４マスルービックキューブの神の数字

研究者
清水　秀一朗　　平　俊知　　中村　光希　　林　大宝

指導教諭
松澤　一弘　先生

要旨(Abstract)

ルービックキューブには，どのような状態からでも完成状態に戻すための最小手数である「神の数字」というものが存在する．３マスルービックキューブにおいては神の数字は２０手であることが証明されているが，４マスルービックキューブでは発見されていない．そこで３マスルービックキューブでの神の数字の研究を参考に，数学の群論によってルービックキューブを数学的に定義し，プログラミングを用いて４マスルービックキューブの神の数字を求める方法について研究をした．

なお今回の研究ではプログラミングを完成させることができず，神の数字を証明することはできなかった．

1. はじめに

1.1 研究の動機

ルービックキューブ(以下，ｎマスルービックキューブを RC{ｎ}と示す）とは，一般的に 3×3×3 の形で知られる立体型パズルである．現在 RC は立方体の他に四面体型，スネーク等様々な形状のものが存在している．RC{ｎ}にはどの状態からでも必ず元に戻すことができる最小の手数が存在し，これは「神の数字（God Number）」と呼ばれている．

1.2 先行研究

2010 年に RC{3}の神の数字は 20 手であることが証明された．その方法は，群論によって考える状態を絞り，更に対称性から調べる必要のある状態を絞るという方法である．全パターンに対し，向きが前後上下左右反対の場合を踏まえて，1/24 に，さらに鏡に写したときの場合も踏まえて 1/2，計 1/48 にまで計算する必要のある状態を減らす．この対称性によって除かれた状態というのは，面とブロック面の位置関係（後に定義する）を踏まえると別の状態として区別されるが，それらの位置関係を無視し，ただかかった手数で考えれば，全く同じになるものである．

絞られたすべての状態を Google にあるコンピュータを多数用いて計算し，最も手数がかかった状態の手数が神の数字となる．これは優れたデスクトップ PC 一台では約 35 年間かかってしまうほどの計算量である．

また証明までには至らなかったが，上記のものほどの計算量を必要とせず神の数字に迫る「シスルウェイト法」というものとその改良版がある．この方法では神の数字の上端を探す．下端は様々な状態を作り必要な手数を数え，より手数が必要な状態を探して上端に近づけていく．神の数字の上端はまず群論より，ルービックキューブを操作の性質で段階分けし，下の段階からひとつ上の段階に行くのに必要な最大手数を足し合わせていく．段階分けの仕方を工夫し，下端に近づけていく．そして上端と下端が一致したときが神の数字となる．この方法は 40 年以上前から研究されたが結局神の数字を証明することはできなかった．

2. 群論

今回の研究で必要となる群論の範囲についてのみ書いておく．

2.1 群の性質

集合に演算を定義した時，ある条件を満たすことで群となる．集合とは元（要素）の集まりのことである．例えば自然数全体の集合などがある．演算とは，集合上で二つの元を入れ別の元を作る事である．足し算などが例として挙げられる．条件は四つあり，一つ目は任意の元の演算結果もまたその群に含まれる元であること．二つ目は単位元（演算に影響を与えない元）が存在すること．三つ目は，全ての元に逆元が存在すること．逆元とは，演算後単位元となる相手のことである．四つめは結合法則が成り立つことである．上記を満たせば群となる．

表記の仕方として（[集合]，[演算]）の形がある．

2.2 部分群 [2]

群（G，・）の部分集合 H が G の演算 ・ で群となるとき， H は部分群であると定義される．必要十分条件として，

①群内の演算が群内で閉じていること
②単位元が存在すること
③逆元が存在すること
を確認すれば良い．すなわち，

『H が G の部分群 ⇔ ①x, y ∈ H なら x・y ∈ H ②G の単位元は H の元 ③ x ∈ H なら x^{-1} ∈ H の 3 つが成り立つ』 である．

2.3 同値関係 [3]

2.3.1 二項関係について

集合 A 上の二項関係とは， A の要素を 2 つ並べたものをいくつか集めた集合のことである．例を挙げると， A ＝ {1，2，3} に対して，{（1，2），（2，1），（3，1），（3，3）} は二項関係となる．（a, b）が考えている二項関係に属するとき，a〜b と表記する．

2.3.2 同値関係について

同値関係とは，たとえばある集合の要素 a, b, c について，

① 反射律（a〜a）

② 対称律(a〜b ⇒ b〜a)

③ 推移律(a〜b ∩ b〜c ⇒ a〜c)

の 3 つを満たす二項関係 〜 のことである．②，③については，「仲間である関係」を表していると言える（②: a が b の仲間なら b は a の仲間，③: a と b が仲間で b と c が仲間なら a と c は仲間）

同値関係は，「同じ」であることを幅広く考えるために用いられている．「＝」は左辺と右辺が数値・集合として完全に一致するときに用いられるが，条件として厳しい場合があるため「＝」を一般化する必要があった．

2.4 類別

2.4.1 類別と同値類［4］

一般に，集合 S の元 a，b の間に同値関係 (a〜b) と呼ばれる関係が定義されるとき，全ての元は a と同値関係に ある or ない の 2 つに分けることができる．全ての元は a と同値であるか否かであるから S は過不足なく綺麗に 2 つに分かれる．このようにして分割した各組を「集合 S の類」と呼び，この組み分け方を「類別」と呼ぶ．

