

1. Proof that Champernowne is not supernormal.

For this proof we will assume the Champernowne sequence as the concatenation of all words of length k for $k = 1, 2, \dots$

For every fixed k , the block $X(k) = \text{“ All the words of length } k \text{ ”}$ is a perfect necklace. Then $X(k)$ has length $k2^k$ and if we look at $X(k)$ as a necklace, all the words of length k occur exactly k times, in positions of $X(k)$ that are different modulo k .

First, let's notice that if we analyze the occurrences of words of length $k + \log(k) + 1$ in the first $2^{k+\log(k)+1}$ symbols of Champernowne, the whole block $X(k)$ will be covered, it is easy to prove by induction that $2^{k+\log(k)+1} = 2k2^k > \sum_{i=1}^k i2^i$.

Secondly, as $X(k)$ has length $k2^k$, the block $X(k)$ accounts for half of the total amount of symbols in the first $2k2^k$ symbols of Champernowne.

Now, let's take a look at what the words of length $k + \log(k) + 1$ that occur in $X(k)$ look like. There are four different cases that can happen of how a word x is formed with elements from $X(k)$. In the following analysis, u, v and w are consecutive words of length k in $X(k)$:

■ Case 1:

$$x = u_1 u_2 \dots u_k \quad v_1 v_2 \dots v_{\log(k)} v_{\log(k)+1}$$

Which means it is the occurrence modulo 0 for a given word u of length k in $X(k)$ plus the remaining $\log(k) + 1$ symbols which are taken from the next word.

■ Case 2:

$$x = u_{k-\log(k)-1} \dots u_k \quad v_1 v_2 \dots v_k$$

Which is the case where the word of length $k + \log(k) + 1$ is formed from the last $\log(k) + 1$ symbols of a word and the whole k symbols of the next word.

■ Case 3:

$$x = u_{n+1} u_{n+2} \dots u_k \quad v_1 v_2 \dots v_{\log(k)+n+1}$$

with $n \in \{1, 2, \dots, k - \log(k) - 2\}$.

Which is the case where the $k + \log(n) + 1$ symbols are taken from two words of length k and none of the words is complete.

■ Case 4:

$$x = u_{k-\log(k)-1+n} u_{k-\log(k)+n} \dots u_k \quad v_1 v_2 \dots v_k \quad w_1 w_2 \dots w_{\log(k)+1-n}$$

with $n \in \{1, 2, \dots, \log(k)\}$

Which is the case where the word of length $k + \log(k) + 1$ is formed by a full word, and the extra $\log(k) + 1$ symbols are taken from both the end of the previous word and the beginning of the next one.

1.1. Case Analysis

The idea will be to count which are the different words that happen in each case to bound the amount of different words that occur in the first $2k2^k$ symbols of Champernowne and check that there are fewer words that occur at least one time than the expected amount.

For the simplicity of the proof we will define $next(w)$ as $|w|$ times 0 if w only consists of 1s and the word that comes after w in lexicographic order in any other case.

1.1.1. Case 1

$$x = u_1 u_2 \dots u_k \quad v_1 v_2 \dots v_{\log(k)} v_{\log(k)+1}$$

This case accounts for the occurrence modulo 0 for a given word u of length k in $X(k)$ plus the remaining $\log(k) + 1$ symbols which are taken from the next word. As an example, some of the words of length $k + \log(k) + 1$ formed from $X(k)$ taking $k = 8, \log(k) = 3$ are shown between brackets:

(00000000	0000) 0001
(00000001	0000) 0010
(00000010	0000) 0011
⋮	
(00001110	0000) 1111
(00001111	0001) 0000
(00010000	0001) 0001
⋮	
(11111110	1111) 1111

There are two important things to notice here. The first one is that as the words of length $k + \log(k) + 1$ are formed by a full word of length k followed by the first $\log(k) + 1$ symbols from the next word, in almost case the first $\log(k) + 1$ symbols are equal to the last $\log(k) + 1$. The only way for this not to happen, is when the last $k - \log(k) - 1$ symbols from the first word u are all 1s, which means that the next word in lexicographic order v will consist of $next(v_1 v_2 \dots v_{\log(k)+1})$ concatenated with $next(v_{\log(k)+2} v_{\log(k)+3} \dots v_k)$.

