^n \in \{(b, v_1, \ldots, v_{n-1}), (v_2, \ldots, v_n, b) : b \in \{0,1\}.$$

The binary deBruijn Network $\mathcal{B}^4$ is given as an example:



A subset $\mathcal{A} \subset \mathcal{V}^n$ is called independent, if no two vertices of $\mathcal{A}$ are connected, and we denote by $\mathcal{I}(\mathcal{B}^n)$ the set of all independent subsets of the deBruijn network. We note, that for all $b \in \{0,1\}$, $(b, b, \dots b) \notin \mathcal{A} \in \mathcal{I}(\mathcal{B}^n)$, because $(b, b, \dots b)$ is dependent itself. The independence number $f(n)$ of $\mathcal{B}^n$ is $f(n) = \max\limits_{\mathcal{A} \in \mathcal{I}(\mathcal{B}^n)} \mid \mathcal{A} \mid$.

Lemma 8 *Let $\mathcal{C}$ be a binary complete fix – free code on three levels: $\mathcal{C} = \mathcal{C}_n \cup \mathcal{C}_{n+1} \cup \mathcal{C}_{n+2}, \mathcal{C}_i \neq \emptyset$. Then*

(i) $\mathcal{C}_n \in \mathcal{I}(\mathcal{B}^n)$ and

(ii) for every $\mathcal{A} \in \mathcal{I}(\mathcal{B}^n)$ there exists a complete fix – free code on three levels $n, n+1, n+2$ for which $\mathcal{A} = \mathcal{C}_n$, and the code is unique.

Proof :

(i) Immideately follows from Theorem 2 (ii).

(ii) For an $\mathcal{A} \in \mathcal{I}(\mathcal{B}^n)$ we construct a complete fix – free code $\mathcal{C} = \mathcal{C}_n \cup \mathcal{C}_{n+1} \cup \mathcal{C}_{n+2}$ as follows: $\mathcal{C}_{n+1} = \{0,1\}^{n+1} \setminus \delta_{n+1}(\mathcal{A})$, $\mathcal{C}_{n+2} = \{xcy \in \{0,1\}^{n+2}, x, y \in \{0,1\} : c \in \mathcal{A}\}$. □

We note, that the exact value of the independence number $f(n)$ of $\mathcal{B}^n$ in general is not known.
Clearly for any $x^n, y^n \in \mathcal{A} \in \mathcal{I}(\mathcal{B}^n), x^n \neq y^n$:

$$bin^{-1}(x^n) \neq 2bin^{-1}(y^n), bin^{-1}(x^n) \neq 2bin^{-1}(y^n) + 1,$$

$$bin^{-1}(x^n) \neq bin^{-1}(y^n) + 2^{n-1} bin^{-1}(y^n) \neq 2bin^{-1}(x^n),$$

$$bin^{-1}(y^n) \neq 2bin^{-1}(x^n) + 1, bin^{-1}(y^n) \neq bin^{-1}(x^n) + 2^{n-1}$$

Hence, the determination of $f(n)$ is a special case of the following number–theoretical problem:
For given $m \in \mathbb{N}$, find a set $\mathcal{S} = \{1 \leq a_1 < \ldots < a_s < m\}$ of maximal cardinality, for which $\{a_i, 2a_i, 2a_i+1, a_i+m\} \cap \{a_j, 2a_j, 2a_j+1, a_j+m\} = \emptyset$ holds for all $1 \leq i < j \leq \mid \mathcal{S} \mid$.
In the case $m = 2^n$ we have exactly the problem of finding a maximal independent set with cardinality $f(n)$ in the deBruijn network. Hence we solve this problem (for $m = 2^n$) asymptotically.

Theorem 3

$$\lim_{n \to \infty} \frac{f(n)}{2^n} = \frac{1}{2}.$$

Proof : Let $\mathcal{A} \in \mathcal{I}(\mathcal{B}^n)$ with $\mid \mathcal{A} \mid = f(n)$. Clearly $f(n) < 2^{n-1}$, because for an $x^n \in \mathcal{A}$:

$$1 \leq bin^{-1}(x^n) < 2bin^{-1}(x^n) < 2bin^{-1}(x^n) + 1 < bin^{-1}(x^n) + 2^n \leq 2^{n+1} - 1$$

and these integers are different for different elements of $\mathcal{A}$. It is easy to see, that always $f(n+1) \geq 2f(n)$, and hence the $\lim\limits_{n \to \infty} \frac{f(n)}{2^n}$ exists. To finish the proof, we have to construct a sequence of sets $\mathcal{A}_n \in \mathcal{I}(\mathcal{B}^n)$ with $\lim\limits_{n \to \infty} \frac{|\mathcal{A}_n|}{2^n} = \frac{1}{2}$. For this it suffices to construct only for even values of n.
Let

$$\mathcal{S}_0^n = \left\{ x^n \in \{0,1\}^n : \sum_{i=1}^{\frac{n}{2}} x_{2i} > \sum_{i=1}^{\frac{n}{2}} x_{2i-1} \right\}$$

and

$$\mathcal{S}_1^n = \left\{ x^n \in \{0,1\}^n : \sum_{i=1}^{\frac{n}{2}} x_{2i} < \sum_{i=1}^{\frac{n}{2}} x_{2i-1} \right\}.$$

