コラッツ予想について

芝浦工業大学　数理科学科

※何か不明な点や計算ミス等がありましたら加筆修正しますので指摘をお願いします

平成 27 年 11 月 6 日 (金)

制作: BV13020　　久保田　一次

目 次

1 はじめに **2**
1.1 研究背景 2
1.2 コラッツ予想とは 2
1.3 今回用いる特殊な語句や記法 3

2 証明方法 **4**
2.1 第二数の存在 4
2.2 巡回列，無限大発散列の非存在 4

3 コラッツ予想と 2 進数 **4**
3.1 2 進数を用いたコラッツ操作 4
3.2 コラッツ予想と対応する動きをとるアルゴリズム 5

4 今後の課題 **6**

5 参考文献 **6**

1　はじめに

1.1　研究背景

コラッツ予想とは数論の未解決問題の一つである. アルゴリズムの基本構造は単純であり, プログラムで再現することは容易である. 実際にプログラムにして視覚的に観察した際, 多少興味深い結果が得られた. 様々なプログラムを用いてこの問題に対してアプローチを行い, コラッツ予想について考察していきたい.

1.2　コラッツ予想とは

1937 年に Lothar Collatz が提唱した問題で, 未解決問題の一つである.

任意の自然数 n に対し, 以下のどちらかの操作を行う.

- n が偶数なら 2 で割る.
- n が奇数なら 3 をかけて 1 を足す.

このとき求まった数を新しく n とし, 再度新たな n に対して同様の操作を行う.
以上の操作を繰り返すことで有限回の操作のうちに必ず 1 に到達することを示せ.

という証明問題である. 実際に 7 に対してこの操作を繰り返した結果を下記に示す.

$7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40$
$\to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$

またこのアルゴリズムを実行する C 言語のプログラムを下記に示す.

ソースコード 1: コラッツ予想

```
#include<math.h>
int main(void){
    int n;
    scanf("%d",&n);
    do{n = (n%2==0)?n/2:n*3+1;printf("%d \n",n);}while (n!=1);
    return 0;}
```

1.3 今回用いる特殊な語句や記法

これ以降登場する数は特に断りがない限り, 全て 0 を含まない自然数とする.

コラッツ操作:ある数 n に対してコラッツ予想で行う動作を 1 回行うことであり, $n \to 3n+1$ と表す. また t 回コラッツ操作を繰り返すとき特別にコラッツ t 回操作といい, $n^{(t)}$ と表す.

例)7 に対してコラッツ操作を行うと 22 $(7 \to 22)$.

7 に対してコラッツ 4 回操作を行うと 17 $(7^{(4)} = 17)$.

増加操作:明らかに奇数である数 n に対してコラッツ操作を行うことであり, $n \nearrow 3n+1$ と表す.

減少操作:明らかに偶数である数 n に対してコラッツ操作を行うことであり, $n \searrow \frac{n}{2}$ と表す.

第一数:ある数 n に対してその数がはじめて 1 にいくまでに行ったコラッツ操作の回数であり, ${}_{(1)}n = k$ と表す. ただし ${}_{(1)}1 = 0$ とする.

例)7 の第一数は 15 である (${}_{(1)}7 = 15$).

第二数：ある数 n に対してその数がはじめて元の数 n 未満にいくまでに行ったコラッツ操作の回数であり, ${}_{(2)}n = k$ と表す. ただし ${}_{(2)}1 = 0$

例)7 の第二数は 10 である (${}_{(2)}7 = 10$).

コラッツ列：ある数 n に対してその数が初めて 1 にいくまでに辿った数字の列であり, $N(n) = \{n,\ n_2,\ n_3,\ \cdots n_k\}$ と表す．ただしコラッツ操作を繰り返しても 1 に到達しない数が存在するときも, 要素数が ∞ のコラッツ列として扱う.

例)$N(7) = \{7,\ 22,\ 11,\ 34,\ 17,\ 52,\ 26,\ 13,\ 40,\ 20,\ 5,\ 16,\ 8,\ 4,\ 2,\ 1\}$

最大数：ある数 n に対するコラッツ列において, そのコラッツ列の要素の内最大の数字のことであり, $M(n) = s$ とかく.

例)7 の最大数は 52 である $(M(7) = 52)$.

無限大発散列:要素数が ∞ のコラッツ列の内, 最大数の存在しないものであり, n のコラッツ列が無限大発散列であるとき, $N_I(n)$ と表す.

巡回列:要素数が ∞ のコラッツ列の内, 全ての要素が複数出現するものであり, n のコラッツ列が巡回列であるとき, $N_R(n)$ と表す.

ただし便宜上巡回列の要素を書くとき, 初めて元の数に戻るまでの要素を書くものとする.

