American Journal of Computational Mathematics, **2012, 2, 244-247**
doi:10.4236/ajcm.2012.23033 Published Online September 2012 (http://www.SciRP.org/journal/ajcm)

Base-X Notation and Tri-Value Logic

Xiaolin Li
Department of Mathematics and Computer Science, Alabama State University, Montgomery, USA
Email: xli@alasu.edu

Received April 29, 2012; revised July 3, 2012; accepted July 13, 2012

ABSTRACT

This paper introduces the base-X notation and discusses the conversion between numbers of different bases. It also introduces the tri-value logic that is associated with the base-3 system.

Keywords: Index Terms—Base-X Notation; Tri-Value Logic

1. Introduction

People use decimal numbers in their daily life. However, our computers use the binary system and the binary logic [1-3]. A computer also involves other number systems such as octal numbers and hexadecimal numbers. There is an easy way to convert a number in one system to another using the base-X notation.

2. Base-X Notation

Let X be an integer, $X > 1$. Using X as the base, we can form a number system. A base-X system uses the digit set $\{0, 1, \cdots, X-1\}$ to constitute a number. Any number Y in the system can be written as:

$$Y = Y_m Y_{m-1} \cdots Y_1 Y_0 \cdot Y_{-1} Y_{-2} \cdots Y_{-n} \tag{1}$$

where m and n are integers, $0 \le m < \infty$, $1 \le n < \infty$. In the above notation, Y_i is the digit at position i and Y_i is in $\{0, 1, \cdots, X-1\}$, $-n \le i \le m$. When $X = 2$, we have the binary system; when $X = 8$, we have the octal system; when $X = 10$, we have the decimal system; when $X = 16$, we have the hexadecimal system.

Let us consider how to convert Y into a decimal number when $X \neq 10$. Notice that for each position i, there is an associated weight X^i, and Y_i represents a term $Y_i X^i$. Thus, adding all terms together, we get the corresponding decimal number:

$$\begin{aligned} Y &= Y_m Y_{m-1} \cdots Y_1 Y_0 \cdot Y_{-1} Y_{-2} \cdots Y_{-n} \\ &= Y_m X^m + Y_{m-1} X^{m-1} + \cdots + Y_1 X + Y_0 + Y_{-1} X^{-1} \\ &\quad + Y_{-2} X^{-2} + \cdots + Y_{-n} X^{-n} \\ &= \sum_{i=-n}^{m} Y_i X^i \end{aligned} \tag{2}$$

The following examples illustrate the above conversion, where the subscript indicates the base:

1) Binary to Decimal

$$\begin{aligned} (11001.101)_2 &= \left(1\times 2^4 + 1\times 2^3 + 1 + 1\times 2^{-1} + 1\times 2^{-3}\right)_{10} \\ &= \left(25\frac{5}{8}\right)_{10} \end{aligned}$$

The binary system has the advantage that it uses only two digits: 0 and 1. Within a binary number, digit 0 always represents a zero term while digit 1 always represents a term identical to the weight associated with the digit position. Notice that we have dropped all zero terms in the above summation.

2) Octal to Decimal

$$\begin{aligned} (756.23)_8 &= \left(7\times 8^2 + 5\times 8 + 6 + 2\times 8^{-1} + 3\times 8^{-2}\right)_{10} \\ &= \left(494\frac{19}{64}\right)_{10} \end{aligned}$$

3) Hexadecimal to Decimal

$$\begin{aligned} &(5FE \cdot 2B)_{16} \\ &= \left(5\times 16^2 + 15\times 16 + 14 + 2\times 16^{-1} + 11\times 16^{-2}\right)_{10} \\ &= \left(1534\frac{43}{256}\right)_{10} \end{aligned}$$

The digit set used in the hexadecimal system is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$, where $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, $F = 15$.

3. Decimal to Base-X Conversion

The conversion of a decimal number to a base-X number is also indicated by Equation (2). Given a decimal number Y, we can treat Y as the summation of a sequence of terms where each term is a power of X, then we can write down the corresponding base-X number. For example, given the decimal number 25.625, we have

$$(25.625)_{10} = \left(16+8+1+\frac{1}{2}+\frac{1}{8}\right)_{10},$$

therefore, $(25.625)_{10} = (11001.101)_2$. In general, one can use the following procedure for the conversion.

Let $Y = Y_i \cdot Y_f$, where Y_i and Y_f are the integer portion and fraction portion of Y, respectively. We can convert Y_i and Y_f separately:

1) Regarding Y_i as the initial quotient and dividing the remained quotient by X repeatedly until it becomes zero, we get a sequence of remainder digits, the integer portion of the base-X number then can be obtained by reversing the order of the remainder digits in the sequence;

2) Regarding Y_f as the initial fraction and multiplying the remained fraction by X repeatedly, we get a sequence of digits from the integer portion of the products, the fraction portion of the base-X number then can be obtained from this digit sequence.

