FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA

LUCA ESCOPELLI

CUBO DE RUBIK COMO UM GRUPO ALGÉBRICO

Rio de Janeiro
2023

LUCA ESCOPELLI

CUBO DE RUBIK COMO UM GRUPO ALGÉBRICO

Trabalho de conclusão de curso apresentada para a Escola de Matemática Aplicada (FGV/EMAp) como requisito para o grau de bacharel em Matemática Aplicada.

Área de estudo: álgebra.

Orientador: Moacyr Alvim

Rio de Janeiro

2023

Agradecimentos

Lembre de agradecer a quem te apoiou, como, por exemplo, orientador, família, agência de fomento, professores conselheiros.

"Se eu vi mais longe, foi por estar sobre ombros de gigantes."
Isaac Newton

Resumo

Segundo a o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. (. . .) As palavras-chave devem figurar logo abaixo do resumo, antecedidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto. Deve ser redigido na terceira pessoa do singular e quanto a sua extensão, o resumo deve ter de 150 a 500 palavras.

Palavras-chave: latex. abntex. editoração de texto.

Abstract

É a tradução do resumo para o inglês (Abstract), com a finalidade de facilitar a divulgação do trabalho em nível internacional.

Keywords: latex. abntex. editoração de texto.

Lista de ilustrações

1 Introdução

O cubo de Rubik, também conhecido como cubo mágico, é um brinquedo inventado em 1974 por Ernő Rubik, um professor húngaro de arquitetura. O objeto foi criado com a intenção de desenvolver a capacidade de visualização espacial em seus alunos. Ao começar a brincar com sua nova invenção, o professor rapidamente percebeu a complexidade que o brinquedo poderia tomar com apenas alguns movimentos, tanto que levou mais de 30 dias para conseguir solucionar o puzzle. Ainda antes da invenção de Rubik, Larry D. Nichols já havia desenvolvido um objeto semelhante a uma versão 2x2x2 do cubo, com uma estrutura utilizando imãs para segurar as peças, porém não conseguiu obter a mesma popularidade da versão 3x3x3.

Desde sua invenção, o cubo foi um objeto de admiração por todas as pessoas que o conhecem. Por se tratar de um objeto com uma mecânica simples, mas com uma enorme complexidade, se tornou um dos brinquedos mais populares de todo o mundo. Esse fascínio pelo "não tão simples" cubo foi o que motivou o desenvolvimento desse estudo em relação à invenção de Rubik e suas variações.

O presente trabalho utiliza das ferramentas da teoria dos grupos, dentro do estudo da álgebra, aplicadas ao cubo para encontrar propriedades interessantes do objeto. A teoria dos grupos introduz o conceito de um grupo matemático, que consiste em um conjunto de elementos juntos de uma operação, de tal forma que a operação seja associativa, exista um elemento neutro e todo elemento possua seu inverso. Com isso em mente, é possível trabalhar com o cubo como um grupo, em que cada estado de embaralhamento é um elemento do conjunto e as operações são os movimentos de rotação das faces.

Uma das interessantes propriedades do cubo e objeto de estudo desse trabalho é o chamado "número de Deus". Esse número se refere ao número mínimo de movimentos suficiente para resolver qualquer estado de embaralhamento do cubo, i.e. independente do estado inicial do cubo, existe uma sequência de movimentos de tamanho menor ou igual à esse valor que resolve o puzzle. Para o cubo tradicional (3x3x3) esse número foi provado ser 20 [1]. Será apresentado como foi feito esse processo de descoberta e como técnicas semelhantes podem ser aplicadas na variação mais simples, o cubo de Nichols (2x2x2,) para descobrir seu Número de Deus, que é o objetivo principal desse estudo.

O trabalho está dividido da seguinte forma:

Seção 2 - Introdução à teoria de grupos: O conteúdo em questão sera apresentado, contemplando suas definições, propriedades e os principais atributos que serão utilizados para o problema principal.

Seção 3 - Análise de estudos anteriores: Será feita uma passagem pelos

estudos anteriores relacionados a esse conteúdo para compreender as técnicas utilisadas e dificuldades encontradas.

Seção 4 - Apresentação dos experimentos: As técnicas e conteúdos aprensentados pelas seções anteriores serão utilizadas em experimentos para conseguir o valor do Número de Deus para o Cubo 2x2x2.

Seção 5 - Conclusão: Será apresentado os resultados obtidos pelo estudo além de desafios encontrados e possíveis futuros avanços no assunto.

