コラッツ予想に対する考察

吟遊詩人

１．準備

コラッツ予想の考察を始める事前準備として、本稿におけるいくつかの２進数の記法に対する定義をしておく。

定義

１．一般：
本稿において２進数の表記をする際に 101001 などと表記し、101001b などとは書かない。

２．有効ビット：
２進数表記で「1」のあるビットを**有効ビット**と呼ぶ。ある２進数の中にある「1」の個数を**有効ビット数**と呼ぶ。ある数 x の有効ビット数を＃(x)で表す。ある２進数の中で有効ビットがある最高のビットを**最上位ビット**と呼ぶ。ある２進数の中で有効ビットがある最低のビットを**最下位ビット**と呼ぶ。また、最上位ビットと最下位ビットの間の長さを**ビット長、ビット距離**と呼ぶ。

例）10010 の有効ビット数は２（#(10010)=2）、最上位ビットは 4、最下位ビットは 1、ビット長は 3 である。（ビットは 0 番から数えるものとする。）

３．有効ビット表示：
２進数を有効ビットの桁を列挙して

{a1, a2, a3 …}(a1, a2, a3 …は有効ビット)

と表記したものを、**有効ビット表示**と呼ぶ。また、有効ビット表示の加算・乗算は以下のように定義できる。

加算：{a1, a2, a3 …}+{ b1, b2, b3 …}= {a1, a2, a3 …,b1, b2, b3 …}
乗算：{a1, a2, a3 …}*{ b1, b2, b3 …}={ a1+ b1, a1+ b2, a1+ b3, …, a2+ b1, a2+ b2, a2+ b3…, a3+ b1, a3+ b2, a3+ b3…}

加算や乗算の結果得られた要素を低い数から列挙した表示{c1,c2,…c_i,c_i+1,…}の中

で、c_i = c_i+1 なる要素があった場合、**繰り上がり**を以下のように定義する。

c_i,c_i+1→（c_i）+1

また、（c_i）+1= c_i+2 の場合は再び繰り上がりを計算し、（c_i）+1, c_i+2→（c_i）+2 となる。このように同じ要素がある場合は同じ要素が無くなるまで繰り上がりの計算を行う。

例）101+11→{0,2}+{0,1}={0,0,1,2}={1,1,2}={2,2}={3}=1000
101*111→{0,2}+{0,1,2}={0+0,0+1,0+2,2+0,2+1,2+2}={0,1,2,2,3,4}={0,1,3,3,4}={0,1,4,4}={0,1,5}→100011

４．距離表示：
有効ビット表示の２進数の全ての要素から最下位ビットを引いたものを、各有効ビットの最下位ビットからの距離を列挙したものとして**距離表示**と呼び、以下のように表記する。

[b1,b2,b3…]

（２進数{a1, a2, a3 …}から、b1=a1-a1(=0), b2=a2-a1, b3=a3-b3…)

距離表示の中に（）や記号が用いられた場合、続く数字は()内の最上位ビットからの距離とする。また、()や記号の後ろに 0 が来る場合、その 0 は無視する。
()内の数が n 回繰り返すような場合には$()^n$と示す。$()^0$は 0 とする。特に$(0,2)^n$はφまたは繰り返し回数を明示して$φ^n$と書いて**零部と**呼ぶ（後で定義する）。

例）10111→[0,1,2,4]→[(0,1,2),2]
101010101→[0,2,4,6,8]→[$(0,2)^4$]→[$φ^{(4)}$]
10101011→[0,1,3,5,7]→[(0,1)$(0,2)^3$]→[(0,1)　$φ^{(3)}$]

また、要素表示と同様に、距離表示の乗算を以下のように定義する。

乗算 : [a1, a2, a3 …]*[b1, b2, b3 …]=[a1+ b1, a1+ b2, a1+ b3, …, a2+ b1, a2+ b2, a2+ b3…, a3+ b1, a3+ b2, a3+ b3…]

繰り上がりも要素表示と同様に定義する。

また、<u>最下位ビットが等しい２つの２進数</u>に対しては加算が定義でき、
[a1, a2, a3 …]+[b1, b2, b3 …]= [a1, a2, a3 …,b1, b2, b3 …]
となる。

例）10100→{2,4}→[0,2]
10100*110=1111000→[0,2]*[0,1]=[0,1,2,3]

なお、距離表示では元の数の最下位ビット以下の情報（0 が何個あるか）が失われるため、距離表示から元の数に復元することはできない。

５．距離表示要素のパラメータ表示
距離表示の要素の中で、パラメータによって表示できる要素はパラメータを用いて

$$[f(i)]_{i=a}^{b}$$

と表記する。上記で f(i)は i の関数で i=a~b の整数を取る。
特に、

例）[0,2,4,6,8,9]=[$[2i]_{i=0}^{4}$, 9]

６．集合要素のパラメータ表示
ある集合の要素の中で、パラメータによって表示できる要素はパラメータを用いて

$$\{f(i)\}_{i=a}^{b}$$

と表記する。上記で f(i)は i の関数で i=a~b の整数を取る。

２．コラッツ予想とは

コラッツ予想とは次のような予想である。

> コラッツ予想：
>
> 以下の２つの操作を考える。
>
> １．n が偶数の場合、n を２で割る。
> ２．n が奇数の場合、n に３をかけて１を足す
>
> 上記の２つの操作を繰り返すと、どんな正整数 n も有限回の操作のうちに必ず１となる。

