

【補数(twos)】

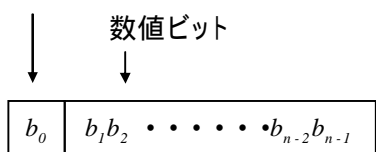
< 補数とは >

計算機における負数の表現方法は3通りある。(1)符号と絶対値, (2)1の補数, (3)2の補数の3通り存在する。符号と絶対値は乗除算に, 1及び2の補数は加減算に使用されている。

(1)符号と絶対値

符号と絶対値の方法は例えば『+123』と書く方法を2進数で表現した方式である。但し, ±も0と1で表現しなければならないため, +を0, -を1に対応させて表現する。

符号ビット(0:正, 1:負)



計算機は数値を図1のような形式で表現されている。左端の1ビットが符号ビットであり, 残りのビットが数値ビットである。数値ビットが整数なのか小数なのかはソフトウェアで区別している。

実際の数値例を図2に示す。

図1:固定小数点数の形式

10進数	符号と絶対値		
+4	表現できない		
+3	0	1	1
+2	0	1	0
+1	0	0	1
+0	0	0	0

10進数	符号と絶対値		
-0	1	0	0
-1	1	0	1
-2	1	1	0
-3	1	1	1
-4	表現できない		

図2:整数の表現方法(符号と絶対値)

(2)2の補数

正の数は符号と絶対値による表現と同じである。ここでは負の数を補数として表示する方法について述べる。理解を容易にするため, 扱う数値を小数の場合と整数の場合とに分けて考えていく。

(a)小数の場合

負の小数を-Aとすれば

$$-A = (1 - A) - 1 \quad - (1)式$$

と表現することができる。(1-A)は1より小さい正の小数であるので,

$$1 - A = 0b_1b_2 \dots b_{n-1} \quad - (2)式$$

とおける。(1)式の右辺第2項の-1と1と置き換えて

$$A = 1b_1b_2 \dots b_{n-1} \quad - (3)式$$

と表す。これを2の補数という。もう少し掘り下げて考えると, (1)式, (2)式より

$$A = 2 - A \quad \text{もしくは} \quad A + A = 2 \quad - (4)式$$

が導かれる。従ってAは基数に対する補数と呼ばれているのである。例えば

$$A = 0.01011_{(2)}$$

の場合,

$$A = 2_{(10)} - 0.01011_{(2)}$$

に対する補数を求めるため、減基数の補数あるいは擬補数とも呼ばれている。実際の値の例を図 5 に示す。

10 進数	1 の補数		
+4	表現できない		
+3	0	1	1
+2	0	1	0
+1	0	0	1
+0	0	0	0

10 進数	1 の補数		
-0	1	1	1
-1	1	1	0
-2	1	0	1
-3	1	0	0
-4	表現できない		

図 5: 整数の表現方法(1 の補数)

なお、1 の補数を機械的に求めるには、求めたい負数の絶対値を 2 進数表現し、その値の 0 と 1 を全てのビットで反転させればよい。つまり 2 の補数は 1 の補数に 1 を加えたものとおなじであるといえる。

それでは最後に実際に補数を使用して計算を行ってみる。ただし、ここでは 2 の補数を使用して計算を行う方法のみにする。この理由は現在の計算機で最もよく使用されている手法が 2 の補数であり、この手法を理解することが第一と考えたからである。

例) 以下の計算を 2 の補数を利用して計算しなさい。但し有効桁数は 7 桁とする。

$$123 - 58$$

有効桁数が 7 桁であるため、123, 58 を 2 進数 7 桁で表現すると以下のようになる。

$$123 = 1111011 \quad 58 = 0111010$$

ここで、-58 の補数 A を求めると、絶対値を 2 進数表示したあと 0 と 1 を反転させ、それに 1 を加えたものであることから、

$$A = 1000110$$

となる。従って求める解は 123 と A を加えて、

$$123 + A = 1111011 + 1000110 = 11000001$$

となり、有効桁数が 7 桁であることから

$$123_{(10)} - 58_{(10)} = 1000001_{(2)} = 65_{(10)}$$

となる。以上より補数を利用して減算を行うことができる。