ルービックキューブを
コンピュータで解く考え方

宇佐見公輔

2025-02-16 / 第32回日曜数学会

自己紹介

- 宇佐見公輔（うさみこうすけ）
- 本業はプログラマー

近況

月刊 I/O に記事をいくつか書きました

- 8〜9 月号 ： ルービックキューブの解法
- 10 月号 ： パズルキューブ スクエア 1
- 12 月号 ： 組版ソフトウェア Typst

今回の話

ルービックキューブの解法はコンピュータ計算で求めることが可能です。その考え方を、群論の言葉を使って説明します。

ルービックキューブを理解するためにわざわざ群論の言葉を使う必要はないのですが、使うとシンプルにとらえることができます。

余談

実は以前、ルービックキューブと群論の話をしました。

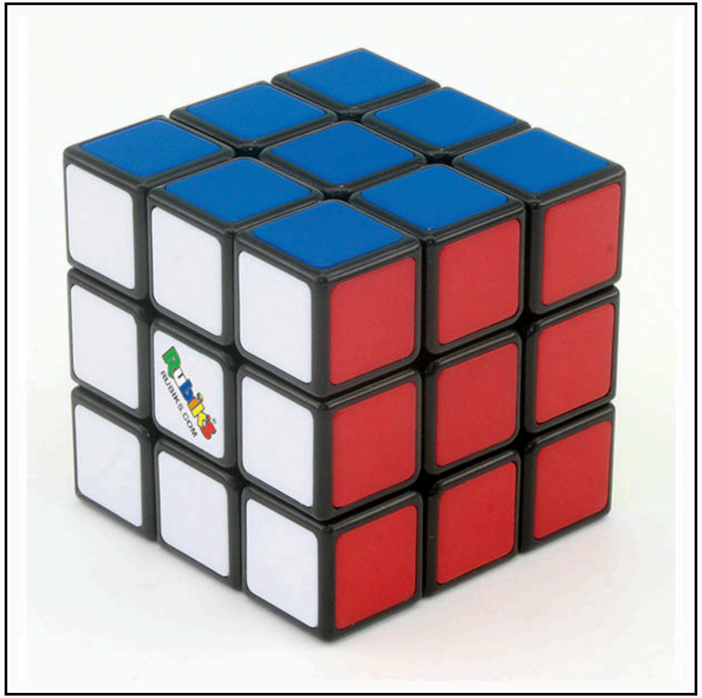
- ルービックキューブと群論
  - （第 7 回関西日曜数学友の会 / 2020-10-03）
- ルービックキューブ群を SageMath で見る
  - （第 19 回日曜数学会 / 2020-10-31）

