

4. p 進法、論理演算規則、セルオートマトン

はやや進んだ話題である。

数字を様々な言語で表現すると、歴史的経緯の一端がみえる。英語では、「one, two, ..., ten, eleven, twelve, thirteen, ...」と 12 進法の名残がある。独語も同様。日本語、中国語は 10 進法、計算機言語は 2 進法である。

二進法、三進法、.....、 p 進法の説明。十進法とは、 $0, 1, 2, \dots, 9$ を使って数字を表わし、10 になったら位を上げる表現法です。二進法とは、 $0, 1$ を使って数字を表わし、2 になったら位を上げる表現法。人間の指の数が 1 本なら 2 進法、2 本なら 4 進法を主に使うようになっていたであろう。十進法による表現と区別するために、十進法以外の表現については、小さな括弧を付けて表わす。例えば、 $10_{(2)} = 2, 100_{(2)} = 4, 111_{(2)} = 7, 1111_{(2)} = 15$ 。

十二進法では 12 個の文字、十六進法では 16 個の文字が必要。このため、0 から 9 までを $0, 1, 2, \dots, 7, 8, 9$ で表わし、10 以上は通常 a, b, c, d, \dots で表わす。また、12 ヶ月で 1 年、7 日で 1 週間など。時間などで使っている 60 進法もおなじみ。例えば、 $1b_{(12)} = 1 \times 12 + 11 = 23, ff_{(16)} = 15 \times 16 + 15 = 255$ 。

3 進法や (-3) 進法も存在する。 $0, 1, 2$ のみで負の数を使わずに全ての実数を表現できる。これが採用されると、掛け算の「99」も「33」となる。[説明] 通常の三進法では $0, 1, 2$ で数を表すが、これを変え、 $0, -1, +1$ で数を表すことにする。今仮にこれを $0, -, +$ と書くことにする。マイナスが入ることがポイントである。

$$\begin{aligned} 0 &= 0, \\ 1 &= +, \\ 2 &= + - = 3 - 1 \\ 3 &= + 0 \\ 4 &= + + = 3 + 1 \\ 5 &= + - - = 9 - 3 - 1 \end{aligned}$$

(1)

計算機などでは、電気が流れる $\rightarrow 1$ 、電気が流れない $\rightarrow 0$ の 2 つを組合せて様々な処理を行っていて、二進法に対応 $\rightarrow 0$ 。もしくは、磁化されていいる $\rightarrow 1$ 、磁化されていないでメモリー(記憶)を表す。正の整数 m を十進法から二進法に変換するのは次のようにする。 m を x に代入する。 x を 2 で割って、余りを求める。 $x/2$ の商を x に代入する。 $x = 0$ であれば終了。そうでなければ上記の操作を繰り返す。余りを求めた順の逆と並べると、それが二進法に変換された結果になる。もちろんマイナスをとって正の数と

同じ方法で二進法表記をつくりマイナスをつければ良い。-15610 = -100111002 である。

パソコンでは特に 8 個のビットをまとめて 1 バイトと言い、もっとも基本的な単位として非常に重要。ブール代数は、論理代数とも呼ばれ、論理回路の演算を対象として考え出された。加減乗の演算において特別な処理。

論理積演算子 (AND 回路)

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 1 \cdot 0 = 0, \quad 1 \cdot 1 = 1$$

論理和演算子 (OR 回路)

$$0 + 0 = 0, \quad 0 + 1 = 1 + 0 = 1, \quad 1 + 1 = 1$$

論理否定演算子 (NOT 回路)

$$0 \rightarrow 1, 1 \rightarrow 0,$$

以下は、上記のもの組み合わせ。排他的論理積演算子 (NAND 回路), 排他的論理積演算子 (NOR 回路)

セルオートマトンの導入。初期条件 $a(x, 0) = \delta_{x,0}$ 、規則、

$$a(t, x) = a(t-1, x-1) + a(t-1, x+1). \quad (2)$$

の作図。排他的論理和演算子 (XOR 回路)

$$0 + 0 = 0, \quad 0 + 1 = 1 + 0 = 1, \quad 1 + 1 = 0$$

参考文献

- [1] 合原一幸 編「カオス」サイエンス社