例として３の場合を考えると、
3→10→5→16→8→4→2→1
となる。

これを２進数表記で表すと次のようになる。
11→1010→101→10000→1000→100→10→1

コラッツ予想は２進数で考えると次のような表現でも良い

> コラッツ予想（２進数表現）：
>
> 以下の２つの操作を考える。
>
> １．n の最下位ビットが０の場合、n を１ビット右シフトする。
> ２．n の最下位ビットが１の場合、n に 11 をかけて１を足す。
>
> 上記の２つの操作を繰り返すと、どんな正整数 n も有限回の操作のうちに必ず１となる。

３．コラッツ予想の距離表示による再定義

３を２進数表記した際のコラッツ予想の操作において、10000→1000→100→10→1 という列が表れた。これは 2^n, つまり２進数表記で 100…00 と有効ビットが１つだけになった場合、必ず１となることを示している。つまりコラッツ予想のゴールは１であるが、任意の 2^n としても良い。

また、２つの操作のうち「１. n の最下位ビットが 0 の場合、n を 1 ビット右シフトする。」という操作は、実は不要である。右シフトをしなくとも

11→1010→100000

と操作が可能である。ただし、この場合２番目の操作を「n の最下位ビットが１の場合、n に 11 をかけて**最下位ビット**を足す」と変更してある。つまり、ゴールを１ではなく 2^n にして、右シフトの操作を無くす代わりに、「11 をかけて**最下位ビット**を足す」という操作に変えることでコラッツ予想の２進数での再定義が可能である。

<table><tr><td>
コラッツ予想（２進数表現の再定義）:

以下の１つの操作を考える。

１．n に 11 をかけて最下位ビットを足す。

上記の１つの操作を繰り返すと、どんな正整数 n も有限回の操作のうちに必ず 2^n となる。
</td></tr></table>

このようにシンプルになるが、距離表示を使うとよりシンプルに記述できる。例えばゴールとなる 2^n は、距離表示では単に[0]と書ける。また、最下位ビットを足すことは+[0]と書ける。

（距離表示は２進数の 2^n 倍（100…000 倍）を同値類としてまとめた表記と言える。）

よって、距離表示によるコラッツ予想の再定義は以下の通りとなる。

<table><tr><td>
コラッツ予想（距離表示による再定義）:

以下の１つの操作を考える。

１．n に[0, 1]をかけて[0]を足す。（以下コラッツ操作： $\boldsymbol{\rho}$ と呼ぶ。）

上記の１つの操作を繰り返すと、どんな正整数 n も有限回の操作のうちに必ず[0]となる。
</td></tr></table>

距離表示による 3(=11→[0,1])のコラッツ操作を再び見てみると、

[0,1]→[0,2]→[0]

と、2 回の操作でゴールである[0]にたどり着いている。また、[0,1]は 11×100…000 の形で書ける同値類を表すため、11000…000 の形で書けるどんな数も 2 階の操作でゴールである 1×100…000 の同値類となることを表している。

４．コラッツ操作による集合

今、コラッツ予想のゴールとなる要素全体の集合を C_0 とすると、

$$C_0=\{[0]\}$$

である。次に、コラッツ操作「[0,1]をかけて[0]を足す。」を ρ で表し、

$$\rho(x)=[0](\in C_0)$$

となる x 全体の集合を C_1 とする。以下、コラッツ操作によって、

$$C\_(k+1)=\{x\mid \rho\,(x)\in C\_k\}$$

と定義する。この時の集合の添え字は、その集合の要素が何回のコラッツ操作によってゴールにたどり着くかを示しており、

$$\forall x\in C_k, \rho^k x=[0]$$

である。C_k の事を、**k 階のコラッツ集合**と呼ぶことにする。また C_0～C_∞を合わせて**コラッツ集合**と呼ぶことにする。

コラッツ予想は

全ての正整数はいずれかの C_k に属する

という予想である。

また、

$$i\neq j, \forall x\in C_i \rightarrow x\notin C_j \text{（ただし[0]を除く）}$$

である。（要証明）

例えば 3 に対するコラッツ操作を考えると、[0,1]→[0,2]→[0]であったため、

$$[0,1]\in C_2, [0,2]\in C_1$$

である。

５．メルセンヌ数によって生成されるコラッツ集合の要素の例

3 に対するコラッツ操作では$[0,1] \in C_2, [0,2] \in C_1$という 2 つのコラッツ集合の要素が得られた。ある数に対し、k 回のコラッツ操作によって[0]となれば、その数は C_k の要素であり、k 回のコラッツ操作で C_0~C_{k-1} までの集合の要素が得られることとなる。

ここではいくつかのメルセンヌ数をスタートとしてコラッツ操作を[0]まで行い、コラッツ集合の要素がどのように得られるか見てみる。

なお、メルセンヌ数を取るのは、有効ビット数が多いため、比較的多くのコラッツ集合の要素が得られると期待されるからである。（コラッツ操作の繰り返しで[0]にすることは、繰り上がりによって有効ビット数を 1 つにまで減らす必要がある。有効ビット数が多い程有効ビット数を減らすのは一般に困難と考えられる。（例外はあるが。））

以下にメルセンヌ数によって生成されるコラッツ集合の要素を示す。

M_3(=111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]
C 5 [0, 1, 2]

