Positional numeral systems with sequences as the base

Conrad sen Kyne

Abstract

The mechanics of the positional numeral system are reviewed. Its definition is gradually expanded, first to the factorial base, then to Fibonacci coding, to the recursive signature function, and finally to any increasing one-beginning natural sequence. Then various edge cases are explored where the sequence base displays some irregular property.

Contents

1 Basic Counting Methods

1.1 Positional numeral system

We take for granted how useful our system of counting is. The **positional numeral system** is a way of cleanly representing any integer using a *base* b and a *digitation* $\{n_0, n_1, ..., n_p\}$. In our **base-ten system**, we separate numbers by ones, tens, hundreds, thousands, and so on, each represented by a different digit. Take, for example, the number 1482. We can partition this number like so:

$$1482 = 1000 + 400 + 80 + 2 \tag{1}$$

What is happening here is each digit from right to left is multiplied by progressively higher powers of ten. We can expand this equation further to get:

$$1482 = \mathbf{1} \cdot 10^3 + \mathbf{4} \cdot 10^2 + \mathbf{8} \cdot 10^1 + \mathbf{2} \cdot 10^0 \tag{2}$$

Formally, the positional numeration of a set $N_b = \{b, \{n_0, n_1, ..., n_p\}\}$ is given by

$$\sum_{k=0}^{p} n_k b^k \tag{3}$$

Let's try counting in a different base. Let's try base-3, or ternary. Take a look at this table:

b^k	$\rightarrow$	9	3	1
N_{10}				
$\downarrow$				
1				1
2				2
3			1	0
4			1	1
5			1	2
6			2	0
7			2	1
8			2	2
9		1	0	0
10		1	0	1
11		1	0	2
12		1	1	0
13		1	1	1
14		1	1	2
15		1	2	0
16		1	2	1
17		1	2	2
18		2	0	0
19		2	0	1

Note that each number's base-ten representation (N_{10}) is computed by multiplying each digit by its corresponding power of 3. For example, the number 112_3 is computed by:

$$112_3 = \mathbf{1} \cdot 9 + \mathbf{1} \cdot 3 + \mathbf{2} \cdot 1 = 9 + 3 + 2 = 14 \tag{4}$$

Note that we are implicitly using base-ten to represent numbers whose base is not explicitly labelled.

1.2 Digit resolution

Digit resolution is the process of carrying a number to the next digit, then resetting the lower number to 0. In base-ten, this occurs when counting from 9 to 10; in base-2, from 1 to 10; and in octal, from 7 to 10. Some examples can be viewed below.

$$2_2 \to 10_2$$

$$120_2 \to 200_2 \to 1000_2$$

$$8_8 \to 10_8$$

$$84_8 \to 104_8$$

$$C_{12} \to 10_{12}$$

$$BC7_{12} \to C07_{12} \to 1007_{12}$$

1.3 Factorial base system

What if we could substitute a function in place of b^k? Well, it turns out one such function, the *factorial* function, can do just that. Take a look at the following table:

$k!$	$\rightarrow$	24	6	2	1	1
N_{10}						
$\downarrow$						
1					1	0
2				1	0	0
3				1	1	0
4				2	0	0
5				2	1	0
6			1	0	0	0
7			1	0	1	0
8			1	1	0	0
9			1	1	1	0
10			1	2	0	0
11			1	2	1	0
12			2	0	0	0
13			2	0	1	0
14			2	1	0	0
15			2	1	1	0
16			2	2	0	0
17			2	2	1	0
18			3	0	0	0
19			3	0	1	0
20			3	1	0	0
21			3	1	1	0
22			3	2	0	0
23			3	2	1	0
24		1	0	0	0	0

Instead of the power function b^k, each digit is equivalent to $n_k k!$.
Notice, for example, that going from 23 to 24 in factorial involves the following resolution:

$$3210_! + 1 = 3211_! \rightarrow 3220_! \rightarrow 3300_! \rightarrow 4000_! \rightarrow 10000_! \tag{5}$$

1.4 Fibonacci coding

What if instead of base-10 or the factorial base, we used a different function as the base? In this case, let's use the Fibonacci numbers, given by

$$F(0) = F(1) = 1 \quad F(n) = F(n-1) + F(n-2) \tag{6}$$

$F(k)$	$\rightarrow$	13	8	5	3	2	1	1
N_{10}								
$\downarrow$								
1							1	0
2						1	0	0
3					1	0	0	0
4					1	0	1	0
5				1	0	0	0	0
6				1	0	0	1	0
7				1	0	1	0	0
8			1	0	0	0	0	0
9			1	0	0	0	1	0
10			1	0	0	1	0	0
11			1	0	1	0	0	0
12			1	0	1	0	1	0
13		1	0	0	0	0	0	0