2 Introdução à teoria de grupos

2.1 Definição

Um grupo é definido como um conjunto não-vazio (G) junto de uma operação binária ($\cdot$) que leva dois elementos de G a outro elemento do grupo e que respeita as 3 propriedades mostradas a seguir. Podemos escrever o grupo G com a notação (G, $\cdot$) para indicar que o grupo é formado pelo conjunto G e pela operação $\cdot$.

Um grupo sempre deve respeitar as propriedades de associatividade, existência de elemento neutro e existência de inverso.

Associatividade: Sejam a, b e c elementos quaisquer de G, devemos ter que:

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad \forall a, b, c \in G$$

Existência de elemento neutro: Todo grupo deve possuir um elemento neutro (e) que leve qualquer elemento nele mesmo, assim temos:

$$a \cdot e = e \cdot a = a \quad \forall a \in G$$

Existência de elemento inverso: Para todo elemento a de um grupo deve existir no mesmo grupo o elemento inverso a^{-1} que leve-o ao elemento neutro, temos então:

$$\forall a \in G, \quad \exists a^{-1} \in G$$
$$a \cdot a^{-1} = a^{-1} \cdot a = e$$

2.1.1 Exemplos

1. Grupo dos inteiros com a operação de soma ($\mathbb{Z}$, +):

$$a + (b + c) = (a + b) + c \quad \forall a, b, c \in \mathbb{Z}$$
$$0 = e, \quad a + 0 = 0 + a = a \quad \forall a \in \mathbb{Z}$$
$$a + (-a) = (-a) + a = 0 \quad \forall a \in \mathbb{Z}$$

2. Grupo dos racionais com a operação de soma $(\mathbb{Q}, +)$:

$$a + (b + c) = (a + b) + c \quad \forall a, b, c \in \mathbb{Q}$$
$$0 = e, \quad a + 0 = 0 + a = a \quad \forall a \in \mathbb{Q}$$
$$a + (-a) = (-a) + a = 0 \quad \forall a \in \mathbb{Q}$$

3. Grupo dos reais não nulos com a operação de multiplicação $(\mathbb{R}^*, \cdot)$:

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c \quad \forall a, b, c \in \mathbb{R}^*$$
$$1 = e, \quad a \cdot 1 = 1 \cdot a = a \quad \forall a \in \mathbb{R}^*$$
$$a \cdot a^{-1} = a^{-1} \cdot a = 0 \quad \forall a \in \mathbb{R}^*$$

Note que nesse caso não podemos incluir o 0 pois não possui inverso.

Apesar dos exemplos acima serem compostos por conjuntos de infinitos elementos, isso não é necessário, podemos ter grupos de qualquer tamanho, veja:

4. Grupo trivial $(\{0\}, +)$:

$$0 + (0 + 0) = (0 + 0) + 0 = 0$$
$$0 = e, \quad 0 + 0 = 0 + 0 = 0$$
$$0 + 0 = 0 + 0 = 0$$

Note que esse é um grupo de um conjunto com apenas 1 elemento. Além disso, percebemos que para qualquer operação sempre é possível criar um grupo trivial com o conjunto sendo apenas o elemento nulo da respectiva operação.

5. Grupo do anel das classes de congruência módulo 2 com a operação de soma $(\mathbb{Z}_2, +)$:

$$a + (b + c) = (a + b) + c \quad \forall a, b, c \in \mathbb{Z}_2$$
$$0 = e, \quad a + 0 = 0 + a = a \quad \forall a \in \mathbb{Z}_2$$
$$a + (-a) = (-a) + a = 0 \quad \forall a \in \mathbb{Z}_2$$

Note que nesse exemplo o conjunto possui apenas 2 elementos (0 e 1) e esse caso pode ser estendido para qualquer valor de congruência, indicando que o conjunto de um grupo pode ser de qualquer tamanho.

6. Grupo do anel das classes não nulas de congruência módulo p, sendo p primo, com a operação de multiplicação $(\mathbb{Z}_p^*, \cdot)$:

Multiplicação é associativa.

1 é elemento neutro da multiplicação, então $1 = e$ nesse caso.

Pelo pequeno teorema de Fermat $a^{p-1} \equiv 1 \pmod{p}$, portanto $a \cdot a^{p-2} \equiv 1 \pmod{p}$ $\implies a^{p-2} \equiv a^{-1}$

2.2 Subgrupos

Dado um grupo G, chamamos H de um subgrupo de G (indicamos por $H \leq G$) se H está contido em G ($H \subseteq G$) e H é um grupo com a mesma operação associada a G.