Clearly $\mid \mathcal{S}_0^n \mid = \mid \mathcal{S}_1^n \mid$,

$$| \{0,1\}^n \setminus (\mathcal{S}_0^n \cup \mathcal{S}_1^n) | \quad = \sum_{i=0}^{\frac{n}{2}} \binom{\frac{n}{2}}{i}^2 = \binom{n}{\frac{n}{2}}.$$

Hence $| \mathcal{S}_0^n | = \frac{2^n - \binom{n}{\frac{n}{2}}}{2}$, and $\lim\limits_{n\to\infty} \frac{|\mathcal{S}_0^n|}{2^n} = \frac{1}{2}$.
It is easely seen that $\mathcal{S}_0^n \in \mathcal{I}(\mathcal{B}^n)$ and we set $\mathcal{A}_n = \mathcal{S}_0^n$. □

4 Computer Results

1.) For $2 \leq n \leq 6$ we have calculated the independent number $(f(n))$ of the binary deBruijn network of order n via a computer program. A maximal independent set $\mathcal{S} = \{1 \leq a_1 < \ldots < a_s < 2^n\}$ is greedy constructable as follows:
If n is odd we take $a_1 = 1$ and $a_1 = 2$ otherwise. Now if a_i is choosen in a step we take in the next one a_{i+1} as the smallest possible number greater than a_i.
From this constructions we obtain that

$f(n) = \frac{4}{9}2^n - \frac{4}{9} - \frac{n}{6}$ and $f(n) = 2f(n-1) + \frac{n}{2}$, if n is even and

$f(n) = \frac{4}{9}2^n - \frac{5}{9} - \frac{n}{3}$ and $f(n) = 2f(n-1)$, if n is odd.

For even n the set $| \mathcal{S} | < | \mathcal{S}_0^n |$ (see Theorem 3) for $n = 8$ and for all $n \geq 52$.

2.) In [4] one finds an example of a complete fix – free code with the codeword lengths

$$2, 3, 3, 3, 3, 4, 4, 4, 4,$$

We know from (i) of Proposition 1 that it is not possible to choose 00 or 11 as codeword of length 2 for this code.
This result suggests the question: “Suppose there is a fix – free code with codeword lengths $\ell_1 \leq \ldots \leq \ell_t$, $l_1 > 1$. Is it possible to construct a fix–free code with these length, where the codewords of smallest length are not the all–zero vector and the all–one vector ?”
The following fix – free code $\{11, 000, 100, 010, 001, 10110\}$ with lengths $2, 3, 3, 3, 3, 5$ shows that the answer is negative. Indeed, assume that the codeword of length 2 is 01. There are exactly 4 codewords of length 3 which are prefix – and suffix free with 01: $000, 100, 110, 111$.
Suppose there is a codeword *abcde* of length 5. Let us show that it is impossible.

Necessary $d = 1$,	because in case $d = 0$,we have $e = 0$, for otherwise, the codeword 01 would be suffix. However, 00 is excluded, because otherwise 000 or 100 would be suffix.
$c = 0$,	because for $c = 1$ we get 110 or 111 as suffix.
$b = 1$,	because for $b = 0$ we get 000 or 100 as prefix.
Finally $a \neq 0$,	because for $a = 0$ we get 01 as prefix.
and $a \neq 1$,	because for $a = 1$ we get 110 as prefix. □

This is a contradiction.

3.) We present an example of a complete binary fix – free code for each possible length–distribution $\mathcal{L}$ with $\mid \mathcal{L} \mid \leq 29$:

0 1
2 : 2 x 1

01 00 10 11
4 : 4 x 2

000 001 010 011 100 101 110 111
8 : 8 x 3

01 000 100 110 111
0010 1010 0011 1011
9: 1 x 2 + 4 x 3 + 4 x 4

0000 1000 0100 1100 0010 1010 0110 1110
0001 1001 0101 1101 0011 1011 0111 1111
16 : 16 x 4

001 0000 1000 0100 1100 1010 0110 1110
0101 1101 1011 0111 1111 00010 10010 00011
10011
17 : 1 x 3 + 12 x 4 + 4 x 5

001 110 0000 1000 0100 1010 0101 1011
0111 1111 01100 11100 00010 10010 01101 11101
00011 10011
18 : 2 x 3 + 8 x 4 + 8 x 5