群の元をさらに小さな部分集合に分けることで，いろいろな種類の元が混じり合った群を分類することができ，問題が簡単になることがある．

2.4.2 剰余類（左剰余類）［5］

H を群 G の部分群とする．a, b ∈ G に対し，a^{-1}・b ∈ H となるとき a〜b と定義する．これは同値関係の定義を満たす二項関係となる．ここで，a と同値関係にある任意の x について a^{-1}・x=h とおくと，x=a・h より x ∈ aH となる．また任意の H の元 h に対し，x=a・h とするとき，a^{-1}・x=h となる．したがって a と同値であるものはすべて aH の要素となり，aH の要素はすべて a と同値となる．a の同値類を Ca と置くとき，Ca⊂aH かつ aH⊂Ca より，aH=Ca となる．そしてこの aH を左剰余類とよぶ．G = H ∪ aH ∪ bH ∪・・・ のような左剰余類による類別を，G の H による左類別と呼ぶ．全ての左剰余類それぞれの位数（元の個数）は等しい．

2.5 完全代表系［6］

H を G の部分群とし，G の部分集合 {a, b, c, ...} で，重複なしに

G = H + aH + bH + ・・・

となるものを完全代表系という．

つまり群 G の H による左類別をしたときに複数の左剰余類に同時に存在する元がないようなとき，そのときのそれぞれの左剰余類の代表を取ってきた集合を完全代表系という．当然この全ての元は群 G 上にあるので G の部分集合となる．

3. RC{4} と群論について

3.1 置換について

置換

例）6次の置換 p

$$p = \left(\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 4 & 6 & 5 \end{array}\right)$$

6個

1→2　2→3　3→1
4→4　5→6　6→5

= (1 2 3)　(5 6)　　1→2→3→1　5→6→5　と巡回

= (1 3)　(1 2)　(5 6)　　互換で表現
互換奇数個→ p は奇置換

置換とは，1，2，3，・・・，n を並び替える操作のことである．この場合は n 次の置換と呼ぶ．置換は，「もとの元」を上に並べて「行き先」を下に並べた形で表現される．置換によって「（1，2，3，・・・，n）を並べ替えた状態」は 1 から n までの順列となるため，順列と置換は対応する．つまり，n 次の置換は全部で n!個存在する．一つの置換内で循環するように置き換えられているのを横に並べて書き，それらの積という形で表現できる．これらは巡回置換という．さらにそれらを二つの元の置き換えの積の形に直し表現できる．この二つの元の置き換えは互換と呼び，ある置換に含まれる互換の数の偶奇でその置換は偶置換か奇置換に分けられる．

置換の積

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 3 & 1 & 4 & 6 & 5 \end{pmatrix} \quad q = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 2 & 5 & 3 & 6 \end{pmatrix}$$

$$q \cdot p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 2 & 1 & 5 & 6 & 3 \end{pmatrix}$$

置換pをしてqをする

考え方
1→2 by p、2→4 by q、
よって1→4 by p・q

複数の置換を連続して行うとき，置換の積として考える．表現の仕方は普通に積の形で表すが，×は使わない．積の順番は右から置換を行う．積の考え方は具体例を見ながら説明すると置換 p では 1 が 2 の場所へ，置換 q では 2 の場所にあるのが 4 へと置き換えられるので，1 が 2 の場所へ行きその場所は 4 へと置き換えられると考えて，1 が 4 に置き換えられる．

ある置換の互換の積による表現の仕方はいくつかあるが，偶奇は常に一定であることが証明されている．よって偶置換同士，奇置換同士の置換の積は偶置換（互換の数の和が偶数）偶置換と奇置換の積は奇置換（互換の数の和が奇数）になる．

3.2 置換群と RC の関連

RC と置換群の定義は以下のように合致する．

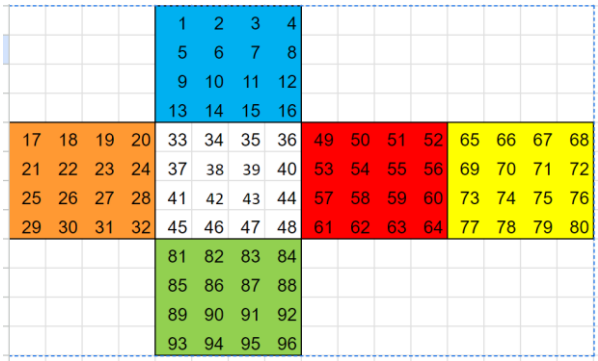
(1) 置換を元，結合を演算とする群
　⇔操作がありその連続で成り立つゲーム
(2) 任意の置換の結合 ⊂ 置換群⇔
　動きを連続させてもそれは RC に存在する
(3) 単位元が存在⇔操作しないという操作
(4) 逆元が存在⇔行った操作の逆操作
(5) 演算に結合法則が成り立つ
　⇔操作①、②、③について
　（①・②）・③＝①・（②・③）

3.3 一般的な解法

解法を群論で考える．以降説明するが部分群法に基づいているとみることができる．どこからそろえていくのかの順番を工夫して解いていっている．

3.4 RC{4}の面とブロック位置の定義

以下の RC{4}の展開図は市販の面の色と対応している．



全ての面には図のように正面に F（Front），上面に U（Up），下面に D（Down）右面に R（Right），左面に L（Left），背面に B（Back）と定義している．どんな操作をしてどんな状態になっても，この最初に定義した面の位置関係は変わらずそのままで面を区別するよう考える．

