Number Drafting, Part 3: A Trip to a Strange Place

The first article in this series [1] described the representation of numbers in bases other than 10. Binary numbers (base 2), octal numbers (base 8), and hexadecimal numbers (base 16) are the most familiar because of their importance in relating computation to computer architecture. But any base larger than 1 is possible.

For bases up to 10, the ordinary digits suffice. Beyond 10, other characters are needed. For base 16, it is conventional to add the letters a, b, … f, with the correspondence between characters and numerical values as follows:

0	1	…	9	a	b	…	f
↓	↓	…	↓	↓	↓	…	↓
0	1	…	9	10	11	…	15

In this system, the decimal number 1496 written in base 16 is 5d8.

Sometimes uppercase letters are used in place of lowercase ones. I doesn't matter which, and the only reason for using the first six letters of the alphabet is ease of recognition. For example, the letters g, o, r, m, e, t could be used. It would make no difference to a computer program, but human beings would find this notation harder to learn than the one using the first six letters of the alphabet.

It's even possible to dispense with digits and use, say, the first 16 letters of the alphabet for representing the hexadecimal digits:

a	b	…	i	j	k	…	p
↓	↓	…	↓	↓	↓	…	↓
0	1	…	9	10	11	…	15

In this system, the decimal number 1496 written in base 16 is fni.

Except for practicality, any 16 characters could be used for representing the 16 hexadecimal digits. Here's a "dingbat" version:

❁ ❂ ❃ ❄ ❅ ❆ ❇ ❈ ❉ ❊ ❋ ● ❍ ■ ❏ ❐

In this system, the decimal number 1496 written in base 16 is ❆ ■ ❉.

All this may seem weird, but, as in madness, there's a method.

In analogy to code drafting [2], strings consisting of uppercase letters and the blank can be commemorated in a weave by considering them to be numbers in base 27. The notation for base 27 numbers could be

␢	A	B	C	D	…	Z
↓	↓	↓	↓	↓	…	↓
0	1	2	3	4	…	26

For example, the string

PLACE YOUR BETS

considered as a base-27 number can be converted to other bases. Using conventional notation, for base 8 the result is

303064343307021135007466

and for base 16, the result is

618d1c6c7089740f36

The corresponding sequences, replacing the digits by their values, 1-adjusted, are

4,1,4,1,7,5,4,5,4,4,1,8,1,3,2,2,4,6,1,1,8,5,7,7

and

7,2,9,14,2,13,7,13,8,1,9,10,8,5,1,16,4,7

A weave derived from the base-8 representation appears at the end of this article.

Note: The reason for using the blank to represent 0 is that leading zeros are discarded in arithmetic contexts: The numbers 1496 and 01496 are the same. In ordinary text, leading blanks are generally considered to be irrelevant and are ignored. If a letter, such as A , had been chosen to represent 0, it would be lost in leading positions in number drafting: AMBIGUOUS and MBIGUOUS would represent the same number.

Comment

As strange as this method of number drafting may seem, it is bland compared to an extension to 95 characters as described in an article on code drafting [3].

The principles described here extend naturally to larger character sets. However, while some existing programs can handle base 27 in the normal notation, base 95 is another matter.

If you are a programmer, however, it is easy to write a program to handle any base and in any notation.

References

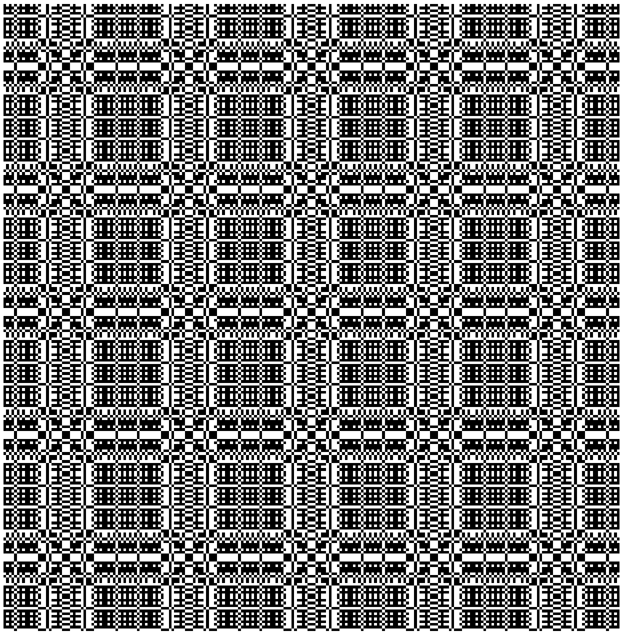
1. *Number Drafting, Part 1: Introduction*, Ralph E. Griswold, 2004: http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_num1.pdf.

2. *Code Drafting, Part 1: Introduction*, Ralph E. Griswold, 2004: http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_cd1.pdf.

3. *Code Drafting, Part 3: A Larger Character Set*, Ralph E. Griswold, 2004: http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_cd3.pdf.

Ralph E. Griswold

Department of Computer Science
The University of Arizona
Tucson, Arizona



PLACE YOUR BETS