The second important thing to notice is that as $X(k)$ is the concatenation of all words of length k , all words of length k occur one time in an alligned position modulo k . This means that the first k symbols of x will take every possible configuration.

These two facts leave two possible schemes for what a word x of case 1 may look like:

$$\begin{array}{ccc}
 \underbrace{A}_{\log(k)+1} & \underbrace{B}_{k-\log(k)-1} & A \\
 \underbrace{A}_{\log(k)+1} & \underbrace{11\dots 1}_{k-\log(k)-1} & next(A)
 \end{array}$$

For the first scheme we have:

$$\begin{aligned}
 & 2^{\log(k)+1} (2^{k-\log(k)-1} - 1) \\
 & 2k \left(\frac{2^k}{2k} - 1 \right) \\
 & 2^k - 2k
 \end{aligned}$$

$2^k - 2k$ different words.

For the second scheme we subtract one two the cases due to the fact that the last word of length k in $X(k)$ has its continuation outside $X(k)$:

$$\begin{aligned}
 & 2^{\log(k)+1} - 1 \\
 & 2k - 1
 \end{aligned}$$

$2k - 1$ different words.

Counting the whole case together we have $2^k - 2k + 2k - 1$ which is less than 2^k different words.

1.1.2. Case 2

$$x = u_{k-\log(k)-1} \dots u_k \quad v_1 v_2 \dots v_k$$

This case accounts for the occurrence modulo 0 for a given word u of length k in $X(k)$ plus the remaining $\log(k) + 1$ symbols which are taken from the previous word. This means that in this case the word of length $k + \log(k) + 1$ corresponding to the first word of $X(k)$ does not have a corresponding word inside $X(k)$. As an example, some of the words of length $k + \log(k) + 1$ formed from $X(k)$ taking $k = 8, \log(k) = 3$ are shown between brackets:

$$\begin{array}{ccc} 0000 & (0000 & 00000001) \\ 0000 & (0001 & 00000010) \\ & \vdots & \\ 0000 & (1111 & 10000000) \\ 1000 & (0000 & 10000001) \\ & \vdots & \\ 1111 & (1110 & 11111111) \end{array}$$

As in the previous case, as $X(k)$ is the concatenation of all words of length k , all words of length k occur one time in an aligned position modulo k . This means that the last k symbols of x will take every possible configuration. The other important thing to notice is that as the first $\log(k) + 1$ symbols of x come from the word u which occurs exactly before v in lexicographic order, then:

$$\text{next}(u_{k-\log(k)-1} u_{k-\log(k)} \dots u_k) = u_{v-\log(k)-1} v_{k-\log(k)} \dots v_k$$

This leaves only one possible scheme for what a word x of case 2 may look like:

$$\underbrace{\quad}_{\log(k)+1} \quad \underbrace{\quad}_{k-\log(k)-1} \quad \text{next}(A)$$

This scheme gives us the following amount of different words that may occur:

$$(2^{\log(k)+1} 2^{k-\log(k)}) - 1$$

$$2k \left(\frac{2^k}{2^k} \right) - 1$$

$$2^k - 1$$

$2^k - 1$ different words which is less than 2^k different words.

1.1.3. Case 3

$$x = u_{n+1} u_{n+2} \dots u_k \quad v_1 v_2 \dots v_{\log(k)+n+1}$$

with $n \in \{1, 2, \dots, k - \log(k) - 2\}$.

