

コラッツ予想マッピング

Abstract

This study investigates the Collatz conjecture with using numbers that give a remainder of 4 when divided by 6. I call these numbers “core numbers.” I examined the patterns and properties of core numbers and used programming to test how these patterns appear in many different cases. Through this process, I found that some special odd numbers, created from core numbers, may play an important role in understanding and possibly solving the Collatz conjecture.

この研究では、核の数と命名される、6 で割ると 4 余る数の性質に着目して、コラッツ予想の性質に迫る。また、その性質から見つかる課題を踏まえ、プログラミングを使用して任意の数を表す。結果、コラッツ予想の解決のキーとなるのは、核の数から生み出される、特殊な奇数であることが分かった。

目的

本研究では、コラッツ予想に対し、自分なりのアプローチで

- ①コラッツ予想の課題を明確にすること
 - ②性質や法則性を見つけ、プログラミングを用いて視覚的に表現する。
- ことを目標とする。

ここで、コラッツ予想とは

自然数 n に対し、④偶数でなら 2 で割る。

⑤奇数でなら 3 を掛け 1 を足す。

という操作を繰り返すと、いずれの数から始めても 1 に収束する。

という予想のこととする。

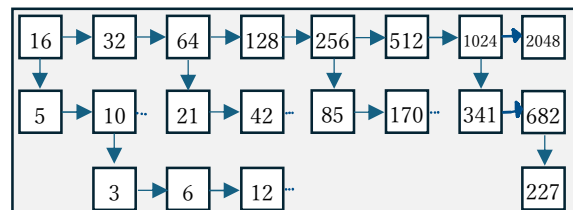
方法

コラッツ予想の計算の流れを観察し、そこから規則性を考える。その規則性を利用し、scratch というプログラミングのフリーサイト上でプログラムを組み、視覚的なマップを作る。

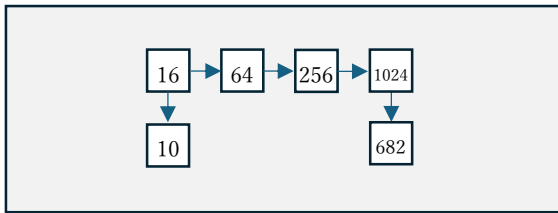
内容-①

コラッツ予想のフローの観察/性質の利用法

以下にコラッツ予想の流れを示す。



このように表すと、矢印が下に伸びる数(3 を掛け 1 を足した数) m (上図では 16,10,64,256,1024,682 が該当)は必ず、 $m \equiv 4 \pmod{6}$ となっていることがわかる。ここから、このような数を**核の数**と呼ぶ。核の数から、別な核の数につながるまでの核の数でない数、例えば 16 を例にすると、32,10 は、核の数に代表させた数の集まりに属するとし、これらを**核の数群**とまとめて呼ぶ。このように、上に示した図を、核の数群の考え方のもと、圧縮すると、



このように簡潔に表すことができる。特に,10 の下側などは素因数に 3 を含むため,前述の表し方だとずっと右側に連なり,そこから核の数は生まれえない。このような事象を切り捨てて簡潔に表すことができるのがこの表し方の利点の一つだ。

この考え方をを用いることで,コラッツ予想の課題は,

- ①すべての数がいずれかの核の数群に含まれるか
- ②核の数は別の核の数に 1 本または 2 本の矢印でつながるか

にまとめることができる。

内容-②

プログラミング

前述した課題が,成立すると仮定して,scratch でプログラムを組んだ。

プログラムについての詳細な説明は本レポートの最後に記載する。

結果

任意の数字(ある程度小さい数,六桁ほどまで)について,異常を示すことはなかった。

以下に実験結果の一例を示す。



結果

前述した関数に関して異常を示すような代入値は見つからなかった。ここから,課題①に対しては,具体的な証明はなされていないものの成り立つだろう。

課題②については,証明の方針や異常が起こりそうな条件などを見つけることはできなかった。これについては,自身が知っている検証法ではあまり効果が無く,より高度な論証によるアプローチが必要だろう。

展望

核の数の性質を調べていると,コラッツ予想においてより重要な要素となるのは,核の数の下側にできる奇数であると感じた。すべての自然数が $2^n \times (\text{奇数})$ (n は 0 以上の整数) で表せるから,すべての奇数が核の数の下側に生成できることを証明できれば,課題①は数学的な証明を得ることができるだろう。