Como H deve ser um grupo também, é necessário que o elemento neutro esteja em H ($e \in H$). Além disso, como G é subconjunto de G e é um grupo, temos que G é subgrupo de G. Assim, podemos afirmar que todo grupo possui 2 subgrupos triviais, apenas o elemento neutro e o próprio grupo.

$$\{e\} \leq G \quad \forall G$$
$$G \leq G \quad \forall G$$

Chamamos de subgrupos próprios (ou não-triviais) todos os subgrupos diferentes dos mecionados acima.

Temos que um subconjunto H de G é subgrupo se e somente se:

- O elemento neutro pertence a H ($e \in H$)
- A operação de dois elementos de H leva a um elemento em H ($ab \in H \quad \forall a, b \in H$)
- O inverso de qualquer elemento de H está em H ($a^{-1} \in H \quad \forall a \in H$)

Note que a associatividade em H vem diretamente da associatividade em G.

2.2.1 Exemplos

O conjunto dos inteiros é subconjunto dos racionais. Como os dois conjuntos formam, individualmente, grupos com a operação de soma, podemos falar que os inteiros são um subgrupo dos racionais nessa operação. Além disso, ambos são subgrupos dos reais, que por sua vez são subgrupos dos complexos.

O conjunto definido por $n\mathbb{Z} = \{nx | x \in \mathbb{Z}\},\ n \in \mathbb{Z}$ é subgrupo dos inteiros com a operação de soma, veja:

$$n0 = 0$$

$$na + nb = n(a + b)$$

$$n(-a) = -na$$

Percebemos acima que as 3 propriedades são satisfeitas.

Um detalhe importante a se notar, sabemos que o anel das classes de congruência módulo 2 (exemplo 5 de grupo) é um subconjunto dos inteiros, no entanto não é considerado um subgrupo pois as operações em questão são diferentes. Note que em um caso o inverso de 1 é -1 e no outro é o próprio 1.

Um exemplo de subgrupo envolvendo anéis pode ser obtido considerando as classes módulo 4 ($\{0, 1, 2, 3\}$) com a operação de soma, em que podemos tomar o subgrupo próprio ($\{0, 2\}$), pois:

$$0 = e$$

$$2 + 2 \equiv 0 \quad (mod\ 4)$$

Como vemos, o subconjunto possui o elemento neutro, todas operações de elementos do subconjunto levam a outro elemento do subconjunto e todos elementos possuem inverso dentro do subconjunto.

2.3 Conjunto gerador de um grupo

Definimos como conjunto gerador de um grupo um subconjunto S tal que todos elementos do grupo G podem ser obtidos por uma combinação finita dos elementos de S, ou seus inversos, sob a operação do grupo. Escrevemos então $G = \langle S \rangle$.

No caso particular de S ser um conjunto unitário ($S = \{s\}$) temos que:

$$G = \langle S \rangle = \langle \{s\} \rangle = \{s^n | n \in \mathbb{Z}\}$$

Nesse caso, chamamos G de grupo cíclico e denotamos apenas por $G = \langle s \rangle$.

Para os casos gerais, temos que:

$$\langle S \rangle = \{s_1 s_2 \cdots s_n | n \in \mathbb{N}, s_i \in S \cup S^{-1}\}$$

Sendo S^{-1} o conjunto dos inversos de S. Note que, como consequência direta da definição, $\langle S \rangle = \langle S^{-1} \rangle$.

2.3.1 Exemplos

- $\mathbb{Z} = \langle 1 \rangle = \langle -1 \rangle$

  O grupo dos inteiros com a operação de soma é ciclíco gerado pelo elemento 1 (ou pelo -1).

- $n\mathbb{Z} = \langle n \rangle = \langle -n \rangle$

  O grupo dos múltiplos de n com a operação de soma é ciclíco gerado pelo elemento n (ou pelo -n).

- Potências de $2 = \langle 2 \rangle = \langle 2^{-1} \rangle$

  O grupo das potências de 2 é gerado pelo 2 considerando a operação de multiplicação.

- $\langle \{2, 3\} \rangle$

  O grupo do produto entre potências de 2 e potências de 3 é gerado pelo conjunto $\{2, 3\}$ junto com a operação de multiplicação.