This case accounts for when the $k + \log(n) + 1$ symbols are taken from two words of length k and none of the words is complete. As an example, some of the words of length $k + \log(k) + 1$ formed from $X(k)$ taking $k = 8, \log(k) = 3$ are shown between brackets. Some extra spaces are added within u and v to make clear the scheme that will be explained later. Taking $n = 1$.

$$\begin{array}{ccc} 0 & (0000 & 000 & 0 & 0000) & 001 \\ 0 & (0000 & 001 & 0 & 0000) & 010 \\ & \vdots & & & & \end{array}$$

0 (0001 110	0 0001) 111
0 (0001 111	0 0010) 000
0 (0010 000	0 0010) 001
⋮	
1 (1111 101	1 1111) 110
0 (1111 110	1 1111) 111

Taking $n = k - \log(k) - 2 = 3$

000 (0000 0	000 0000) 1
000 (0000 1	000 0001) 0
000 (0001 0	000 0001) 1
000 (0001 1	000 0010) 0
⋮	
111 (1110 1	111 1111) 0
111 (1111 0	111 1111) 1

In this case, it also happens that as $X(k)$ is the concatenation of all words of length k , for each value of n , all words of length k take the u position once, except the last of the words of length k in $X(k)$.

It is important to notice that for a given value of n , the first n symbols of u will not be considered to form x . This means that it can be interpreted that the symbols from u that are considered are, the first $\log(k) + 1$ symbols after n which will be called A and the remaining $k - \log(k) - 1 - n$ symbols which will be called B .

Now, if we divide the $n + \log(k) + 1$ symbols that are used from v to form x into the first n symbols which will be called C and the remaining $\log(k) + 1$ symbols, it is possible to see that these $\log(k) + 1$ symbols will always be equal to the symbols from A except for the case where $B = 11 \dots 1$ as they account for the same indexes of u and v and v comes immediately after u in lexicographic order.

These leave two possible schemes for what a word x of case 3 may look like:

$\underbrace{\quad}_{\log(k)+1} A$	$\underbrace{\quad}_{k-\log(k)-1-n} B$	$\underbrace{\quad}_n C$	A
$\underbrace{\quad}_{\log(k)+1} A$	$\underbrace{11 \dots 1}_{k-\log(k)-1-n}$	$\underbrace{\quad}_n C$	$next(A)$

Looking closely at the first scheme, it is possible to see, if we put together B and C which have length $k - \log(k) - 1 - n$ and n respectively, that we have the following scheme:

$\underbrace{\quad}_{\log(k)+1} A$	$\underbrace{\quad}_{k-\log(k)-1} B$	A
------------------------------------	--------------------------------------	-----

which is exactly the same one as in case 1. This means that all the possible words that can be formed following this scheme don't yield any new words.

The same thing happens with the second scheme when concatenating $11 \dots 1$ with C :

$\underbrace{\quad}_{\log(k)+1} A$	$\underbrace{11 \dots 1C}_{k-\log(k)-1}$	$next(A)$
------------------------------------	--	-----------

Which is a particular case of case 2.

This means that for case 3 there are no words that appear that should be taken into account as new words.

1.1.4. Case 4

$$x = u_{k-\log(k)-1+n} u_{k-\log(k)+n} \dots u_k \quad v_1 v_2 \dots v_k \quad w_1 w_2 \dots w_{\log(k)+1-n}$$

with $n \in \{1, 2, \dots, \log(k)\}$

This case accounts for when the $k + \log(n) + 1$ symbols are taken from three words of length k . The k symbols of v are used and the remaining $\log(k) + 1$ symbols are taken from both the previous and the following words u and w . As an example, some of the words of length $k + \log(k) + 1$ formed from $X(k)$ taking $k = 8, \log(k) = 3$ are shown between brackets. Some extra spaces are added within u and v to make clear the scheme that will be explained later.