巡回列最大数:ある数 n の巡回列であるコラッツ列に対して, そのコラッツ列の要素の内最大の数字のことであり, $M_R(n) = s$ とかく.

巡回列最小数:ある数 n の巡回列であるコラッツ列に対して, そのコラッツ列の要素の内最小の数字のことであり, $\mathrm{m}_R(n) = s$ とかく.

巡回列周期数:ある数 n の巡回列であるコラッツ列に対して, その数がはじめて元の数に戻るまでに行ったコラッツ操作の回数であり, $(n)_R = k$ と表す..

コラッツ逆操作:ある数がコラッツ列の要素として, その一つ前の要素を表す操作をコラッツ逆操作といい, $n^{(-1)}$ と表す. 一般に $n_1 \to n_2$ だからといって $n_2^{(-1)} = n_1$ とは限らない.

2　証明方法

コラッツ予想について考える上で単純にコラッツ予想を解くだけでなく, 別のアプローチからでも証明することが可能である. 以下その紹介と, それがコラッツ予想が成り立つことと同値である証明を行う.

2.1　第二数の存在

任意の自然数に対して第二数が存在するなら, コラッツ予想は成り立つ.

証明:n_0 の第二数が k_0 とする. $n_0^{(k_0)} = n_1,\ n_0 > n_1$.
コラッツ操作で出る数は必ず自然数であるため, n_1 に対しても第二数 k_1 が存在する.
同様の操作をどんなに多くとも n_0 回繰り返すことで必ず 1 に帰着する.

2.2　巡回列, 無限大発散列の非存在

任意の自然数に対してそのコラッツ列が巡回列及び無限大発散列でないなら, 要素数は有限で収まる.

証明:ある数 n のコラッツ列が要素数無限でありながら, 巡回列及び無限大発散列でないと仮定する. このコラッツ列に対する最大数を M とする. このコラッツ列は巡回列でもないため M 回コラッツ操作を行うと $M+1$ 種類の数が現れる. M 以下の数は M 種類しかないため仮定条件に反する. よってコラッツ列が巡回列及び無限大発散列でないなら, 要素数は有限で収まると言える.

3　コラッツ予想と 2 進数

コラッツ予想はそのアルゴリズムの計算の性質上 2 進数や 3 進数で表すとまたおもしろい見方が見えてくる. 今回は 2 進数を用いてコラッツ予想に対するアプローチを行う.

3.1　2 進数を用いたコラッツ操作

ある数 n に対してその数を 2 進数で表すことで短期的なコラッツ操作の増減の流れを推測することが出来る.

- 最小桁が 0 のとき, そこから連続する 0 の個数を t とし, 減少操作を t 回行う.
- 最小桁が 1 のとき, そこから連続する 1 の個数を t とし, 増加操作と減少操作の組みを交互に t 回繰り返す.

以下 k 進数の表記を, $n_{(k)}$ と表すこととする. ただし特に表記がない場合その数は 10 進数で表記されているものとする.

例)$100 = 100_{(10)},\ 10 = 1010_{(2)},\ 120_{(3)} = F_{(16)}$

2 進数の性質上, 2 で割るという操作は右に一つシフトすることと同じである. 2 進数で表したとき 0 が連続する数だけ減少操作を行うというのは明らか.

ある数 n に対してその数を 2 進数で表したとき, 最小桁数から 1 が t 回連続している場合を考える.

$$n = 1\cdots\cdots 0\ \underbrace{1\,1\cdots 1}_{t\text{ 個}}{}_{(2)}$$

3 倍して 1 を足すとは, 元の数と元の数に 2 倍して 1 を足したものとの和と同じである. n は明らかに奇数であるため, n に対して増加操作を 2 進数上で行う.

n に 2 倍して 1 を足したものは最小桁数から 1 が $t+1$ 回連続している数となる. よってこの数と元の数 n の和は最小桁数から 2 が t 回連続し, その次の桁が 1 となる. ただし今この計算は 2 進数上の計算であるため 2 以上の数に対しては繰り上がりを行う. 繰り上がりまで計算したときその値は最小桁数が 0, そこから $t-1$ 回 1 が続く数となる. この数は偶数であり減少操作を行うと, 最小桁数から 1 が $t-1$ 回連続している数となる. 今特に t に指定がなく, 任意の t に対して同様の性質が成り立つため, 増加操作と減少操作の組みを交互に t 回繰り返すことができるといえる.

$$\begin{array}{rr}
 & 1\cdots\cdots 0\ 1\ 1\cdots 1_{(2)} \\
+ & 1\cdots\cdots 0\ 1\ 1\cdots 1\ 1_{(2)} \\
\hline
 & 1\cdots\cdots\quad 1\ 2\cdots 2\ 2_{(2)} \\
\hline
 & 1\cdots\cdots\quad 0\ \underbrace{1\cdots 1}_{t-1\text{ 個}}\ 0_{(2)}
\end{array}$$

3.2 コラッツ予想と対応する動きをとるアルゴリズム

数字を 2 進数で表すことで短期的ではあるがコラッツ操作の増加減少の動きを予測できることが今までの結果から分かった. 増加減少の動きが予測できる場合, その動きが分かっている部分の計算を簡略的に行うアルゴリズムが作ることが出来るのではないかと考える. 以下そのアルゴリズムの紹介を行う.