全てのブロック面に図のように数字を振り分けることでブロック面の位置を定義する．任意の操作によってブロック面の場所は変化しても，ブロック面の位置関係は変わらずそのままでブロック面の位置を区別するよう考える．

以降説明の中で一面体，二面体，三面体とブロックを呼び分けるが，これは文字通り面の数が何枚かで区別した呼び方である．

さらに今後説明上便利にそれぞれのブロック面の位置を表現する方法も決めておく．表現の仕方はまずどの面にあるか，次にどの面と隣り合わせか，三面体については隣の面は時計回りに隣にある面で，それぞれの面の記号の小文字をこの順に並べて表現，二面体はそこにさらにねじれの位置にある二つの面の近い方にある面の記号の小文字を並べて表現する．具体的には，40:fru　44:frd　84:dr　96:db の様になる．

群論の考え方を用いて考察やプログラミングを行う際には，ブロック面がすべて区別されている必要がある．

3.5 基本操作の定義

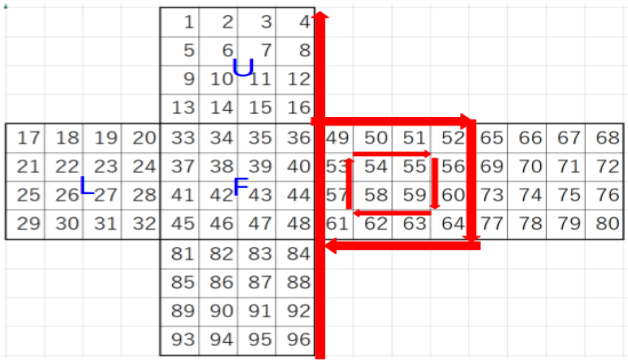
面の位置の区別を踏まえて基本操作の種類を，ブロック面の位置の区別と置換を踏まえて操作自体の数学的定義を行う．

基本操作の種類の定義はすでに定義した面それぞれの，時計回り 90° 回転とする．その操作を表す記号は面を区別する記号をそのまま使う．また RC{4}においては回転可能な部分（枚と数える）が面に隣接した内側部分にもあるので，それらを動かす操作についても似たように定義

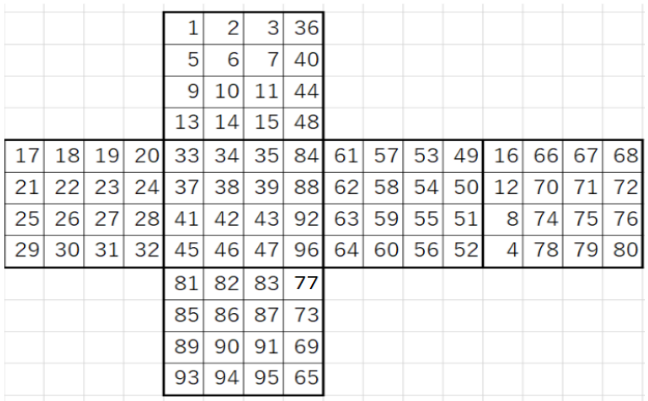
する．全ての面一つにつき一枚ずつ隣接していることが分かるので，それぞれの面のそれよりも中間側にあるのを，これらの部分を隣接する面の記号を小文字にして表現する．面は 6 つ，内側に隣接して回転させられる部分が面一つにつき一枚ずつあるため基本操作の種類は全てで 12 種類（F, B, R, L, U, D, f, b, r, l, u, d）である．

操作の数学的定義の具体例及び図の説明を以下に示す．

最初



基本操作 R



R=(4,77,84,36) (8,73,88,40) (12,69,92,44) (16,65,96,48) (49,52,64,61) (50,56,63,57) (51,60,62,53) (54,58,59,55)

最初と書かれた図はブロックの位置関係を定義した段階で何の操作もしていない状態のことで，赤い矢印は操作 R によるブロックの移動の仕方である．そして基本操作 R と書かれた図は基本操作 R によって動いたブロックそれぞれが操作の前の段階でいた位置を書いてある．さらにそれらの下には，操作 R によってどの位置にいたブロックがどの位置に移動したかを置換の考え方と対応させ，操作 R を置換として表現したものがある．置換の中の一つの巡回を見ると，4→77，77→84，84→36，36→4 とブロックが移動していることがわかる．つまり操作 R によって 4 の位置にあるブロックは 77 の位置に，77 の位置にあるブロックは 84 の位置に，・・・と変化するということを表現している．図を見ると確かにそうなっていることがわかる．