M_4(=1111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 4, 5]
C 3 [0, 1, 5]
C 4 [0, 1, 2, 4]
C 5 [0, 1, 2, 3]

M_5(=11111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 4, 5]
C 3 [0, 1, 5]
C 4 [0, 1, 2, 4]
C 5 [0, 2, 3, 4, 5]
C 6 [0, 2, 6, 8]

C 7 [0, 4, 5, 7, 8]
C 8 [0, 6, 9]
C 9 [0, 2, 10, 11]
C 10 [0, 1, 11]
C 11 [0, 1, 2, 4, 6, 8, 10]
C 12 [0, 1, 2, 3, 7, 8, 9]
C 13 [0, 2, 3, 4, 5, 6, 8, 11]
C 14 [0, 1, 4, 6, 9, 10]
C 15 [0, 1, 2, 4, 5, 10]
C 16 [0, 1, 2, 3, 6, 7, 9]
C 17 [0, 1, 2, 3, 4, 6, 7, 8]
C 18 [0, 1, 2, 3, 4, 5, 8]
C 19 [0, 3, 5, 7, 8]
C 20 [0, 1, 3, 4, 8]
C 21 [0, 3, 4, 5, 6, 8]
C 22 [0, 1, 3, 4, 5, 6, 7]
C 23 [0, 1, 2, 5, 7]
C 24 [0, 2, 3, 4, 5, 7, 8]
C 25 [0, 4, 6, 9]
C 26 [0, 1, 3, 7, 8]
C 27 [0, 1, 2, 8]
C 28 [0, 1, 2, 3, 5, 7]
C 29 [0, 3, 5, 6, 7]
C 30 [0, 1, 3, 4, 7]
C 31 [0, 1, 2, 5, 6]
C 32 [0, 3, 7]
C 33 [0, 1, 3, 4, 6]
C 34 [0, 3, 4, 5, 6]
C 35 [0, 5, 7]
C 36 [0, 1, 3, 5, 6]
C 37 [0, 1, 2, 6]
C 38 [0, 1, 2, 3, 5]
C 39 [0, 1, 2, 3, 4]

M_6(=111111)によって生成されるコラッツ集合

C 0 [0]

C 1 [0, 2]

C 2 [0, 2, 4, 5]

C 3 [0, 1, 5]

C 4 [0, 1, 2, 4]

C 5 [0, 2, 3, 4, 5]

C 6 [0, 2, 6, 8]

C 7 [0, 4, 5, 7, 8]

C 8 [0, 6, 9]

C 9 [0, 2, 10, 11]

C 10 [0, 1, 11]

C 11 [0, 1, 2, 4, 6, 8, 10]

C 12 [0, 1, 2, 3, 7, 8, 9]

C 13 [0, 2, 3, 4, 5, 6, 8, 11]

C 14 [0, 1, 4, 6, 9, 10]

C 15 [0, 1, 2, 4, 5, 10]

C 16 [0, 1, 2, 3, 6, 7, 9]

C 17 [0, 1, 2, 3, 4, 6, 7, 8]

C 18 [0, 1, 2, 3, 4, 5, 8]

C 19 [0, 3, 5, 7, 8]

C 20 [0, 1, 3, 4, 8]

C 21 [0, 3, 4, 5, 6, 8]

C 22 [0, 1, 3, 4, 5, 6, 7]

C 23 [0, 1, 2, 5, 7]

C 24 [0, 2, 3, 4, 5, 7, 8]

C 25 [0, 4, 6, 9]

C 26 [0, 1, 3, 7, 8]

C 27 [0, 1, 2, 8]

C 28 [0, 1, 2, 3, 5, 7]

C 29 [0, 3, 5, 6, 7]

C 30 [0, 1, 3, 4, 7]

C 31 [0, 1, 2, 5, 6]

C 32 [0, 3, 7]

C 33 [0, 1, 3, 4, 6]

C 34 [0, 2, 5, 6, 7, 8]
C 35 [0, 1, 6, 8]
C 36 [0, 1, 2, 4, 6, 7]
C 37 [0, 1, 2, 3, 7]
C 38 [0, 1, 2, 3, 4, 6]
C 39 [0, 1, 2, 3, 4, 5]

M_7(=1111111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]
C 5 [0, 2, 3, 4]
C 6 [0, 2, 3, 6]
C 7 [0, 2, 3, 6, 7]
C 8 [0, 2, 6, 10]
C 9 [0, 4, 5, 7, 8, 10]
C 10 [0, 1, 3, 6, 7, 8, 9]
C 11 [0, 1, 2, 7, 9]
C 12 [0, 1, 2, 3, 5, 7, 8]
C 13 [0, 1, 2, 3, 4, 8]
C 14 [0, 1, 2, 3, 4, 5, 7]
C 15 [0, 1, 2, 3, 4, 5, 6]

M_8(=11111111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]
C 5 [0, 2, 3, 4]
C 6 [0, 2, 3, 6]
C 7 [0, 2, 3, 6, 7]
C 8 [0, 2, 4, 8, 12]
C 9 [0, 1, 5, 6, 8, 9, 11]

C 10 [0, 1, 2, 4, 7, 8, 9, 10]
C 11 [0, 1, 2, 3, 8, 10]
C 12 [0, 1, 2, 3, 4, 6, 8, 9]
C 13 [0, 1, 2, 3, 4, 5, 9]
C 14 [0, 1, 2, 3, 4, 5, 6, 8]
C 15 [0, 1, 2, 3, 4, 5, 6, 7]