Esse conceito fornece uma ferramenta prática muito importante para criação de grupos interessantes para determinados problemas. Visto que, dado um conjunto e uma operação que faça sentido, podemos gerar um grupo utilizando a ideia de conjunto gerador sem a preocupação de conhecer todos elementos individualmente. Mais adiante, veremos a aplicação dessa ideia no estudo em questão.

2.4 Classes laterais

Dado um subgrupo H de G ($H \leq G$), definimos as classes laterais à esquerda ou à direita de H, respectivamente, como:

Dado $x \in G$:

$$xH = \{x \cdot h | h \in H\}$$

$$Hx = \{h \cdot x | h \in H\}$$

As classes laterais são subconjuntos de G, no entanto, não são subgrupos, pois não possuem o elemento neutro. Podemos perceber isso, pois $x, h \in G \implies xh \in G \implies xH \subset G$, porém $x \notin H \implies xh \notin H \implies e \notin xH$, logo xH não é subgrupo (para a classe à direita é analogo). Um detalhe importante é de que consideramos $x \notin H$, isso pode ser utilizado, visto que, caso $x \in H$ teremos que $xH = H$ e não faz sentido considerarmos a classe lateral nesse caso.

Além disso, forma-se um bijeção natural entre H e xH. Note que $x \in G \implies x^{-1} \in G$ logo $xh_1 = xh_2 \implies x^{-1}xh_1 = x^{-1}xh_2 \implies h_1 = h_2$, portanto a cardinalidade de qualquer classe lateral é a mesma e é igual à cardinalidade de H.

Temos também que dados $x, y \in G \backslash H$ com $x \notin yH$ então as classes laterais xH e yH não possuem elementos em comum. Vamos provar isso por contradição.

$$h_1, h_2 \in H, \quad xh_1 = yh_2 \implies x = yh_2h_1^{-1} \implies x \in yH$$

Absurdo, logo $xH \cap yH = \emptyset$.

Além disso, se $x \in yH$ então $xH = yH$, pois $\exists h_1$ tal que $x = yh_1 \implies xH = yh_1H = yH$.

Como todo elemento de G pode gerar sua classe lateral, e acabamos de verificar que todas as classes laterais são disjuntas (desconsiderando os casos de classes equivalentes) e possuem a mesma cardinalidade (igual à cardinalidade de H), temos que as classes laterais formam uma partição muito interessante do grupo G original. Adicionalmente, como todo subgrupo possui classes laterais, temos para cada subgrupo uma partição diferente do grupo principal.

Essa propriedade é muito interessante para o nosso problema, pois permite a divisão do desafio em desafios menores e disjuntos. Veremos mais adiante como isso é feito na prática.

3 Estudos anteriores

4 Apresentação dos experimentos

Nosso objeto de estudo é o cubo mágico de dimensões 2x2x2. Como a ideia do trabalho é aplicar a teoria de grupos ao cubo, nosso primeiro desafio é transformar o objeto em um grupo algébrico.

Antes de fazer isso, vamos entender como é a estrutura dessa versão menor do brinquedo. O 2x2x2 é formado apenas por 8 cubinhos menores, cada um possuindo 3 cores e dispostos nos vértices do cubo. Fazendo uma comparação com o 3x3x3, é como se só tivéssemos as peças de canto do cubo. Dado esse formato, temos uma diferença bem relevante que é a de não possuir peças fixas, portanto, não há definição prévia de qual face deverá ser qual cor quando o cubo estiver resolvido. Com isso, podemos "fixar" uma peça para simplificar o problema e evitar casos que são iguais exceto por uma operação de rotação do cubo. Essa fixação será feita tomando uma peça qualquer e mantendo sempre ela no lugar correto, ou seja, antes de executar movimentos no cubo, vamos rotacioná-lo para colocar a peça no lugar estabelecido sem executar movimentos. Em seguida, vamos aplicar movimentos no cubo que não alterem a posição dessa peça (verificaremos adiante como isso é feito).

Para fazer isso, vamos utilizar a mesma ideia dos estudos anteriores. Os elementos do grupo são representados por uma sequência de movimentos e cada uma dessas sequências gera um embaralhamento no cubo a partir do estado inicial (resolvido). No nosso grupo, o elemento neutro é representado pela não execução de movimentos, ou seja, manter o cubo resolvido. Como notação, vamos utilizar I para representar esse caso (uma referência à matriz identidade). Da mesma forma que já vimos para o cubo tradicional, os movimentos são associativos e cada movimento possui um inverso (girar a mesma face no sentido oposto e o mesmo número de vezes).