以上のように基本操作の種類とそれぞれの操作自体の定義及び表現を行う．

すべての基本操作について置換による表現を行った．

RC{4}の基本操作

R=(4, 77, 84, 36)(8, 73, 88, 40)(12, 69, 92, 44)(16, 65, 96, 48)(49, 52, 64, 61)(50, 56, 63, 57)
(51, 60, 62, 53)(54, 55, 59, 58)
r=(3, 78, 83, 35)(7, 74, 87, 39)(11, 70, 91, 43)(15, 66, 95, 47)
L=(1, 33, 81, 80)(5, 37, 85, 76)(9, 41, 89, 72)(13, 45, 93, 68)(17, 20, 32, 29)(18, 24, 31, 25)
(19, 28, 30, 21)(22, 23, 27, 26)
l=(2, 34, 82, 79)(6, 38, 86, 75)(10, 42, 90, 71)(14, 46, 94, 67)
U=(17, 65, 49, 33)(18, 66, 50, 34)(19, 67, 51, 35)(20, 68, 52, 36)(1, 4, 16, 13)(2, 8, 15, 9)
(3, 12, 14, 5)(6, 7, 11, 10)
u=(21, 69, 53, 37)(22, 70, 54, 38)(23, 71, 55, 39)(24, 72, 56, 40)
D=(29, 45, 61, 77)(30, 46, 62, 78)(31, 47, 63, 79)(32, 48, 64, 80)(81, 84, 96, 93)(82, 88, 95, 89)
(83, 92, 94, 85)(86, 87, 91, 90)
d=(25, 41, 57, 73)(26, 42, 58, 74)(27, 43, 59, 75)(28, 44, 60, 76)
F=(13, 49, 84, 32)(14, 53, 83, 28)(15, 57, 82, 24)(16, 61, 81, 20)(33, 36, 48, 45)(34, 40, 47, 41)
(35, 44, 46, 37)(38, 39, 43, 42)
f=(9, 50, 88, 31)(10, 54, 87, 27)(11, 58, 86, 23)(12, 62, 85, 19)
B=(1, 29, 96, 52)(2, 25, 95, 56)(3, 21, 94, 60)(4, 17, 93, 64)(65, 68, 80, 77)(66, 72, 79, 73)
(67, 76, 78, 69)(70, 71, 75, 74)
b=(5, 30, 92, 51)(6, 26, 91, 55)(7, 22, 90, 59)(8, 18, 89, 63)

基本操作のみの組み合わせでどんな種類の操作も表現できる．

3.6　RC{4}の単位操作の定義

「シングマスター記法」

単位操作とは一手と数える操作のことで，シングマスター記法を用いて定義，表記することとする．

63 通りあり，具体的には以下の通り

F　F^2　F^3　f　f^2　f^3　Fw　Fw^2　Fw^3

同様のパターンが B U D R L についてもある

内側二層の回転　M(=l・r^{-1})　E(=d・u^{-1})

S(=b・f^{-1})　それぞれ l, d, b の向きに回転

表記と操作の対応は以下の通りになる．

F^2 とは操作 F（前面を時計周りに 90°回転）を二回続けて行う操作（180°）

F^3（前面時計回り 270°）＝　F^{-1}（前面反時計回り 90°）

Fw←F と f を同時に行う　Fw＝F・f

これらは全て基本操作から定義されることができている．

群論において RC の群を定義するときには基本操作のみで十分だが、実際に RC を使用する際の操作の種類はこれらの様になる．

3.7　RC 群の定義

基本操作より RC{４}のルールに則った操作（バラバラにするなどしない）によって作られる全ての状態の群（＝全ての状態を表現できる置換の群）を RC 群 G とすると

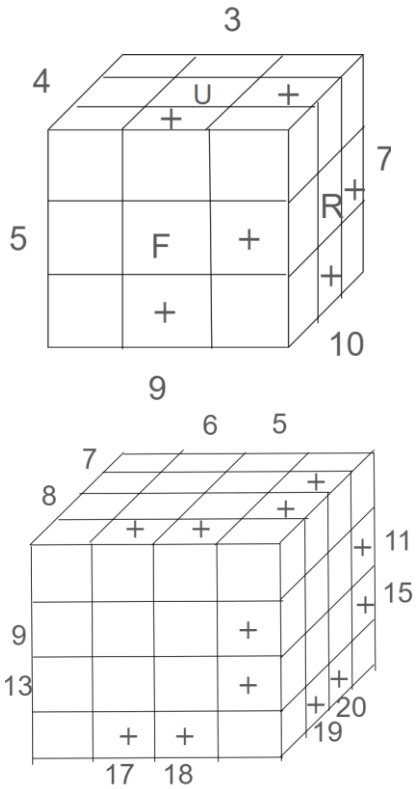
G＝　{F,B,R,L,U,D,f,b,r,l,u,d}　となる

これが意味するのは，RC 群はこれらの任意の置換（基本操作）を何個でも持ってきた時の演算結果（続けて操作したときのこの，置換の積）の置換全ての群である．

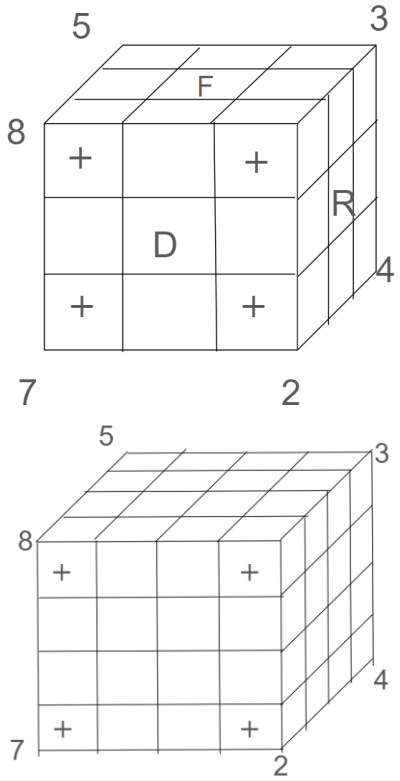
この定義によって具体的に操作によって作られる状態は，プログラミングを用いて置換の演算と群論の計算で考えることができる．この群の位数はルービックキューブの全パターンとなる．