M_9(=111111111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]
C 5 [0, 1, 2]
C 6 [0, 2, 5]
C 7 [0, 4, 5]
C 8 [0, 6]
C 9 [0, 2, 3, 5, 7]
C 10 [0, 2, 4, 6, 7, 10, 11, 12]
C 11 [0, 4, 5, 6, 9, 10, 13]
C 12 [0, 6, 8, 9, 12, 13]
C 13 [0, 1, 3, 5, 9, 13]
C 14 [0, 1, 2, 6, 7, 9, 10, 12]
C 15 [0, 1, 2, 3, 5, 8, 9, 10, 11]
C 16 [0, 1, 2, 3, 4, 9, 11]
C 17 [0, 1, 2, 3, 4, 5, 7, 9, 10]
C 18 [0, 1, 2, 3, 4, 5, 6, 10]
C 19 [0, 1, 2, 3, 4, 5, 6, 7, 9]
C 20 [0, 1, 2, 3, 4, 5, 6, 7, 8]

M_{10}(=1111111111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]

C 5 [0, 1, 2]
C 6 [0, 2, 5]
C 7 [0, 4, 5]
C 8 [0, 6]
C 9 [0, 2, 3, 5, 7]
C 10 [0, 2, 4, 6, 7, 10, 11, 12]
C 11 [0, 2, 6, 7, 8, 11, 12, 15]
C 12 [0, 1, 7, 9, 10, 13, 14]
C 13 [0, 1, 2, 4, 6, 10, 14]
C 14 [0, 1, 2, 3, 7, 8, 10, 11, 13]
C 15 [0, 1, 2, 3, 4, 6, 9, 10, 11, 12]
C 16 [0, 1, 2, 3, 4, 5, 10, 12]
C 17 [0, 1, 2, 3, 4, 5, 6, 8, 10, 11]
C 18 [0, 1, 2, 3, 4, 5, 6, 7, 11]
C 19 [0, 1, 2, 3, 4, 5, 6, 7, 8, 10]
C 20 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

M_{10}(=11111111111)によって生成されるコラッツ集合
C 0 [0]
C 1 [0, 2]
C 2 [0, 2, 3]
C 3 [0, 4]
C 4 [0, 1, 3]
C 5 [0, 2, 3, 4]
C 6 [0, 2, 3, 6]
C 7 [0, 2, 3, 6, 7]
C 8 [0, 2, 6, 10]
C 9 [0, 4, 5, 7, 8, 10]
C 10 [0, 1, 3, 6, 7, 8, 9]
C 11 [0, 1, 2, 7, 9]
C 12 [0, 1, 2, 3, 5, 7, 8]
C 13 [0, 1, 2, 3, 4, 8]
C 14 [0, 1, 2, 3, 4, 5, 7]
C 15 [0, 1, 2, 3, 4, 5, 6]
C 16 [0, 3, 5, 7]
C 17 [0, 2, 7, 8, 9]

C 18 [0, 4, 5, 7, 10]
C 19 [0, 6, 9, 10]
C 20 [0, 1, 3, 5, 10]
C 21 [0, 2, 3, 4, 8, 9, 11]
C 22 [0, 4, 6, 7, 9, 10, 11]
C 23 [0, 6, 7, 8, 9, 12]
C 24 [0, 1, 3, 5, 8, 10, 11]
C 25 [0, 2, 4, 5, 6, 10, 11, 15]
C 26 [0, 1, 5, 7, 8, 10, 11, 12, 14]
C 27 [0, 3, 4, 6, 7, 10, 11, 12, 13, 14]
C 28 [0, 1, 3, 4, 5, 8, 9, 12, 14]
C 29 [0, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14]
C 30 [0, 2, 3, 5, 6, 7, 8, 9, 10, 13, 16]
C 31 [0, 4, 7, 9, 11, 15, 16]
C 32 [0, 1, 3, 8, 9, 10, 16]
C 33 [0, 3, 4, 5, 7, 9, 10, 11, 12, 14, 16]
C 34 [0, 5, 7, 8, 9, 12, 14, 15, 16]
C 35 [0, 7, 8, 9, 10, 11, 13, 14, 17]
C 36 [0, 2, 3, 5, 7, 9, 10, 11, 12, 13, 14, 17, 18]
C 37 [0, 1, 4, 5, 6, 10, 12, 14, 18]
C 38 [0, 1, 2, 4, 5, 6, 7, 9, 13, 14, 15, 17]
C 39 [0, 3, 6, 8, 11, 12, 14, 15, 16, 17]
C 40 [0, 1, 3, 4, 6, 7, 12, 15, 17]
C 41 [0, 1, 2, 5, 6, 7, 8, 10, 12, 13, 15, 16]
C 42 [0, 1, 2, 3, 5, 6, 7, 8, 9, 13, 16]
C 43 [0, 1, 2, 3, 4, 7, 9, 14, 15]
C 44 [0, 1, 2, 3, 4, 5, 7, 8, 15]
C 45 [0, 2, 3, 4, 5, 6, 7, 8, 11, 12, 14, 16]
C 46 [0, 4, 6, 8, 10, 11, 14, 15, 16]
C 47 [0, 1, 3, 7, 8, 9, 12, 13, 16]
C 48 [0, 1, 2, 8, 10, 11, 14, 15]
C 49 [0, 1, 2, 3, 5, 7, 11, 15]
C 50 [0, 1, 2, 3, 4, 8, 9, 11, 12, 14]
C 51 [0, 1, 2, 3, 4, 5, 7, 10, 11, 12, 13]
C 52 [0, 1, 2, 3, 4, 5, 6, 11, 13]
C 53 [0, 1, 2, 3, 4, 5, 6, 7, 9, 11, 12]