ルービックキューブ群

ルービックキューブ

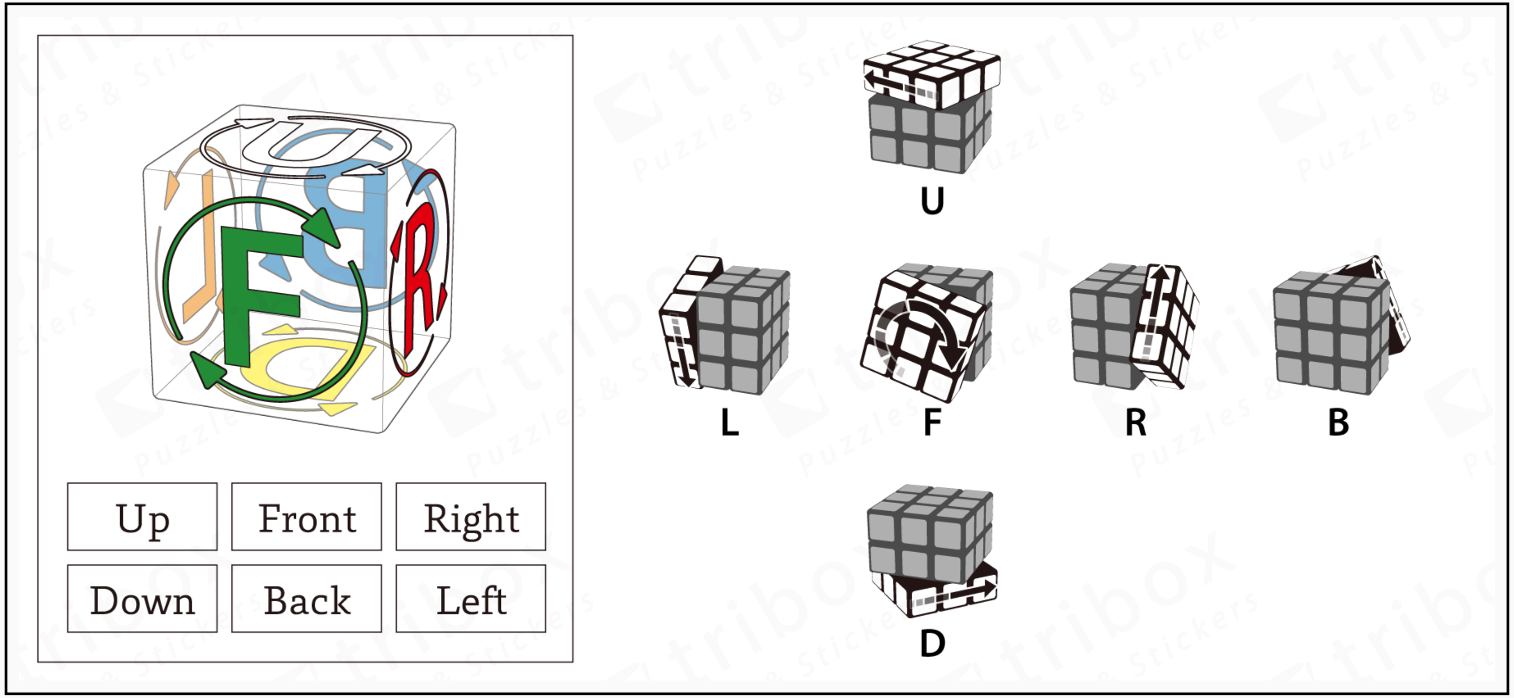
今回は標準的な 3x3x3 のルービックキューブを扱います。

ルービックキューブを解くとは、各面の色がバラバラになっている状態から、面を回転させる操作によって、完成状態（各面が 1 色）にすることです。



基本操作

U / D / F / B / R / L : 各面を時計回りに 90 度回転



（図は https://tribox.com/3x3x3/solution/notation/ より引用）

ルービックキューブ群

基本操作で生成される置換群を、ルービックキューブ群と呼びます。

$$G = \langle U, D, F, B, R, L \rangle$$

- 単位元 1 ： 完成状態
- R ： 完成状態から右面を時計回りに 90 度回転
- UR ： 右面を 90 度回転、さらに上面を 90 度回転

補足

記法の都合上、右から順に操作を適用することにしておきます。

ルービックキューブの解法

主な解法

コーナー・エッジ法

- まずコーナーだけ揃え、次にエッジを揃える
- 数学的な理屈として自然な発想

LBL（Layer by Layer）法

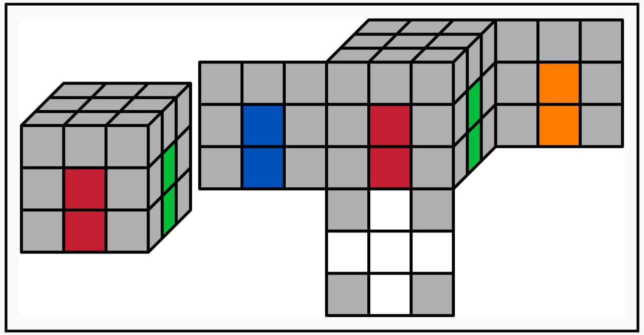
- 1 段目、2 段目、3 段目の順に揃える
- 初心者向けの手順が整備されている

CFOP 法

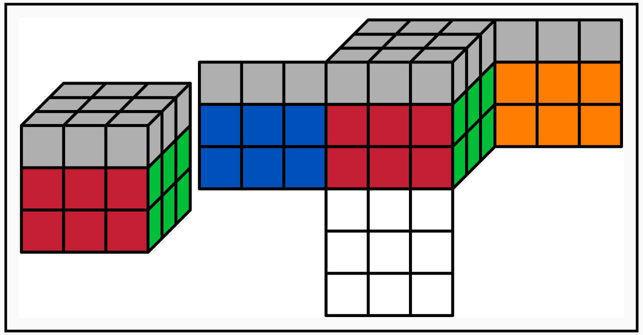
- Cross、F2L、OLL、PLL の 4 段階で揃える
- スピード競技向けにパターン化されている