3.8　RC の 2 面体の向きと番号付け

2 面体



3 面体



RC 群を定義するときとは別に RC の二面体と三面体について向きと位置を定義していく．これは RC の群構造について考察しキューブ理論の証明を行うのに必要である．番号を付けて位置を，印を付けて基準参照面を決めて向きを区別する．

RC{4}と参考元の RC{3}についても書いておく．

RC{3}の見えない部分については，二面体では dl,db,lf,lu,bl,bu ブロック面に印を，三面体には U 面側の全ての三面体に印を付けている．RC{4}の見えない部分には RC{3}の時と照らし合わせて見て，同じように印がついている．

この定義によって，二面体の置換と反転，三面体の置換と回転を考えることができる．二面体の反転について，ある操作によって置換されたとき，それぞれの位置にある二面体の付けられている印の場所が完成状態の時のその位置における印の場所と同じか逆かに分かれ，逆の時は反転したと表現する．三面体の回転について，二面体と同様に考え，それぞれの位置における印が時計回りに 0°,120°,240° の三パターンに回転しているのを区別する．

二面体は RC{3}に 12 個，RC{4}に 24 個あり，三面体は RC{3}，RC{4}ともに 8 個ある．

3.9 キューブ理論の第一基本定理（RC{3}）

次の決定過程により，RC の配置は決定される．

(a) どのように二面体が置換されたか

(b) どのように 三面体が置換されたか

(c) どの二面体の印が反転したか

(d) どの三面体の印がどれだけ回転したか

以上の定理から全ての基本操作に対応する新たな置換（二面体，三面体）とそれぞれの位置の反転，回転を表す関数（巡回群と同一視）を定義して，群の部分群による半直積，準同形写像の考え方を用いて，四つの要素を四つ組みとして考え，演算を行っていくことでキューブ理論の第 2 基本定理を証明する．詳しいことは今回の研究では触れない．

3.10　キューブ理論の第 2 基本定理（RC{3}）

(a) 二面体と三面体どちらもどんな操作でも偶置換か奇置換かが一致している．つまり，どんな操作でも二面体と三面体の入れ替わった数の偶奇は一致する．

(b) 三面体の回転について，0°（360°）回転を 0，120° を 1，240° を 2 として値を振ったとき，全ての三面体の回転の値の和は 3 を法として 0 と合同である．つまり全ての三面体のうち一つ以外の回転量が決まれば残り一つの回転量も決まる．

(c) 二面体の反転について反転の有無にそれぞれ 1，0 として値を決めると全ての二面体の反転の値の和は 2 を法として 0 と合同である．つまり全ての二面体のうち一つ以外の全ての反転の有無が決まれば残り一つも決まる．

これらの定理は第一基本定理を踏まえ RC{3} 群の群構造を解析していく中で満たされなければいけない条件として与えられる．

3.11　神の数字を求める方法

部分群法→シスルウェイト法

3.11.1　部分群法

以下部分群法の数式による処理は「群論で解き明かすルービックキューブの数理」より引用．（参考文献［7］）

RC{4}　群　G={F, B, R, L, U, D, f, b, r, l, u, d}（RC{3} なら G={F, B, R, L, U, D}）の部分群系列 Gn={1}⊂Gn-1⊂Gn-2⊂…...⊂G1⊂G0=G を用いて次の処理を行う．

与えられた元 g0∈G0 は与えられた RC の状態 Gk/Gk+1 の完全代表系を決めておく．

Gk/Gk+1 = ∪(i=1~rk) gk+1, i Gk+1

【初期処理】（ある i∈{1, 2, 3, ..., r1} について）

g0∈g1, i G1 となるとき

g1=g1, i 及び, g’1=$g1^{-1}$ g0 とする．

(g’1∈G1 となる)