- 状態が0で n が偶数なら 2 で割る. $\cdots(1)$
- 状態が0で n が奇数なら 1 を足し, 状態を 1 にする. $\cdots(2)$
- 状態が1で n が偶数なら $\frac{3}{2}$ をかける. $\cdots(3)$
- 状態が1で n が奇数なら 1 を引き, 状態を 0 にする. $\cdots(4)$

初期状態を 0 とし任意の自然数に対してこの操作を繰り返すことで, 有限回の操作のうちに状態が 0 で値が 1 に到達するならば, コラッツ予想を成り立つといえる.

またコラッツ操作を行いたい数が 2 で何度割ることが出来るのかさえ分かるのなら, コラッツ操作におけるステップ数の短縮が可能となる.

(1) の動作はコラッツ操作と同様に偶数なら 2 でわるというもの. (2) の動作は奇数が出たとき 2 進数での 1 が連続する回数だけ増加, 減少操作を繰り返すという性質を応用したもの. 先に 1 を足すことで, 連続していた 1 が全て 0 に代わり, その 0 の回数だけ増加, 減少操作を繰り返せば似た結果が現れる. (3) は増加, 減少操作を組み合わせたもの. ただし, 先に (2) で 1 を足し, (4) で調整を行うため増加操作は 3 倍する操作のみを行う. (4) はこの計算のズレの調整を行う.

例として16で割ったとき7余る数に対してこのアルゴリズムと同じ計算を行う．
例)nは16で割ったとき7余る数，よって2進数で表すと下の桁は$0111_{(2)}$となる．そのため3回増減操作を繰り返す．

$$n^{(6)} = (3\ (3\ (\ (3n+1)/2\ \ +1)/2\ )+1)/2 = (27n+19)/8$$

nに対して(2) ∼ (4)の操作を繰り返すと$n+1 = \cdots 1000_{(2)}$であるため，

$$(n+1)\cdot 27/8 - 1 = (27n+19)/8$$

となり一致する．以下7と15に対してコラッツ操作と今回のアルゴリズム，また今回のアルゴリズムの計算を省略したものを記載する．

7

$$7 \to 22 \to 11 \to 34 \to 17 \to 52 \to 26 \to 13 \to 40$$
$$\to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

$$7 \to 8 \to 12 \to 18 \to 27 \to 26 \to 13 \to 14 \to 21$$
$$\to 20 \to 10 \to 5 \to 6 \to 9 \to 8 \to 4 \to 2 \to 1$$

$$7 \to 8 \to 27 \to 26 \to 13 \to 14 \to 21$$
$$\to 20 \to 5 \to 6 \to 9 \to 8 \to 1$$

15

$$15 \to 46 \to 23 \to 70 \to 35 \to 106 \to 53 \to 160 \to 80 \to 40$$
$$\to 20 \to 10 \to 5 \to 16 \to 8 \to 4 \to 2 \to 1$$

$$15 \to 16 \to 24 \to 36 \to 54 \to 81 \to 80 \to 40$$
$$\to 20 \to 10 \to 5 \to 6 \to 9 \to 8 \to 4 \to 2 \to 1$$

$$15 \to 16 \to 81 \to 80 \to 5 \to 6 \to 9 \to 8 \to 1$$

4　今後の課題

今回証明には至らなかったが，前回の大宮祭の内容よりも2進数に対するアプローチを深めることで，幾つかおもしろい発見や考え方などが生まれた．今後の課題としては，短期的な増減の流れだけでなく，それ以降の動きに対してどのように対応するのかを考える方法を模索したい．また今回新たに導入したアルゴリズムに対してはあまり考察できなかったため，その点からも考えていきたい．

5　参考文献

[1]　Richard K. Guy, 金光滋, 数論＜未解決問題＞の辞典，朝倉書店，2010.

[2]　太田悠暉, コラッツ問題にぶつかりましたっ(仮),
http://sitmathclub.web.fc2.com/seisaku/collatz-04-18-2.pdf, 2015/10/18.