Essa ideia de transformar o cubo em um grupo é exatamente a mesma que apresentamos para a versão 3x3x3, no entanto, o que vamos diferenciar aqui serão os movimentos. Já apresentamos na versão tradicional que os movimentos possíveis são relacionados às faces R, L, F, B, U, D. Porém, no cubo 2x2x2, girar a face da direita no sentido horário gera o mesmo resultado de girar a face esquerda no anti-horário, exceto por uma rotação do cubo. O mesmo ocorre para os outros pares de faces opostas e outros sentidos de rotação de face. Com isso, podemos simplificar nosso caso e utilizar somente os movimentos R, F e U, i.e. o grupo é gerado apenas por esses movimentos $G_2 = \langle R, F, U \rangle$. Note que, com esses movimentos, a peça do canto traseiro, inferior, esquerdo nunca será movida, portanto essa será a peça que teremos fixada, como mencionado anteriormente.

4.1 Cardinalidade

Uma das principais informações que devemos obter para nosso estudo é a cardinalidade do grupo. Para calcular esse valor vamos verificar quantos estados do cubo podemos obter.

Vamos iniciar pensando nas posições das peças. Temos uma peça fixa, porém todas as outras podem permutar entre si formando qualquer possível permutação, logo temos 7! possíveis permutações entre as peças.

Agora pensando na rotação das peças, cada peça possui 3 possíveis estados de rotação, com exceção da peça fixa. No entanto, ao definir a rotação de 7 peças, a última rotação estará definida (o cenário de o cubo todo estar resolvido exceto por uma peça rotacionada é impossível de ser obtido com os movimentos permitidos). Portanto, temos 6 peças que podem ter qualquer umas das 3 rotações cada. Gerando assim 3^6 diferentes estados de rotação para uma mesma posição das peças.

Juntando as duas ideias, chegamos a conclusão de que o cubo possui $7! \cdot 3^6 = 3.674.160$ possíveis embaralhamentos. Logo, acabamos de encontrar a cardinalidade do grupo $|G_2| = 3.674.160$

4.2 Primeira cota inferior

Agora que temos a cardinalidade do grupo, podemos pensar em uma maneira fácil de obter uma cota inferior para o desejado número de Deus.

A primeira ideia que temos é a de quantos cenários podemos obter com cada quantidade de movimentos. Sabemos que com 0 movimentos só temos o cenário trivial I. Com 1 movimento temos os seguintes cenários: R, 2R, R', U, 2U, U', F, 2F, F', totalizando 9 cenários com 1 movimento e 10 cenários com até 1 movimento. Esse valor claramente é menor que a cardinalidade do grupo, portanto sabemos que é impossível resolver todos os cenários do cubo com apenas 1 movimento, sendo essa uma cota inferior trivial.

Calcular o verdadeiro número de estados possíveis para um número maior de movimentos é algo um pouco complicado, pois teríamos que sempre conferir a possibilidade de estados equivalentes. Porém, ignorando esse detalhe, podemos obter cotas superiores para esses valores. E essas cotas superiores são muito interessantes, pois, enquanto a cota superior for menor do que a cardinalidade do grupo, temos a certeza de que nem todos estados são possíveis de se obter com aquele número de movimentos, portanto o número de Deus tem que ser um valor superior.

Exemplo, para calcular o número de estados que podemos obter com até 2 movimentos vamos seguir o seguinte raciocínio. Para o primeiro movimento, podemos utilizar qualquer um dos 9 movimentos apresentados anteriormente. Para o segundo movimento,

devemos mover uma face diferente da anterior, portanto temos apenas 6 possíveis movimentos. Logo, temos $9 \cdot 6 = 54$ como cota superior de estados possíveis de se obter com 2 movimentos, e $54 + 9 + 1 = 64$ considerando até 2 movimentos.

Essa ideia pode ser aplicada para qualquer quantidade de movimentos. Podemos definir $S_0 = 1$ e $S_n = 9 \cdot 6^{n-1} \quad \forall n \geq 1$ e essa será uma cota superior do número de cenários possíveis de se obter com n movimentos. Além disso, podemos definir $T_n = \sum_{i=0}^{n} S_i$ e essa será uma cota superior da quantidade de estados que se pode obter com até n movimentos.