001 100 0000 1010 0110 1110 0101 1101
1011 0111 1111 01000 11000 00010 00011 010010
110010 010011 110011
19 : 2 x 3 + 9 x 4 + 4 x 5 + 4 x 6

001 100 101 0000 0110 1110 0111 1111
01000 11000 00010 01010 11010 00011 01011 11011
010010 110010 010011 110011
20 : 3 x 3 + 5 x 4 + 8 x 5 + 4 x 6

001 010 011 0000 1000 1100 1110 1101
1111 10100 10110 10101 10111 000100 100100 000110
100110 000101 100101 000111 100111
21 : 3 x 3 + 6 x 4 + 4 x 5 + 8 x 6

01 0000 1000 1100 1110 1111 00100 10100
00010 10010 11010 00110 10110 00011 10011 11011
00111 10111 001010 101010 001011 101011

22 : 1 x 2 + 5 x 4 + 12 x 5 + 4 x 6

001 100 110 0000 1010 0101 1011 0111
1111 01000 00010 01101 11101 00011 011000 111000
010010 010011 0110010 1110010 0110011 1110011

22 : 3 x 3 + 6 x 4 + 5 x 5 + 4 x 6 + 4 x 7

01 0000 1000 1100 1110 0011 1111 00100
10100 00010 10010 11010 10110 11011 10111 001010
101010 000110 100110 001011 101011 000111 100111

23 : 1 x 2 + 6 x 4 + 8 x 5 + 8 x 6

01 0000 1000 1100 0010 1110 1111 10100
11010 00110 10110 00011 10011 11011 00111 10111
000100 100100 101010 101011 0001010 1001010 0001011 1001011

24 : 1 x 2 + 6 x 4 + 9 x 5 + 4 x 6 + 4 x 7

001 100 110 101 0000 0111 1111 01000
00010 01010 00011 01011 011000 111000 010010 011010
111010 010011 011011 111011 0110010 1110010 0110011 1110011

24 : 4 x 3 + 3 x 4 + 5 x 5 + 8 x 6 + 4 x 7

01 0000 1000 1100 0010 1110 0011 1111
10100 11010 10110 11011 10111 000100 100100 101010
000110 100110 101011 000111 100111 0001010 1001010 0001011
1001011

25 : 1 x 2 + 7 x 4 + 5 x 5 + 8 x 6 + 4 x 7

01 100 0000 1110 1111 11000 00010 11010
00110 10110 00011 11011 00111 10111 001000 101000
110010 001010 101010 110011 001011 101011 0010010 1010010
0010011 1010011

26 : 1 x 2 + 1 x 3 + 3 x 4 + 9 x 5 + 8 x 6 + 4 x 7

10 0000 0100 0001 1101 0011 0111 1111
11000 01100 11100 11001 00101 01011 001000 001001
010101 011011 111011 0101000 0101001 0110101 1110101 01101000
11101000 01101001 11101001

27 : 1 x 2 + 7 x 4 + 6 x 5 + 5 x 6 + 4 x 7 + 4 x 8

10 001 0000 1101 0111 1111 01000 11000
01100 11100 00011 01011 000100 010100 000101 010101
010011 110011 011011 111011 0100100 1100100 0110100 1110100
0100101 1100101 0110101 1110101

28 : 1 x 2 + 1 x 3 + 4 x 4 + 6 x 5 + 8 x 6 + 8 x 7

10	0000	0100	1100	0001	1101	0011	0111
1111	00101	01011	001000	011000	111000	001001	011001
111001	010101	011011	111011	0101000	0101001	0110101	1110101
01101000	11101000	01101001	11101001				

28 : 1 x 2 + 8 x 4 + 2 x 5 + 9 x 6 + 4 x 7 + 4 x 8

10	001	0000	1100	0111	1111	01000	01101
11101	00011	01011	11011	011000	111000	000100	010100
110100	000101	010101	110101	010011	0100100	0100101	0110011
1110011	01100100	11100100	01100101	11100101			

29 : 1 x 2 + 1 x 3 + 4 x 4 + 6 x 5 + 9 x 6 + 4 x 7 + 4 x 8

References

[1] R. Ahlswede and I. Wegener, Suchprobleme,
Teubner, Stuttgart, 1979.

[2] R.B. Ash, Information theory,
Interscience Tracts in Pure and Applied Mathematics 19,
Interscience, New York, 1965.

[3] Jean Berstel and Dominique Perrin,
Theory of codes, Pure and Applied Mathematics, 1985.

[4] David Gillman and Ronald L. Rivest,
Complete variable – length fix – free – codes,
Designs, Codes and Cryptography, 5, 109–114, 1995.
1995 Kluwer Academic Publishers, Boston.
Manufactured in The Netherlands.

[5] C.E. Shannon, Prediction and entropy of printed English,
Bell Systems Technical Journal 30, 50–64, 1951.