【再帰的処理】

g’k∈Gk がすでに定義されていて

（ある j∈{1, 2, ..., rk} について）

g’k∈gk+1, j Gk+1　となるとき，gk+1=gk+1, j 及び　g’k+1=gk+1^{-1} g’k とする(g’k+1∈Gk+1 となる)．

以上から g’1=$g1^{-1}$ g0　g’2=$g2^{-1}$ g’1 …… g’n=gn^{-1} g’n-1　(gn=1)となり

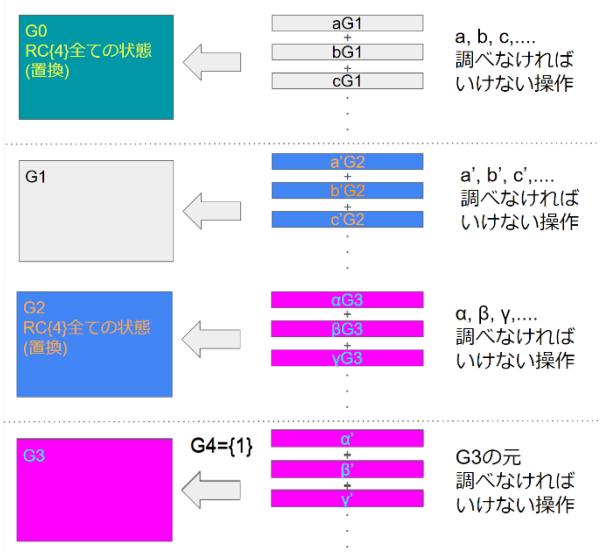
すべてまとめると

1=gn^{-1} $gn-1^{-1}$ $gn-2^{-1}$　……　$g2^{-1}$ $g1^{-1}$ g0

g0=g1 g2 g3……gn-1 gn

ここまでで示されているのは，ルービックキューブ群の部分群の部分群の部分群の…と考えたときに，より位数の小さいという意味でひとつ下の段階の群(順番を示す右下に添えられた数字は下の段階の群ほど大きい)によって左類別をしたときの完全代表系を決め，（決め方はたくさんある）一つ前の段階の代表の置換（操作）と下の段階との関係を以上の処理に従いまとめることによって，ルービックキューブにおけるある状態は，最も下の段階の操作（完全代表系に含まれる操作）から順々に上の段階の操作を行っていき得られる状態によって作られているということである．

これらの図のイメージは以下のようになる。



段階分けの上下の段階の関係はスライドのように、上の段階の中身は下の段階といくつかの操作それぞれとの連続の集合でできている。そのいくつかの操作全てが神の数字の上端を求めるのに調べなければいけない操作全てになる。この関係性が続き、最も下の段階である G4 は元が単位元の 1 のみなので、スライドの様になりる。以上を逆算していき、G3 の含むすべて置換の最大手数に、G3 との連続で G2 を作る置換の最大手数を足す、といった具合に一番上の段階であるルービックキューブ群まで調べれば神の数字の上端を求めることができる。

3.11.2 シスルウェイト法(RC{3})

段階分け

G0=<R,L,F,B,U,D>

(RC{3}群　RC{3}の全ての状態)

G1=<R,L,F,B,U^2,D^2>

（上面と下面は 180°回転のみ)

G2=<R,L,F^2,B^2,U^2,D^2>

（右面と左面以外は 180°回転のみ)

G3=<R^2,L^2,F^2,B^2,U^2,D^2>

（180°回転のみ)

G4=<1>

（操作しない　位数 1)

G⊂G1⊂G2⊂G3⊂G4 となっている.

部分群法を用いた神の数字の上端の求め方

G0/G1 の完全代表系 {g1,i |1 以上 i n1 以下|} (n1=2048)ですべての手順 g1,i が高々7 手となる物がある.

これは g1,i（複数の，あるいは単にそれだけの単位操作によって得られる状態をつくる操作)は 2048 種類あり，その全てについてプログラミングで手数を数えると高々7 手となる，ということを示している.

同様に G1/G2(n2=1082565)[高々13 手],

G2/G3(n3=29400)[高々15 手],

G3/G4(n4=663552)[高々17 手]

と調べ，7+13+15+17=52 手と神の数字の上端を求めることができる.

n1(2,3,4)の値については単に群論の計算で求められる.

ちなみに n1×n2×n3×n4＝［3RC 群の位数］となっており，この段階の分け方における 3RC の全ての状態を考察できていることがわかる.

段階の性質

G0/G1

この完全代表系に含まれる手順は二面体ブロックの向きを変えるだけである．これは G1 に含まれる操作では二面体の向きのみを変えることができないことが理由である.

G1/G2　三面体の向きを変えるだけ

これは G2 に含まれる操作では三面体の向きを変えることができないからである．G2 は G1 の部分群なのでもちろん二面体の向きも変えられない.

G2/G3　すべての三面体および二面体を正しい位置に移す

G3/G4　これは単に G3 全体を表す(G4＝1)

以上の段階分けによって完成状態を作るのに必要な操作の性質を絞ることができ，その操作の全てをプログラミングによって調べることで，神の数字を求めるのに必要なコンピュータによる計算量を大幅に減らすことができる.

この数学的処理のイメージは，具体的に一手一手を動かしながらできていく状態を考えるのでなく，ブロック全体を同時に入れ替えながら，その入れ替えの条件を絞っていき，最終的にそれぞれのブロックの位置を定めるということである.

段階分けの仕方の改善，またはコンピュータによる計算で考える幅を増やすなどの工夫をしていくことで神の数字の上端を小さくしていく.

4. 結果及び考察

以上で行った定義を踏まえ，RC{3}においての神の数字に迫る研究を RC{4}にも当てはめる形で考察を行った.

4.1　キューブ理論の第 1 基本定理（RC{4}）

RC{3}の時との相違点は，二面体三面体だけでなく一面体の置換についても考えないと RC の配置が決定されない可能性があることである．これは RC{3}の時は面の位置と一面体の位置が 1 対 1 に対応しているが RC{4}の時は一つの面に一面体が 4 つあるからである．一面体の位置のみを入れ替える操作を見つけられれば，一面体の配置は二面体，三面体の配置，向きにかかわらず置換できるため，このことが証明できる．群論用プログラミング GAP を用いて RC 群を定義すれば確かめることができる．一面体については向きは関係ない.