Podemos buscar o maior valor de n tal que $T_n < |G_2|$ e esse valor será nossa primeira cota inferior não trivial. Temos então que:

$$T_n = \sum_{i=0}^{n} S_i = 1 + \sum_{i=1}^{n} 9 \cdot 6^{i-1} = 1 + 9 \sum_{i=0}^{n-1} 6^i$$

$$T_n = 1 + 9 \cdot \frac{6^n - 1}{5}$$

Logo, queremos achar o maior n tal que:

$$1 + 9 \cdot \frac{6^n - 1}{5} < 3.674.160$$

$$6^n - 1 < 3.674.159 \cdot \frac{9}{5}$$

$$6^n < 6613487,2$$

Como 6^n é inteiro, temos:

$$6^n \leq 6613487$$

$$n \leq log_6(6613487) < 8,77$$

Portanto, com até 8 movimentos, sabemos que não é possível de se resolver todos os embaralhamentos do cubo. Logo, temos nossa primeira cota inferior como sendo 9.

Nossa primeira cota superior é de 20, pois como vimos, é o número de Deus para o 3x3x3. E podemos considerar o cubo 2x2x2 como uma versão do tradicional em que só consideramos os cantos, logo uma solução do 3x3x3 implica em uma solução do caso análogo no 2x2x2.

4.3 Orientação das peças

Para nossas próximas análises será bom ter bem definido o significado da orientação de uma peça, portanto vamos apresentar agora.

Sabemos que o cubo é composto de 8 peças de canto, sendo 7 delas móveis. Iremos utilizar a combinação de cores mais usual do cubo, em que temos 3 pares de cores opostas: Amarelo e Branco; Vermelho e Laranja; e Azul e Verde. Além disso, vamos deixar o cubo posicionado de tal forma que a face Amarela fique em cima, a face Azul na frente e a face Vermelha na direita. Dessa forma, temos bem definido a posição do cubo resolvido e também, dado o que já foi apresentado, temos que a peça fixa será a peça de cores Branco, Verde e Laranja.

Cada peça do cubo possui 3 cores, sendo cada uma delas vinda de um dos 3 pares apresentados acima. Com isso em mente, podemos definir a orientação de uma peça como correta caso sua cor Amarela ou Branca esteja na face superior ou inferior do subo, independente da posição do cubo. Caso ela esteja incorreta, temos dois cenários: está girada no sentido horário de seu estado correto, ou, está girada no sentido anti-horário de seu estado correto. Para cada um desses 3 cenários vamos atribuir um valor pensando na congruência módulo 3.

Para a orientação correta é natural que demos o valor 0 para sua identificação. Caso esteja girada no sentido horário, vamos atribuir o valor 1. Caso esteja girada no sentido anti-horário, vamos atribuir o valor 2 (ou ainda -1).

Dadas essas definições, podemos avaliar a orientação de qualquer peça do cubo em qualquer embaralhamento. Essa identificação irá no auxiliar para as próximas etapas do estudo.

4.4 Subgrupo interessante

Dadas as definições já apresentadas, vamos introduzir um subgrupo interessante para o nosso estudo. A ideia do nosso subgrupo é de concentrar todos os cenários em que todas as peças estão orientadas corretamente.

Para isso, vamos notar que, partindo do cubo resolvido, qualquer movimento duplo apenas troca algumas cores com seu respectivo par. Dessa forma, tanto a face superior quanto inferior do cubo só terá as cores amarelas e brancas, logo todas as peças do cubo se mantém orientadas corretamente.

Além disso, o movimento U não altera a face superior nem inferior, portanto também mantém a orientação de todas as peças do cubo.

Com isso, temos que o subgrupo gerado por $\langle R2, F2, U\rangle$ terá apenas estados em que todas suas peças estão orientadas corretamente. Veremos posteriormente que esse

subgrupo realmente contém todos os estados em que todas as peças estão com a orientação correta.

Antes de verificar que o subgrupo acima realmente contém todos os cenários de orientação correta, vamos pensar nesse conjunto pela sua definição. Dado que todas as peças estão orientadas corretamente, basta sabermos as posições delas para termos o estado do cubo completo. Como não há restrição em relação às suas posições, temos que qualquer permutação das 7 peças móveis é possível. Portanto, esse conjunto possui $7! = 5040$ elementos.

Vamos então provar que esse conjunto e o subgrupo são iguais.