C 54 [0, 1, 2, 3, 4, 5, 6, 7, 8, 12]
C 55 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11]
C 56 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

６．コラッツ逆操作によって生成されるコラッツ集合の要素の例

メルセンヌ数からスタートし、[0]にいたるまでコラッツ操作ρを繰り返して生成されるコラッツ集合の要素の例を見た。ここでは逆に、[0]からコラッツ操作の逆操作（**コラッツ逆操作：** ρ^{-1}）を考え、コラッツ逆操作によって生成されるコラッツ集合の要素を考える。

コラッツ逆操作は以下のように定義できる。

コラッツ逆操作
１．n から n の最下位ビット以下の任意のビットを引く。引いた数を m とする。
２．m が 3({0,1})で割り切れる場合,3({0,1})で割る。

ここで「１． n から n の最下位ビット以下の任意のビットを引く」には注意が必要である。これはコラッツ操作ρの「[0]を足す」の逆操作に対応するのだが、例えば[0,1]という数は 11×2^n の数を全て表しているので、11000 かもしれないし 110000000 かもしれない。これらの数が「[0]を足す」の結果できたとした場合、「[0]を足す」前の数は 11000 の場合は 10111 であり、110000000 の場合は 101111111 である。つまり、ある距離表示のコラッツ集合の要素にコラッツ操作を行って生成される要素は 1 つであったが、コラッツ逆操作によって生成される要素は 1 つではない。

また、「２．m が 3({0,1})で割り切れる場合,3({0,1})で割る」に関しても、コラッツ操作ρの場合はどんな数に対しても 3 をかける事ができたが、3 で割ることは 3 の倍数にしかできないことが異なる。

コラッツ集合の定義から、あるコラッツ集合 C_k の要素にコラッツ逆操作を行って生成された要素は、コラッツ集合 C_(k+1)の要素である。

$$C\_(k+1)=\{y \mid \rho^{-1}(x), x\in C\_k\}$$

実際にコラッツ逆操作によって生成される要素の例を見ていく。コラッツ逆操作をする際には距離表示に対して、最下位ビットの後ろに 0 の個数を 1 から 100 まで増やした場合の数に対して逆操作を行い要素の探索を行った。

C_1 の要素：

まず、[0]にコラッツ逆操作を行うと、C_1 の要素として、{[0],[0,2],[0,2,4],[0,2,4,6]…}を得る。メルセンヌ数からコラッツ操作を行った際に得られた C_1 の要素は[0,2]のみであったが、一般に$[2i]_{i=0}^{n}$の形の要素を持つことがわかった。C_1 は

$$C_1 = \{[2i]_{i=0}^{n}\}_{n=0}^{\infty}$$

と書くことができる。$\phi := [2i]_{i=0}^{n}$と定義し、コラッツ操作ρによって[0]になることから、**零部**と呼ぶ事にする。($\phi = (0,2)^n$と同じである。)

$$\rho^{-1}([0]) = \{\phi\}_{n=0}^{\infty} = C_1$$

$$\rho(\phi) = [0] = C_0$$

また、

$$[0] \in \phi$$

である。

C_2の要素：

　次にC_1の要素、つまりϕのコラッツ逆操作によって生成されるC_2の要素を見る。ϕはコラッツ操作により[0]のただ一つの距離表示の要素になるが、コラッツ逆操作では複数の要素が生成されるため、1つ1つ見ていく必要がある。

　まず、[0,2]のコラッツ逆操作によって生成されるC_2の要素は、

{[0,1],[0,2,3],[0,2,4,5],[0,2,4,6,7]…}=$\{[\phi^n, 1]\}_{n=0}^{\infty}$

となる。ここで、以下の通り記法を簡略化する。

$$\{[\phi^n, 1]\}_{n=0}^{\infty} \to \{[\phi^n, 1]\}$$

　以下、C1の別の要素によって生成されるC_2の要素は、

$$\rho^{-1}([0,2,4]) = \emptyset$$

$$\rho^{-1}([0,2,4,6]) = \{[\phi^n, 4,5,6]\}$$

$$\rho^{-1}([0,2,4,6,8]) = \{[\phi^n, 1,5,6,7]\} = \{[\phi^n, (1), (4,5,6)]\}$$

$$\rho^{-1}([0,2,4,6,8,10]) = \emptyset$$

$$\rho^{-1}([0,2,4,6,8,10,12]) = \{[\phi^n, 4,5,6,10,11,12]\} = \{[\phi^n, (4,5,6)^2]\}$$

$$\rho^{-1}([0,2,4,6,8,10,12,14]) = \{[\phi^n, 1,5,6,7,11,12,13]\} = \{[\phi^n, (1), (4,5,6)^2]\}$$

$$\rho^{-1}([0,2,4,6,8,10,12,14,16]) = \emptyset$$

$$\cdots$$

結局、C_2の要素は、ρによって移る先のC_1の要素の有効ビットによって以下の3つのパターンに分かれる。(C_1の要素はϕのみである。)