4.2 キューブ理論の第 2 基本定理(RC{4})

詳しい定義や演算をすることが困難であったので，証明とはいえないが，第 2 基本定理が成り立っているであろうことを主張できるよう，具体的に定義した基本操作について見ていく.

(a)二面体と三面体どちらもどんな操作でも偶

置換か奇置換かが一致している

全ての基本操作について置換される二面体，三面体の数の偶奇は同じであるため，基本操作における偶奇は一致している．また，全ての置換は互換の積による表現の仕方がいくつかあっても偶奇は決まっているので，偶置換同士，奇置換同士の積は偶置換，偶置換と奇置換の積は奇置換になるという約束を守り続けて，最終的な偶奇も一致すると考えられ，この定理も成り立っている．

基本操作 F,B,R,L,U,D←全て三面体を 4 つ（奇置換），二面体を 8 つ（奇置換）置換

基本操作 f,b,r,l,u,d←全て二面体を 4 つ（奇置換）置換

(b)三面体の総捻り量は保存される

三面体の位置の置換と向きの入れ替えに作用する操作の種類，数は完全に RC{3}と一致しているので RC{4}においても保存されると考えられる．

(c)二面体の総反転量は保存される

基本操作 f,b,r,l,u,d について，操作の前の段階で位置関係を最初の定義通りに基準参照面の印を付けたとき，操作後も二面体の反転を行わない．よってどのような状態でも反転している二面体は置換後も反転し，反転していないものは置換後も反転しない．

基本操作 F,B,R,L,U,D について，操作の前の段階で位置関係を最初の定義通りに基準参照面の印を付けたとき，置換後二面体のうち 4 つ反転し，4 つ反転しない．よってどのような状態でも操作後に 4 つは反転の有無を変えず，4 つは反転の有無が逆転するので反転の値の和が 2 を法として合同であることは変化しない．従って全ての基本操作でどんな状態で操作してもこの定理は満たされ，全ての操作は基本操作の積なので常にこの定理は成り立つ．

以上のように考えると RC{4}でも第 2 基本定理が成り立っていると考えられる．

4.3　シスルウェイト法の応用（RC{4}での方法）

RC{3}同様 RC 群を定義でき，群の性質は変わらないので，群 RC{4}でも部分群法を用いることができる．

従って段階分けの仕方を工夫し，プログラミングをうまく作ることができれば，RC{4}でも効率よく神の数字の上端を求められると考えられる．ただし，RC{3}の時より圧倒的に全パターンが多いので必要な計算量もより大きくなってしまう．しかし，シスルウェイト法の段階分けの仕方を工夫することで，計算量も減らし，効率的に求めることができるのではないかと考えられる．以下に二通りの我々の考えた段階分けの仕方をのせる．

4.3.1　元のシスルウェイト法を参考にした段階分け

G0=<R, L, F, B, U, D, r, l, f, b, u, d>

G1=<R, r, L, l, F, f, B, b, U^2, u^2, D^2, d^2>

G2=<R, r, L, l, F^2, f^2, B^2, b^2, U^2, u^2, D^2, d^2>

G3=<R^2, L^2, F^2, B^2, U^2, D^2, r^2, l^2, f^2, b^2, u^2, d^2>

G4=<1>　（操作しない　位数 1）

これは RC{3}におけるシスルウェイト法をそのまま RC{4}に当てはめたように段階分けをしたものである．下の段階が上の段階の部分群となっているため，シスルウェイト法の考え方を用いることができる．

4.3.2　RC{3}の神の数字を活用

まず，RC{3}が RC{4}の部分群であることを示す．RC{3}群を 3G，RC{4}群を 4G として，

3G=<R,L,F,B,U,D>

4G=<R, L, F, B, U, D, r, l, f, b, u, d>

ここで小文字の操作について，l・r(=M) d・u(=E) b・f(=S)と固定する（l,r,f,d,u,b の操作を単独で行うような操作は行わない）．

すると

4G=<R, L, F, B, U, D, M, E, S>

さらに，$M=R^{-1}\cdot L^{-1}$，$E=D^{-1}\cdot U^{-1}$，$S=F^{-1}\cdot B^{-1}$となるため，

4G=<R, L, F, B, U, D>

これは 3G と同型となる．つまり，中央に挟まれた面の操作について，一回転方向について 2 つ面があるが，この面を一つの面として考え，分けて操作しないという制限を加えると，4G は 3G と同じ群となる．

動かすブロックの数は異なるが、この操作の制限によって区別されなくなるブロックを一つのブロックとして見れば RC{3}と全く同じものになることから想像できる．

これらのことから以下の方法が提案できる．

RC{3}が RC{4}の部分群であるから

4G=<R, L, F, B, U, D, r, l, f, b, u, d>
3G=<R,L,F,B,U,D>

4G/3G の完全代表系すべてを計算し，高々の手数に 20(RC{3}の神の数字)を足せば神の数字の上端となる．

【RC{4}の全パターン÷RC{3}全パターン】
=513354037818415700000000000000 パターン

4G と 3G の間に段階を増やせばこの計算量を減らすことができる．

4.4　プログラミング

プログラミングについては，数学計算用の SageMath，群論用の GAP，そして HTML で複数の操作後の展開図を表示するプログラミングを活用しようとした．

本研究では，時間の関係により，GAP を用いた RC{4}の神の数字の計算に努めた．

群論用関数 GAP(以後，GAP と表記する）とは，群を扱うために開発された計算機代数システムである．このシステムは C 言語を用いることで使うことができる．また，GAP 特有のプログラミング関数 GAP を使うことで数学の計算も行える．