4.4.1 Prova

Para provar que são iguais, vamos demonstrar que por meio dos movimentos do subgrupo conseguimos chegar em qualquer elemento do conjunto, dessa forma eles serão equivalentes.

Para isso, vamos verificar que quaisquer duas peças móveis do cubo podem ser permutadas utilizando movimentos do subgrupo. Se isso for verdade, então podemos obter qualquer permutação de peças com movimentos do subgrupo e, portanto, o subgrupo e o conjunto são iguais.

Primeiramente, vemos que podemos permutar as duas peças superiores à direita por meio da sequência de movimentos F2 R2 U F2 U' R2 U R2 U' F2 U, gerando o estado indicado em 1:

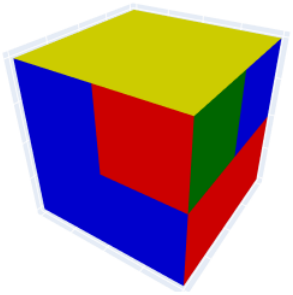

Figura 1 – Permutação

Agora vamos verificar que essa permutação pode ser realizada entre quaisquer peças do cubo. Para isso, vamos chamar a sequência de movimentos acima de T, representando a troca de duas peças. Em seguida, vamos desenvolver um método para criação de um

movimento P de preparação do cubo, que consiste em colocar as peças que desejamos permutar como um par na posição superior à direita. Tendo esses movimentos bem definidos, aplicamos a sequência $P\ T\ P^{-1}$, que prepara a cubo, realiza a troca e volta todos os movimentos de preparação para que as outras peças não sejam afetadas.

Para criar o movimento P vamos primeiro selecionar as duas peças que queremos permutar. Nosso primeiro objetivo será deixar as peças lado a lado como um par. Assim que esse par estiver formado, levamos o par para a camada superior. Estando na camada superior, utilizamos movimentos U até o par estar exclusivamente na direita.

Para a primeira etapa, deixamos uma peça na camada superior e a outra na inferior. Em seguida vemos onde que a peça da camada inferior iria ao subir com um movimento F2 ou R2. Então, posicionamos a peça da camada de cima com movimentos U de forma que ao realizar a subida eles formem um par. Assim, temos a primeira etapa concluída. Por meio desse processo, a segunda etapa já estará realizada, tendo em vista que formamos o par na camada superior. Portanto ela só será necessária caso originalmente o par esteja formado na camada inferior, no qual basta utilizarmos movimento F2 ou R2 para subir com o par inteiro. A última etapa é simplesmente colocar o par na direita com movimentos U, assim teremos posicionado nosso par da forma desejada.

Dessa forma, temos um processo bem claro de criação do movimento P junto com a definição dos movimentos a serem executados para permutar duas peças. Assim, conseguimos provar que sempre é possível permutar duas peças com movimentos do subgrupo e, portanto, ele é equivalente ao conjunto indicado.

4.4.2 Classes laterais

Acabamos de provar que o subgrupo gerado por $\langle R2, F2, U\rangle$ tem tamanho 5040 e mantém a orientação de suas peças conforme nossa definição. Com isso, podemos obter suas classes laterais. Pelos nossos conhecimentos de teoria de grupos, sabemos que teremos $3.674.160/5.040 = 729$ classes laterais para esse subgrupo.

Dadas as nossas definições para o subgrupo, temos que as classes laterais corresponderão à todas as possíveis orientações de suas peças. Para obter um elemento qualquer de cada classe lateral, podemos tomar o cubo em que todas as peças estão na sua posição correta, porém suas orientações variam. Como temos que existem $3^6 = 729$ orientações possíveis dada uma posição, confirmamos que os valores estão de acordo com as definições.

Dessa forma, temos bem definido qual o nosso subgrupo e quais são suas classes laterais. Portanto, podemos utilizar essas informações para auxiliar nosso trabalho.

4.5 Implementação computacional

Para poder testar nossas ideias e verificar os valores, tivemos que realisar um trabalho computacional.

Nossa implementação se baseou em criar uma classe em python chamada “cube” que representa o cubo. Dentro dessa classe temos definições claras do estado atual do cubo, além dos possíveis movimentos e como eles afetam a posição das peças. Adicionalmente, implementamos uma função que cria um “plot” 3D, permitindo uma visualização do cubo (Essa função foi responsável pela criação de algumas imagens presentes nesse trabalho).