CFOP 法

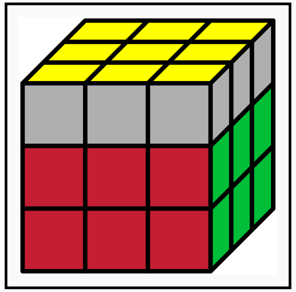
Cross



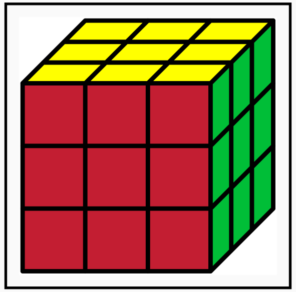
F2L (First 2 Layers)



OLL (Orientation of the Last Layer)



PLL (Permutation of the Last Layer)



共通する考え方

- まず、一部分を完成状態にする
- 次に、完成部分を崩さずに、他の部分を完成状態にする
- これを何段階かにわけて行う

補足

- 「完成部分を崩さずに」
  - 一時的には崩れても、あとで元に戻るような手順を使う
- 「何段階かにわけて」
  - たとえば CFOP 法なら 4 段階にわけている

解法を群論の言葉で

部分群で考える

- 解法の考え方「完成部分を崩さずに」

↓

- 完成部分を崩さない操作手順だけを使う

↓

- 完成部分を崩さない操作がなす部分群で考える

部分群の系列を考える

- 解法の考え方「何段階かにわけて」

↓

- 部分群の系列を考える

$$G = G_0 \supset G_1 \supset G_2 \supset \ldots \supset G_n = \{1\}$$

剰余類を使う

$$G = G_0 \supset G_1 \supset G_2 \supset \ldots \supset G_n = \{1\}$$

G_k から G_{k+1} への移行は、左剰余集合 G_k/G_{k+1} を考えます。

- $g_k \in G_k$ を含む左剰余類 xG_{k+1} をとる
- $g_{k+1} \coloneqq x^{-1}g_k$ とすると、$g_{k+1} \in G_{k+1}$ となる

CFOP 法を群論の言葉で

$$G = G_0 \supset G_1 \supset G_2 \supset G_3 \supset G_4 = \{1\}$$

- G_1 ： Cross を崩さない操作がなす部分群
- G_2 ： F2L を崩さない操作がなす部分群
- G_3 ： OLL を崩さない操作がなす部分群
- G_4 ： PLL を崩さない操作がなす部分群（完成状態のみ）
- Cross を揃える ： G から G_1 へ（G/G_1 の元を探す）
- F2L を揃える ： G_1 から G_2 へ（G_1/G_2 の元を探す）
- OLL を揃える ： G_2 から G_3 へ（G_2/G_3 の元を探す）
- PLL を揃える ： G_3 から G_4 へ（$G_3/G_4 \simeq G_3$ の元を探す）

コンピュータで解く考え方

部分群の系列を考える

人間が解くための解法では、それぞれの段階の完成状態が認識しやすいものでないと困ります。（Cross 状態、F2L 状態、など）

しかし、コンピュータで解く場合はそのような制約はなく、自由に部分群の系列を考えることができます。

補足

剰余類を使うので、単なる部分集合でなく部分群を考えます。

コシエンバ法（Two Phase Algorithm）

$$G = G_0 \supset G_1 \supset G_2 = \{1\}$$

$$G_1 := \langle U, D, F^2, B^2, R^2, L^2 \rangle$$

- G/G_1 の元を探す
- $G_1/G_2 \simeq G_1$ の元を探す

G_1 は次のような特徴があります。

- 各ピースの向きを変えない
- U 面や D 面を側面に移動できない

（https://kociemba.org/twophase.htm に解説があります）

コシエンバ法の手数

- G/G_1 の剰余類の代表元は、高々 12 手である
- G_1 の元は、高々 18 手である
- この 2 つの群の元を探すことで、高々 29 手で解く手順が見つかる

手数が少なめで、それぞれの元の探索は、アルゴリズムの工夫により実用的な時間で行えます。（今回はこのあたりの話は省略）

まとめ

群論の言葉は便利

- ルービックキューブの解法は、群論の言葉を使って説明できる
- 必要な知識は、部分群や剰余類といった基本的なものだけ