

# ソフトウェア 1 課題 (lec4 宿題 2)

340481H 電子情報工学科内定 中里徳彦

2014/10/30

N ビットのときの  $n$  の 2 の補数とは、 $2^N - n$  のことである。例えば 4 ビットで  $n=3$  の補数は

$$10000(2) - 0011(2) = 1101(2) \text{ (10 進法で 13)}$$

である。 $n$  の 2 の補数はビットを反転 ( $2^N - 1 - n$  を計算) し +1 することで求めることができる。コンピュータでは  $-n (n > 0)$  を  $n$  の 2 の補数として表現することが多い。つまり -3 は 1101(2) として表現される。

このように定義することによって、負の数の和を求める際に、正の数同士の和を求めるのと同じ方法を使うことができる。N ビットにおいて任意の  $n, m (n, m > 0, n, m < 2^{N-1}, n, m \text{ は整数})$  で、 $n + (-m)$  を計算することを考える。2 の補数を用いると  $-m = 2^N - m$  と表現できる。これを用いて  $-m$  を正の数と考えて計算すると、

$$n + (-m) = n + 2^N - m = (n - m) + 2^N = n - m \pmod{2^N}$$

である。N ビットの場合を考えているので  $\pmod{2^N}$  とした。このように負の数の加算においても正の数の加算と同じ方法を用いることができるのが負の数に 2 の補数表現を用いた時の利点である。

ただし負の補数表現を用いると、先頭 1 ビットが 1 の時は負の数となるので表現できる最大数が  $2^N - 1$  ではなく、 $2^{N-1} - 1$  となる。