C_2の要素

$$\emptyset：(\#\phi(\in C_1)\ \mathrm{mod}3=0)$$
$$[\phi^n,(4,5,6)^m](\mathrm{m}=1\sim\infty)：(\#\phi(\in C_1)\ \mathrm{mod}3=1)$$
$$[\phi^n,(1),(4,5,6)^m](\mathrm{m}=0\sim\infty)：(\#\phi(\in C_1)\ \mathrm{mod}3=2)$$

ここで零部（ϕ^n）以外の部分を**主要部**と呼ぶことにする。

C_3の要素：

C_3の要素を考える際には以下のことに留意する必要がある。

C_1の要素はϕのみであったが、ϕ(=ϕ^n)は$(0, 2)^n$で書け、n 回の繰り返しがあり、C_2の主要部$(4,5,6)^m$も m 回の繰り返しを持つ。

C_2の要素は零部ϕと主要部である$(4, 5, 6)^m$または$(1)(4, 5, 6)^m$からなるため、零部と主要部それぞれに対した繰り返しがある。よって、C_3の要素も、それぞれの繰り返しに応じた繰り返しを持つ要素を有すると考えられる。

まず、C_2の主要部を固定し、零部ϕの有効ビット数を変化させた場合の対応するC_3の要素を示す。

C_3の要素　（C_2の主要部を固定した場合）

c_2（∈C）=[ϕ, 1]の場合

$$\emptyset：(\#c_2\ \mathrm{mod}3=2)$$
$$[\phi,(4,5,6)^n,(4)](\mathrm{n}=0\sim\infty)：(\#c_2\ \mathrm{mod}3=1)$$
$$[\phi,(1),(4,5,6)^n,(4)](\mathrm{n}=1\sim\infty)：(\#c_2\ \mathrm{mod}3=0)$$

c_2（∈C）=[ϕ, 4, 5, 6]の場合

$$\emptyset：(\#c_2\ \mathrm{mod}3=2)$$
$$[\phi,(1),(4,5,6)^n,(2,5)](\mathrm{n}=0\sim\infty)：(\#c_2\ \mathrm{mod}3=1)$$
$$[\phi,(4,5,6)^n,(2,5)](\mathrm{n}=1\sim\infty)：(\#c_2\ \mathrm{mod}3=0)$$

上記で(4,5,6)の部分はρによってc_2の零部に移る部分であり、**第 1 主要部**と呼ぶことにする。また、(4), (2,5)の部分はρによってc_2の主要部に移る部分であり、**第 2 主要部**と呼ぶことにする。

c_3の第一主要部はc_2の主要部と一致している。

第 2 主要部は c_2 の主要部によって変化し、c_2 が(1)を有する場合は(4)となり、c2 が (4,5,6)を有する場合は(2,5)となっている

次に、c_2 の主要部の繰り返しによって c_3 がどのように変化するかを見る。

C_3 の要素　（C_2 の零部を 0 に固定した場合）

c_2（∈C）=[0,（4, 5, 6)^m]の場合

$$\emptyset：(\mathrm{m\ mod3}=2)$$
$$[\phi,(1),(2,5),(5,7,8,10,11,12,13,15,18)^n](\mathrm{n}=0\sim\infty)：(\mathrm{m\ mod3}=1)$$
$$[\phi,(1),(5,7,8,10,11,12,13,15,18)^n](\mathrm{n}=1\sim\infty)：(\mathrm{m\ mod3}=0)$$

c_2（∈C）=[0,（1),（4, 5, 6)^m]の場合

$$\emptyset：(\mathrm{m\ mod3}=0)$$
$$[\phi,(1,2,4,7),(5,7,8,10,11,12,13,15,18)^n](\mathrm{n}=0\sim\infty)：(\mathrm{m\ mod3}=1)$$
$$[\phi,(3,4,6,7,8,9,11,14),(5,7,8,10,11,12,13,15,18)^n](\mathrm{n}=0\sim\infty)：(\mathrm{m\ mod3}=2)$$

C_2 の零部を 0 としたため、第 1 主要部である(4, 5, 6)の部分は表れていない。第 2 主要部の中の繰り返し部分である(5, 7, 8, 10, 11, 12, 13, 15, 18)は、ρ によって C_2 の主要部$(4, 5, 6)^3$に移る部分である。

７．C_2の要素に関する補足

C_2の要素は、対応するC_1の要素（c1）の有効ビット数によって 3 つに分かれたが、このことについて補足する。

まず、c1 mod3=1 の場合を考える。まずこのような要素の代表として[0]を考える。[0]は 2 進数表記で 1, 10, 100,…などであり､それぞれ 1 を引いて 3 で割ることができるか見ると、

1−1=0(十進数で 0)，0÷3=0（〇）
10−1=1(十進数で 1)，1÷3（×）
100−1=11(十進数で 3)，3÷3=1（〇）
1000−1=111(十進数で 7)，7÷3（×）
…