This is similar to counting in binary, except it *resolves the digit pattern 1 to 10 and 11 to 100*. Observe what happens when counting from 12 to 13:

$$101010_F + 1 = 101011_F \rightarrow 101100_F \rightarrow 110000_F \rightarrow 1000000_F \tag{7}$$

We can use this table to calculate the base-10 value of any number on the table. For example, the value of 1010_F is calculated by

$$\mathbf{1} \cdot 3 + \mathbf{0} \cdot 2 + \mathbf{1} \cdot 1 + \mathbf{0} \cdot 1 = 3 + 1 = 4 \tag{8}$$

The value of 101000_F is calculated by

$$\mathbf{1} \cdot 8 + \mathbf{0} \cdot 5 + \mathbf{1} \cdot 3 + \mathbf{0} \cdot 2 + \mathbf{0} \cdot 1 + \mathbf{0} \cdot 1 = 8 + 3 = 11 \tag{9}$$

2 Alternate Counting Methods

2.1 The signature function

To understand what is happening in the Fibonacci coding, let's look at a generalization of the Fibonacci function called the **recursive signature function**. The recursive signature function (or just the signature function for short) is defined as

$$F_d(0) = 1 \qquad F_d(n) = \sum_{k=0}^{n-1} F_d(n-1-k) d_k \tag{10}$$

For example, $F_{\{1,1\}}$ is the Fibonacci numbers and $F_{\{2,1\}} = \{1, 2, 5, 12, 29, 70, ...\}$ is the Lucas numbers. The signature function reveals an exciting way to count similar to Fibonacci coding but for an arbitrary sequence d.

2.2 Signature numeration

Let's attempt to use the sequence $F_{\{1,1,2\}}$ to create a "coding" like the Fibonacci coding. First, the sequence is given by

$$F_{\{1,1,2\}} = \{1, 1, 2, 5, 9, 18, ...\} \tag{11}$$

Observe the coding given below:

$F_{\{1,1,2\}}(k)$ →	18	9	5	2	1	1
N_{10} ↓						
1					1	0
2				1	0	0
3				1	1	0
4				1	1	1
5			1	0	0	0
6			1	0	1	0
7			1	1	0	0
8			1	1	1	0
9		1	0	0	0	0
10		1	0	0	1	0
11		1	0	1	0	0
12		1	0	1	1	0
13		1	0	1	1	1
14		1	1	0	0	0
15		1	1	0	1	0
16		1	1	1	0	0
17		1	1	1	1	0
18	1	0	0	0	0	0
19	1	0	0	0	1	0
20	1	0	0	1	0	0
21	1	0	0	1	1	0
22	1	0	0	1	1	1
23	1	0	1	0	0	0

Like Fibonacci coding, this is resolving digits, but according to the sequence $\{1, 1, 2\}$ rather than $\{1, 1\}$. For example, counting from 4 to 5 is done by

$$111_{\{1,1,2\}} + 1 = 112_{\{1,1,2\}} \rightarrow 1000_{\{1,1,2\}}$$

whereas counting from 8 to 9 is done by

$$1110_{\{1,1,2\}} + 1 = 1111_{\{1,1,2\}} \rightarrow 1120_{\{1,1,2\}} \rightarrow 10000_{\{1,1,2\}}$$

In general, the **signature numeration** of $N_d = \{d, \{n_0, n_1, ..., n_p\}\}$ is given by

$$\sum_{k=0}^{p} n_k F_d(k) \tag{12}$$

For example, Fibonacci coding is the case where $d = \{1, 1\}$. Some more examples of this numeration:

$$F_{\{1,1,2\}} = \{1, 1, 2, 5, 9, ...\} \quad 11010_{\{1,1,2\}} = \mathbf{1}{\cdot}9{+}\mathbf{1}{\cdot}5{+}\mathbf{0}{\cdot}2{+}\mathbf{1}{\cdot}1{+}\mathbf{0}{\cdot}1 = 9{+}5{+}1 = 15 \tag{13}$$

$$F_{\{2,1\}} = \{1, 2, 5, 12, 29, ...\} \quad 2011_{\{2,1\}} = \mathbf{2}\cdot12{+}\mathbf{0}\cdot5{+}\mathbf{1}\cdot2{+}\mathbf{1}\cdot1 = 24{+}2{+}1 = 27 \tag{14}$$

$$F_{\{2,3\}} = \{1, 2, 7, 20, 61, ...\} \quad 1221_{\{2,3\}} = \mathbf{1}{\cdot}20{+}\mathbf{2}{\cdot}7{+}\mathbf{2}{\cdot}2{+}\mathbf{1}{\cdot}1 = 20{+}14{+}4{+}1 = 39 \tag{15}$$

2.3 Arbitrary sequence numeration

2.3.1 Conditions for sequence as a base

Instead of F_d, we can select an arbitrary discrete function f which meets the following conditions:

$$f(0) = 1$$
$$f(n) \in \mathbb{N}$$
$$f(n) >= f(n-1)$$

In other words, any **increasing one-beginning natural sequence** may be used as a base.