Consider the decimal number 123.65625 for example, we can convert it into binary, octal and hexadecimal number:

1) Decimal to Binary

Integer portion conversion:

Quotient	Remainder
123	
61	1
30	1
15	0
7	1
3	1
1	1
0	1

Fraction portion conversion:

Fraction	Integer Product
0.65625	
0.3125	1
0.625	0
0.25	1
0.5	0
0	1

$$(123.65625)_{10} = (1111011.10101)_2$$

2) Decimal to Octal

Integer portion conversion:

Quotient	Remainder
123	
15	3
1	7
0	1

Fraction portion conversion:

Fraction	Integer Product
0.65625	
0.25	5
0	2

$$(123.65625)_{10} = (173.52)_8$$

3) Decimal to Hexadecimal

Integer portion conversion:

Quotient	Remainder
123	
7	*B*
0	7

Fraction portion conversion:

Fraction	Integer Product
0.65625	
0.5	*A*
0	8

$$(123.65625)_{10} = (7B \cdot A8)_{16}$$

Notice that in the above conversion, we always get a finite digit sequence for the integer portion because the remained quotient always becomes zero by repeated division. On the other hand, we may get an infinite digit sequence for the fraction portion if the remained fraction never becomes zero through the repeated multiplication. In such a case, it may produce an endless circulate fraction.

For example, $(0.4)_{10} = (0.011001100110\cdots)_2$.

4. Base-3 System and Tri-Value Logic

Let us consider the base-3 number system that uses the digit set {0, 1, 2} to constitute a number.

Denote $T = \{0, 1, 2\}$ and let A, B in T, the results of $A + B$ are shown in **Table 1**.

Consider a tri-value logic associated with the digit set T. Let 0, 1, 2 represent false, true, and neutral (neither false nor true) state, respectively. We can generalize the basic logic operators $\cap$ (logic and), $\cup$ (logic or) and $'$(logic not) in Boolean algebra to this tri-value case such that $\cap : T^2 \to T$, $\cup : T^2 \to T$, and $' : T \to T$. Let A and B be two variable, A, B in T, then the definition of these generalized logic operators is given in **Table 2**.

Notice that if we delete all rows that contain state 2 from **Table 2**, then we get the truth table in Boolean algebra as **Table 3**.

From **Table 2**, it is easy to see that the following equalities hold:

$$A \cap A = A$$

$$A \cup A = A$$

$$0 \cap A = 0$$

$$0 \cup A = A$$

$$1 \cap A = A$$

$$1 \cup A = 1$$

It is obvious that the De Morgan's law and the binary implication can be extended into this tri-value case:

$$(A \cap B)' = A' \cup B'$$

$$(A \cup B)' = A' \cap B'$$

$$A \to B = A' \cup B$$

In addition to the above, we can also define other compound logic operators from the basic logic operators $\cap$, $\cup$, $'$, such as the exclusive or operator $\oplus : T^2 \to T$:

$$A \oplus B = A \cap B' \cup A' \cap B$$

5. Conclusion

In this paper, we have introduced the base-X notation and have discussed the conversion between numbers of different bases. We have also introduced a tri-value logic that is associated with the base-3 system. We have shown that the tri-value logic is compatible with the binary logic, and the De Morgan's law and the implication rule can be extended into this tri-value case. Here we point out that all base-X systems are equivalent to each other because any number in one system can be uniquely mapped into

Table 1. Base-3 addition.

A/B	0	1	2
0	0	1	2
1	1	2	10
2	2	10	11

Table 2. Tri-value logic.

A	B	$A \cap B$	$A \cup B$	A'	B'	$A' \cap B'$	$A' \cup B'$
0	0	0	0	1	1	1	1
0	1	0	1	1	0	0	1
0	2	0	2	1	2	2	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0
1	2	2	1	0	2	0	2
2	0	0	2	2	1	2	1
2	1	2	1	2	0	0	2
2	2	2	2	2	2	2	2

Table 3. Binary logic.

A	B	$A \cap B$	$A \cup B$	A'	B'	$A' \cap B'$	$A' \cup B'$
0	0	0	0	1	1	1	1
0	1	0	1	1	0	0	1
1	0	0	1	0	1	0	1
1	1	1	1	0	0	0	0

another system as implied by Equation (2). Furthermore, any rational number in one system remains rational in another system because its integer numerator and denominator can be converted into integers in another system as indicated by the integer conversion process.

REFERENCES

[1] J. Sanchez and M. P. Canton, "Microcontroller Programming: The Microchip PIC," CRC Press, Boca Raton, 2006.

[2] J. E. Whitesitt, "Boolean Algebra and Its Applications," Dover Publications, New York, 2010.

[3] S. Givant and P. Halmos, "Introduction to Boolean Algebras (Undergraduate Texts in Mathematics)," Springer, New York, 2008.