というように、1 つ飛ばしで 3 で割れる場合と割れない場合がある。これは 1 を引いた際の数がメルセンヌ数になっており。111…111 となっているが、この 1 の数が偶数個であれば 11(=3)で割り切れるためである。(p が素数の時、Mp は 1 と自身以外のメルセンヌ数を約数に持たない。)

c1 mod3=1 は、[0,(2,4,6)^n]であるが、(2,4,6)の部分は 2 進数では 101010×x であり、101010=42=3×14 であるから 3 で割り切れる。よって c1 mod3=1 の場合、[0]の部分から 1 を引いた時の有効ビット数が偶数であれば 3 で割り切ることができる。

c1 mod3=2 の場合も同様に、まず[0,2]の部分を見ると

101−1=100(十進数で 4)，4÷3（×）
1010−1=1001(十進数で 9)，9÷3=3（〇）
10100−1=10011(十進数で 19)，19÷3（×）
101000−1=100111(十進数で 39)，39÷3=13（〇）
…

となるが、1 を引いた後の数は 1001 の後に 111…111 となっている。1001 は 9 であり 3 で割れるから、続く 111...111 の部分の有効ビット数が偶数個なら 3 で割り切ることができることになる。

最後に c1 mod3=0 の場合は、まず[0,2,4]の部分を見ると

10101−1=10100(十進数で 20)，20÷3（×）
101010−1=101001(十進数で 41)，41÷3（×）
1010100−1=1010011（十進数で 83)，83÷3（×）

10101000-1=10100111（十進数で 167），167÷3（×）

となり、10100 の後に 111…111 と続く形になるが、10100 は 20 であり 3 で割れないため、続く 111…111 の有効ビット数に関わらず割ることができない。よって c1 mod3=0 の場合 C_2 の要素は∅となっている。(コラッツ操作によってそのような c1 となる C_2 の要素は存在しない)

８．微分・積分とのアナロジー

C1~C3 までの要素がどのような形をしているか見た結果、C1 は零部、C2 は零部と（第一）主要部、C3 は零部と第一主要部と第二主要部からなることが分かった。また、コラッツ操作によって、零部は[0]に、第一主要部は零部に、第二主要部は第一主要部に移ることが分かった。このことから C4 は、少なくとも第三主要部を持ち、第二主要部、第一主要部および零部を任意に含む要素からなることが予想され、C5 以降も同様と考えられる。

こうしてみると、コラッツ操作は多項式の微分と同様な働きをすると考えることができる。例として、

$$x^3 + x^2 + x + b$$

という多項式を考えると、この式は微分によって

$$x^3 + x^2 + x + b \to 3x^2 + 2x + 1 \to 6x + 2 \to 6 \to 0$$

と、3 回の微分で 0 となるが、定数項を零部、x の 1 乗の項を第一主要部、x の 2 乗の項を第二主要部……というように考えると、3 階のコラッツ集合が 3 回のコラッツ操作によって[0]となることと同様にみなすことができる。また、定数項はどれだけ大きな値でも微分すれば 0 になる点と零部の繰り返し回数がどれだけ多くてもコラッツ操作によって[0]になる点は似ている。x^3 を微分すると $3x^2$ というように次数が係数にかかってくる点と第 2 主要部の$(5, 7, 8, 10, 11, 12, 13, 15, 18)$がコラッツ操作によって$(4, 5, 6)^3$となるように、第 1 主要部の繰り返し回数にかかってくる点も似ている。

９．1~100 までの数のコラッツ集合の階数

以下に十進数で 0~100 までの数が、何階のコラッツ集合の要素となるかを示す。

十進数	二進数	階数	距離表示
1	1	C 0	[0]
2	10	C 0	[0]
3	11	C 2	[0, 1]
4	100	C 0	[0]
5	101	C 1	[0, 2]
6	110	C 2	[0, 1]
7	111	C 5	[0, 1, 2]
8	1000	C 0	[0]
9	1001	C 6	[0, 3]
10	1010	C 1	[0, 2]
11	1011	C 4	[0, 1, 3]
12	1100	C 2	[0, 1]
13	1101	C 2	[0, 2, 3]
14	1110	C 5	[0, 1, 2]
15	1111	C 5	[0, 1, 2, 3]
16	10000	C 0	[0]
17	10001	C 3	[0, 4]
18	10010	C 6	[0, 3]
19	10011	C 6	[0, 1, 4]
20	10100	C 1	[0, 2]
21	10101	C 1	[0, 2, 4]
22	10110	C 4	[0, 1, 3]
23	10111	C 4	[0, 1, 2, 4]
24	11000	C 2	[0, 1]
25	11001	C 7	[0, 3, 4]

十進数	二進数	階数	距離表示
26	11010	C 2	[0, 2, 3]
27	11011	C 41	[0, 1, 3, 4]
28	11100	C 5	[0, 1, 2]
29	11101	C 5	[0, 2, 3, 4]
30	11110	C 5	[0, 1, 2, 3]
31	11111	C 39	[0, 1, 2, 3, 4]
32	100000	C 0	[0]
33	100001	C 8	[0, 5]
34	100010	C 3	[0, 4]
35	100011	C 3	[0, 1, 5]
36	100100	C 6	[0, 3]
37	100101	C 6	[0, 2, 5]
38	100110	C 6	[0, 1, 4]
39	100111	C 11	[0, 1, 2, 5]
40	101000	C 1	[0, 2]
41	101001	C 40	[0, 3, 5]
42	101010	C 1	[0, 2, 4]
43	101011	C 9	[0, 1, 3, 5]
44	101100	C 4	[0, 1, 3]
45	101101	C 4	[0, 2, 3, 5]
46	101110	C 4	[0, 1, 2, 4]
47	101111	C 38	[0, 1, 2, 3, 5]
48	110000	C 2	[0, 1]
49	110001	C 7	[0, 4, 5]
50	110010	C 7	[0, 3, 4]