Taking $n = 1$.

$$\begin{array}{lll} 0000000 (0 & 000 0000 1 & 000) 00010 \\ 0000000 (1 & 000 0001 0 & 000) 00011 \\ & \vdots & \\ 0011111 (0 & 001 1111 1 & 001) 10000 \\ 0011111 (1 & 010 0000 0 & 010) 00001 \\ & \vdots & \\ 1111110 (1 & 111 1111 0 & 111) 11111 \end{array}$$

Taking $n = 2$.

$$\begin{array}{lll} 000000 (00 & 00 0000 01 & 00) 000010 \\ 000000 (01 & 00 0000 10 & 00) 000011 \\ & \vdots & \\ 001111 (01 & 00 1111 10 & 00) 111111 \\ 001111 (10 & 00 1111 11 & 01) 000000 \\ 001111 (11 & 01 0000 00 & 01) 000001 \\ & \vdots & \\ 111111 (01 & 11 1111 10 & 11) 111111 \end{array}$$

In this case, it also happens that as $X(k)$ is the concatenation of all words of length k , for each value of n , all words of length k take the u position once, except the last of the words of length k in $X(k)$.

We will call A the first n symbols of x which are taken from the end of v . The following $\log(k) + 1 - n$ symbols which are the first of v will be called B and it happens that unless the remaining symbols of v are all 1s, they will be the same as the last $\log(k) + 1 - n$ of x because these symbols are the first $\log(k) + 1 - n$ from w . Now, we will consider the remaining $k - \log(k) - 1 + n$ symbols from v as two blocks, one block C of length $k - \log(k) - 1$ and the remaining n symbols which are exactly $next(A)$ as v is the next word in lexicographic order after u . This yields the two following schemes:

$$\begin{array}{lllll} \underbrace{A}_{n} & \underbrace{B}_{\log(k)+1-n} & \underbrace{C}_{k-\log(k)+1} & next(A) & B \\ \underbrace{11 \dots 10}_n & \underbrace{B}_{\log(k)+1-n} & \underbrace{11 \dots 1}_{k-\log(k)+1} & next(A) & next(B) \end{array}$$

For the first scheme we have:

$$\begin{aligned} & \sum_{n=1}^{\log(k)} (2^n - 1)(2^{\log(k)+1-n})(2^{k-\log(k)-1}) \\ & \sum_{n=1}^{\log(k)} (2^n - 1) \left(\frac{2k}{2^n}\right) \left(\frac{2^k}{2k}\right) \end{aligned}$$

$$2^k \sum_{n=1}^{\log(k)} \frac{2^n - 1}{2^n}$$

Here we would need to subtract the cases for when v is the last word of $X(k)$, however as we are giving an upper bound for the words that appear in Champernowne, the result still holds taking this number.

For the second scheme we have:

$$\begin{aligned} & \sum_{n=1}^{\log(k)} 2^{\log(k)+1-n} \\ & \sum_{n=1}^{\log(k)} 2k2^{-n} \\ & 2k \sum_{n=1}^{\log(k)} 2^{-n} \end{aligned}$$

When putting them both together we get:

$$2^k \sum_{n=1}^{\log(k)} \frac{2^n - 1}{2^n} + 2k \sum_{n=1}^{\log(k)} 2^{-n} < \log(k)2^k + 2k$$

1.2. Bounding words that appear in Champernown

The only remaining thing to do is to consider all the cases together and bound the total amount of words that appear in the first $2^{k+\log(k)+1}$ symbols of Champernowne. So the amount of different words of length $k + \log(k) + 1$ that appear in $X(k)$ is less than:

$$2^k + 2^k + \log(k)2^k + 2k$$

As it was stated before, $X(k)$ accounts for half of the total symbols that need to be analyzed. Let's consider that the other $k2^k$ are all different words. That gives us:

$$k2^k + 2^k + 2^k + \log(k)2^k + 2k$$

Now, if Champernowne is supernormal, then the expected amount of words that appear in the first $2^{k+\log(k)+1}$ would be $2k2^k(1 - e^{-1})$. Let's see that

$$k2^k + 2^{k+1} + \log(k)2^k + 2k < 2k2^k(1 - e^{-1})$$

$$\frac{k + 2 + \log(k)}{2k} + \frac{2k}{2^k} < 1 - e^{-1}$$

$$\frac{1}{2} + \frac{1}{k} + \frac{\log(k)}{2k} < 1 - e^{-1}$$

which holds taking k sufficiently large.