These sequences are a subset of the signature function, whose range is all one-beginning sequences, so we can refer to these sequences by their signature when convenient.

2.3.2 Base-{2, 0, -1}

For example, we can count in base-$\{2, 0, -1\}$. The sequence $F_{\{2,0,-1\}}$ is given by:

$$F_{\{2,0,-1\}} = \{1, 2, 4, 7, 12, 20, ...\} \tag{16}$$

Then we can count in this base like so:

$F_{\{2,0,-1\}}(k)$ →	20	12	7	4	2	1
N_{10} ↓						
1						1
2					1	0
3					**1**	**1**
4				**1**	**0**	**0**
5				1	0	1
6				**1**	**1**	**0**
7			**1**	**0**	**0**	**0**
8			1	0	0	1
9			1	0	1	0
10			1	0	1	1
11			**1**	**1**	**0**	**0**
12		**1**	**0**	**0**	**0**	**0**
13		1	0	0	0	1
14		1	0	0	1	0
15		1	0	0	1	1
16		1	0	1	0	0
17		1	0	1	0	1
18		1	0	1	1	0
19		**1**	**1**	**0**	**0**	**0**
20	**1**	**0**	**0**	**0**	**0**	**0**

Notice the bolded sections which showcase the identity

$$F_{2,0,-1}(n) = F_{2,0,-1}(n-1) + F_{2,0,-1}(n-2) + 1 \tag{17}$$

2.4 Edge cases

Let's take a look at some special cases where counting follows some less intuitive rules.

2.4.1 Base-1

Base-1 is fairly simple: every digit either does not exist or is a 1, so we have

$$N_1 = \sum_{k=0}^{p} 1 = p + 1 \tag{18}$$

For example, 1111 is 4, 1111111 is 7, and 11 is 2.

2.4.2 Sequences which converge

Suppose we select the sequence $f = \{1, 4, 7, 8, 8, 8, ...\}$ to serve as a base for numeration. Observe the behavior of counting in this base:

$f(k)$ $\to$	8	8	8	7	4	1
N_{10} $\downarrow$						
1						1
2						2
3						3
4					1	0
5					1	1
6					1	2
7				1	0	0
8			1	0	0	0
9			1	0	0	1
10			1	0	0	2
11			1	0	0	3
12			1	0	1	0
13			1	0	1	1
14			1	0	1	2
15			1	1	0	0
16		1	1	0	0	0
17		1	1	0	0	1
18		1	1	0	0	2
19		1	1	0	0	3
20		1	1	0	1	0
21		1	1	0	1	1
22		1	1	0	1	2
23		1	1	1	0	0
24	1	1	1	0	0	0

Notice that, similar to base-1, the digits with a value of 8 can only have a value of 1. This still satisfies the condition that

$$N_f = \sum_{k=0}^{p} n_k f(k) \tag{19}$$

where, for example, $11000_f = 16$, $1010_f = 12$, etc. This behavior occurs for any f which converges.

2.4.3 Sequences where f(n) = f(n-1) for some n

Suppose we select the sequence $f = \{1, 3, 7, 7, 13, ...\}$ to serve as a base for numeration. Observe the behavior of counting in this base:

$f(k)$	$\to$	13	7	7	3	1
N_{10}						
$\downarrow$						
1						1
2						2
3					1	0
4					1	1
5					1	2
6					1	3
7			**1**	**0**	**0**	**0**
8			1	0	0	1
9			1	0	0	2
10			1	0	1	0
11			1	0	1	1
12			1	0	1	2
13		1	0	0	0	0
14		1	0	0	0	1
15		1	0	0	0	2
16		1	0	0	1	0
17		1	0	0	1	1
18		1	0	0	1	2
19		1	0	0	1	3
20		**1**	**1**	**0**	**0**	**0**

Note that the digit resolves past the first 7 to the "highest" 7. This behavior would be similar for $f = \{1, 3, 7, 7, 7, 13, ...\}$ or $f = \{1, 3, 7, 7, ..., 7, 13, ...\}$.

3 Conclusion

I hope that I have given this topic a proper treatment, even though there are some weaknesses to this method. For example, radix points do not work for sequences, so we are restricted to integer representations.

Still, I think this method is interesting in its own right. Because the space of increasing one-beginning natural sequences is uncountably infinite, we may now access an uncountably infinite number of positional numeral systems with sequences as the base. Perhaps these new systems will show that our "normal" way of counting is a bit less special, but is part of something a lot more beautiful.