十進数	二進数	階数	距離表示
51	110011	C 7	[0, 1, 4, 5]
52	110100	C 2	[0, 2, 3]
53	110101	C 2	[0, 2, 4, 5]
54	110110	C 41	[0, 1, 3, 4]
55	110111	C 41	[0, 1, 2, 4, 5]
56	111000	C 5	[0, 1, 2]
57	111001	C 10	[0, 3, 4, 5]
58	111010	C 5	[0, 2, 3, 4]
59	111011	C 10	[0, 1, 3, 4, 5]
60	111100	C 5	[0, 1, 2, 3]
61	111101	C 5	[0, 2, 3, 4, 5]
62	111110	C 39	[0, 1, 2, 3, 4]
63	111111	C 39	[0, 1, 2, 3, 4, 5]
64	1000000	C 0	[0]
65	1000001	C 8	[0, 6]
66	1000010	C 8	[0, 5]
67	1000011	C 8	[0, 1, 6]
68	1000100	C 3	[0, 4]
69	1000101	C 3	[0, 2, 6]
70	1000110	C 3	[0, 1, 5]
71	1000111	C 37	[0, 1, 2, 6]
72	1001000	C 6	[0, 3]
73	1001001	C 42	[0, 3, 6]
74	1001010	C 6	[0, 2, 5]
75	1001011	C 3	[0, 1, 3, 6]

十進数	二進数	階数	距離表示
76	1001100	C 6	[0, 1, 4]
77	1001101	C 6	[0, 2, 3, 6]
78	1001110	C 11	[0, 1, 2, 5]
79	1001111	C 11	[0, 1, 2, 3, 6]
80	1010000	C 1	[0, 2]
81	1010001	C 6	[0, 4, 6]
82	1010010	C 40	[0, 3, 5]
83	1010011	C 40	[0, 1, 4, 6]
84	1010100	C 1	[0, 2, 4]
85	1010101	C 1	[0, 2, 4, 6]
86	1010110	C 9	[0, 1, 3, 5]
87	1010111	C 9	[0, 1, 2, 4, 6]
88	1011000	C 4	[0, 1, 3]
89	1011001	C 9	[0, 3, 4, 6]
90	1011010	C 4	[0, 2, 3, 5]
91	1011011	C 33	[0, 1, 3, 4, 6]
92	1011100	C 4	[0, 1, 2, 4]
93	1011101	C 4	[0, 2, 3, 4, 6]
94	1011110	C 38	[0, 1, 2, 3, 5]
95	1011111	C 38	[0, 1, 2, 3, 4, 6]
96	1100000	C 2	[0, 1]
97	1100001	C 43	[0, 5, 6]
98	1100010	C 7	[0, 4, 5]
99	1100011	C 7	[0, 1, 5, 6]
100	1100100	C 7	[0, 3, 4]

１０．　今後の展望

コラッツ予想を解くには、全ての奇数の数とコラッツ集合の要素の数が一致することを示せばよい。これには例えばn桁の2進数で表すことのできるコラッツ集合の要素というものを考えると良さそうである。

例えば2桁の2進数は01, 10, 11の3種類であるが、C_1の要素は最小で101と3桁必要であるため含まれず、01, 10は共にC_0の要素で、11となるのはC2の要素の1つだけである。このようにある桁数に入るコラッツ集合の数は制限されるため、その数が自然数と1対1に対応していることが示せればよい。

これはコラッツ集合の階数が低い間は容易そうに見える。例えばn桁の2進数で表すことのできる自然数の総数は2^nであり、その間に含まれるC_0の要素はn個で、C_1の要素はn/2-0.5（切り捨て）個である。ただし、コラッツ集合の階数が高くなると容易には判断がつきそうにない。例えば5桁の2進数の時点でC_{41}やC_{39}といった高階のコラッツ集合の要素が顔を出しているがC_8～C_{38}の要素は未だ出てきていない。C_{41}やC_{39}の最小の要素が5桁の2進数で表現できるという事はそれほど自明ではないように思われる。

ある2進数を与えられた時の複雑さのような尺度を与えられると良い。例えば1000000や10101010という2進数は、どれだけ桁が多くても単純であるためC_0やC_1に属するが、11011という数は複雑であってC_{41}となることが判断できるとよい。

ただし、この複雑さの判断も難しく、例えば[0,2,4,6,7,8]という数を考えると、これはC_5であるが、[0,2,4,6,7,8]=[φ,1,2]であり、前半の[0,2,4,6]の部分は1回のコラッツ操作ですでに[0]となり、残りの[7,8]の部分が第4主要部のため5回の操作を必要としている。ある数が与えられたときに、その複雑さを表す主要部がどこであるかを判断することもそれほど容易ではない。

参考文献

[1] コラッツの問題, Wikipedia,
https://ja.wikipedia.org/wiki/%E3%82%B3%E3%83%A9%E3%83%83%E3%83%84%E3%81%AE%E5%95%8F%E9%A1%8C

[2]

来歴

2021/1/11	初稿
2021/1/31	コラッツ逆操作の項を追記
2021/2/21	C2 の要素と C3 の要素の項を追記
2021/5/3	C_2 の要素に関する補足、微分・積分とのアナロジー、1~100 までの数のコラッツ集合の階数、今後の展望を追記