GAP を使えば，RC 群を定義したとき，その群の位数(RC の全パターンの数)，その群に含まれる置換(具体的な操作，状態)を調べることなどができる．

GAP と Excel を用いて RC{3}の解き方を表示する文献では，表示させている現在の状態から完成状態までを計算させることによって求めている．この文献を基に RC{4}の解き方を表示するプログラムを書こうとすると一面体の識別が課題となる．

RC{4}の全てのブロックの面すべてに対して，番号を振り，その動きを群として表すと戻すまでの過程が増えるため，神の数字とはならない．この課題は RC{3}と違い一つの面につき，本来では区別されない一面体が 4 つあることで起きている．群論の計算では全ての一面体を区別しなければいけないので，群論による計算の後に工夫して解決しなければいけない．

一つの面に注目して例を挙げると下の通りである．色として注目するとこの面は完成している．しかし，一つずつの面に注目すると図 2 は完成していない．

図 1

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

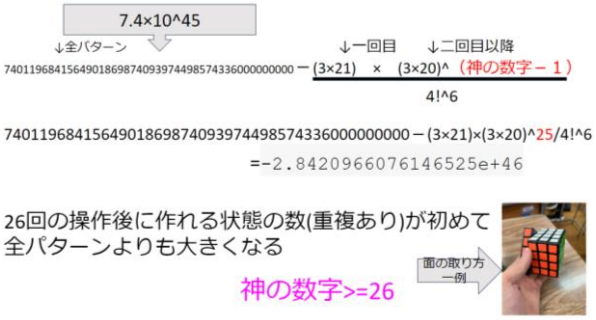
図 2

1	2	3	4
5	10	6	8
9	11	7	12
13	14	15	16

本研究では，全ての面の色の完成時を神の数字と表すため，図 2 でも完成とする．その識別をする方法は見つけられなかった．

今回の研究では計算において結果を求めるプログラミングを作成できなかった．

4.5　ほかの神の数字を求める計算



これは簡単で精度が低く，群論の考え方も用いない神の数字を求める計算である．この式は全パターンから，簡単な理屈を用いたある手数において作れる状態の数を引くものである．手数と作れる状態について，ルービックキューブでは一手と数える操作は 63 通り．これは定義した単位操作と対応する．しかし二回目以降では，一回前の操作での面の取り方と同じ面の取り方で操作をすることはなく，その場合，前の操作と合わせて一手と数える．よって 2 手目以降では面の取り方は 20 通り，操作の種類は 60 とおりとなる．加えて，同じ色の一面体は区別できないため，4 の階乗の 6 乗で割る．結果 26 回操作したときにはじめて作れる状態の総数が全パターンよりも大きくなる．これは 26 回以上の操作をすることでやっと作れる状態があることを示す，ただし一致しているわけではなく，重複が含まれることも考えられるので，神の数字は 26 手以上であるだろうことが分かる．しかし本当に 26 手以上であるかは実際に調べなければ証明とはいえず，さらにここで求めたのは神の数字の下端の目安のみで，結局この計算のアプローチでは神の数字の研究を深めることは難しいと考えられる．

5. 結論

群論の考え方をもとに RC{4}の RC 群を定義した.
群論を用いて数学的に神の数字を求める方法について考察した.
RC{3}における神の数字の研究を参考に，RC{3}との相違点と同じ点を考察，RC{4}においても神の数字の研究をするのに必要な定義，論理を構築した.
プログラミングに挑戦した.

6. 引用，参考文献・使用ソフト

[1] 高校数学の美しい物語
https://manabitimes.jp/math
[2] 物理のかぎしっぽ
https://hooktail.sub.jp/index.html
[3] 予備校のノリで学ぶ「大学の数学・物理」群論
https://www.youtube.com/watch?v=8VAx9ylEgUo
[4] ルービックキューブへの群論（論文）
https://nwuss.nara-w.ac.jp/media/sites/11/ssh06_14.pdf
[5] ルービックキューブと群（論文）
https://sitmathclub.github.io/research/pdf/2016/omiya/document/osada_s.pdf
[6] 東邦大学理学部情報科学科 2015 年度卒業論文
https://www.lab2.toho-u.ac.jp/sci/is/shirayanagi/lab/dl/2015/mizuno.pdf
[7] 群論で解き明かすルービックキューブの数理
https://www.ivis.co.jp/text/20100901.pdf
[8] Jaap's Puzzle Page
https://www.jaapsch.net/puzzles/thistle.htm
[9] Rubik's Cube Thistlethwaite Four Phase Algorithm
https://math.stackexchange.com/questions/1362471/rubiks-cube-thistlethwaite-four-phase-algorithm
[10] 数学　パズル・ゲームの広場へようこそ
https://sugakuyugi.web.fc2.com/
[11] Wikipedia-GAP（数学処理システム）
https://ja.wikipedia.org/wiki/GAP
[12] SageMath　https://www.sagemath.org/
[13] GAP　https://www.gap-system.org/
[14] CodePen　https://codepen.io/
[15] 一昨年(2022)の伊那北高校課題研究数学班報告書