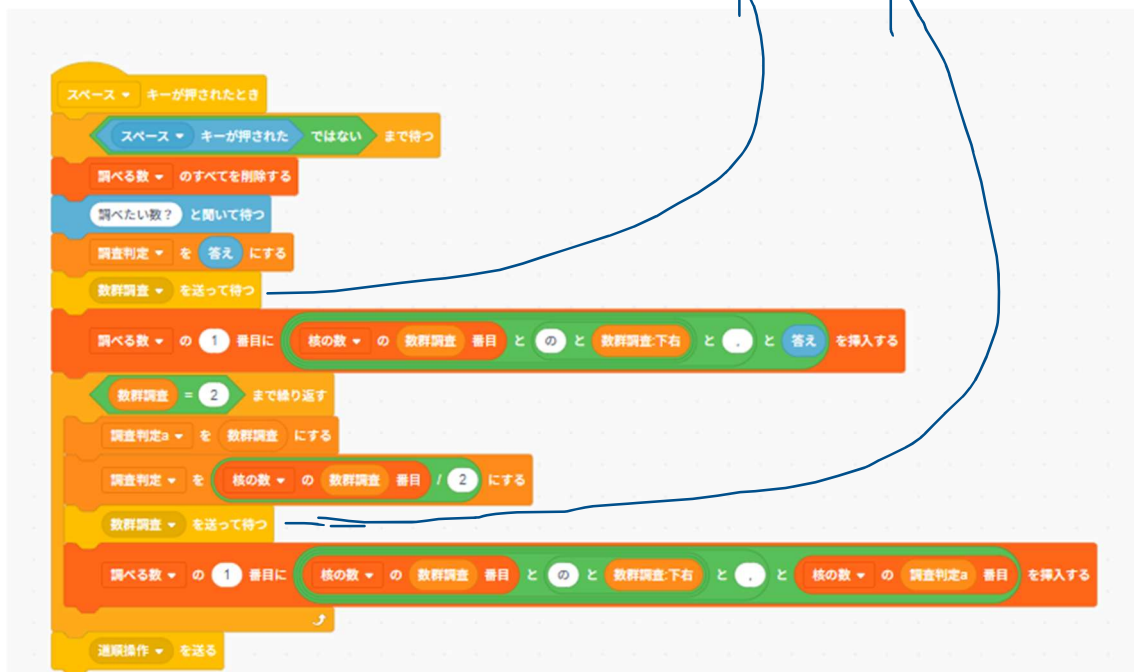
課題②に関しては,今のところ解決の糸口はつかめていない。より専門的な数学的論理を学んでから再挑戦することとする。

参考文献

無し



←後述する,代入した値に対し核
の数群を調べる関数の定義
同時に核の数群の右/下側かも判
定している



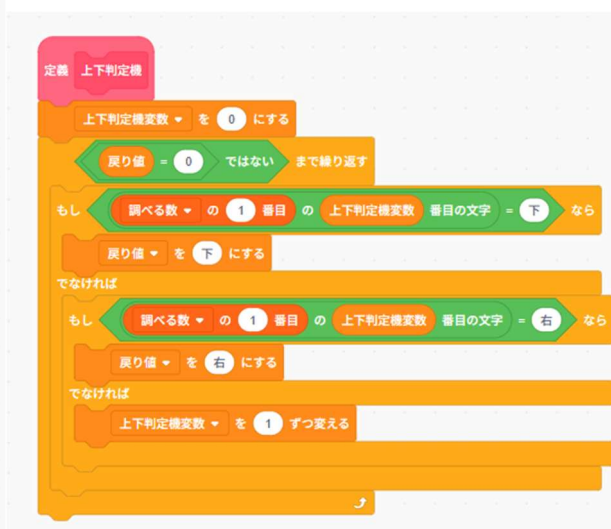
↑入力された数に対し,関数に繰り返し
代入し道順を判定/記録する



←記録された道順を読み取り,画面上につながりを表示する。

↓道順記録を読み取るための関数

道順の記録は人間用に読みやすくしているので,プログラムとして必要な箇所を読み取るために定義として設定した。



これらのプログラムでは、『代入された数字に対して,その数が何の核の数群に属しているか調査し,代表の核の数を,核の数が小さい順に並べられたリストの通し番号で返す関数』を制作し,入力された数を代入し調査,その結果をさらに関数に代入して調査することを繰り返す。

同時に,核の数がとある数群に含まれる時に,数群の代表の核の数の右にあるか下にあるか(核の数 10 を例にとると,20 は右,3,6,12 は下)を同時に判定,結果と右か下かを記録してリストに保存する。関数による調査が終わり,道順が最初から最後までリストに記入されたら,そのリストをもとに,画面上で道のりを表示する。