A classe foi estrturada pensando em uma sequência definida de peças que indica qual peça está naquela posição e seu estado de rotação. O estado de rotação foi estabelecido com base na definição já apresentada. Para definir as peças, fizemos uma numeração com base no cubo resolvido. A numeração foi realizada com base no cubo resolvido, sendo a face superior numerada de 1 a 4 com base no sentido horário partindo do canto superior esquerdo olhando de cima. Em seguida, numeramos as peças inferiores de 5 a 7 com a mesma lógica. De modo que no final as peças foram numeradas conforme imagem 2 (a peça de número 7 se encontra na parte de trás do cubo e não pode ser visualizada na imagem):

Figura 2 – Identificação das peças

Dada essa identificação das peças, criamos a classe por meio do código abaixo. A ideia é de que o cubo é uma lista da ordem de seus cantos, e cada canto é uma lista de tamanho 2 com a identificação de qual canto ele é e qual a sua rotação. Seguindo essa ideia o cubo resolvido seria representado por [[1,0], [2,0], [3,0], [4,0], [5,0], [6,0], [7,0]], que é o estado de inicialização padrão da classe.

```
class cube:
    def __init__(self, corners_list = []):

        if len(corners_list) == 0:
            self.corners = [
                [1,0],
                [2,0],
                [3,0],
                [4,0],
                [5,0],
                [6,0],
                [7,0]
            ]

        else:
            self._check_corners(corners_list)
            self.corners = corners_list
```

Dentro da classe, a função apresentada "check_corners" apenas realiza a verificação se os valores indicados geram um cenário possível para o cubo. Além disso, temos implementado as funções que realizam os movimentos possíveis, elas são responsáveis por atualizar a lista dos cantos conforme a alteração causada pelo movimento.

Tendo a classe bem implementadoa, conseguimos realizar nossos primeiros testes em relação aos possíveis estados do cubo.

É importante pontuar que, para calcular o número mínimo de movimentos para resolver um estado do cubo, utilizamos o processo inverso. Nesse processo, partimos do cubo resolvido e realizamos uma sequência de movimentos para chegar em um novo estado. Dessa forma, sabemos que para resolver esse estado, basta realizarmos os movimentos inversos aos utilizados para chegar nele. Portanto, nosso método de busca foi realizado da seguinte forma:

Partimos do cubo resolvido e realizamos cada movimento possível nele, gerando todos os estados de cubo resolvíveis em 1 movimento. Em seguida, para cada estado de 1 movimento, realizamos todos movimentos possíveis para gerar novos cenários. Esses novos cenários são comparados com todos outros já existentes para garantir que são novos e não repetidos. Se realmente forem novos, eles são adicionados à lista de estados resolvíveis em 2 movimentos. Seguimos com esse procedimento até que todos cenários gerados sejam repetidos, assim garantimos a passsagem por todos cenários do estudo.

4.5.1 Análise do subgrupo

Nossa primeira análise foi em relação ao subgrupo que desenvolvemos acima. Pretendíamos calcular quantos movimentos eram necessários para resolver um estado contido no subgrupo utilizando apenas movimentos do subgrupo, ou seja, sem sair do subgrupo em questão.

Para isso, rodamos o seguinte código:

```
distance = {}
distance[0] = [cube()]

i = 0

all_cube_states = [cube()]

while distance[i] != []:
    cube_list = []
    for state in distance[i]:
        state_copy = state.copy()
        state_copy.make_move('f', 2)
        if not state_copy.is_in(all_cube_states):
            cube_list.append(state_copy)
            all_cube_states.append(state_copy)

        state_copy = state.copy()
        state_copy.make_move('r', 2)
        if not state_copy.is_in(all_cube_states):
            cube_list.append(state_copy)
            all_cube_states.append(state_copy)

        for j in range(3):
            state_copy = state.copy()
            state_copy.make_move('u', j + 1)
            if not state_copy.is_in(all_cube_states):
                cube_list.append(state_copy)
                all_cube_states.append(state_copy)

    distance[i + 1] = cube_list
    i += 1
```

A ideia do código é de criar um dicionário ("distance") cuja chave seja o número de movimentos necessários para os cenários, e o valor seja a lista de todos os cubos que precisam daquele número de movimentos para serem resolvidos. Assim, a cada iteração pegamos todos os cenários da iteração anterior ("distance[i]") e aplicamos a cada um deles todos os movimentos do subgrupo, gerando novos estados. Cada um desses novos estados ("state_copy") é verificado para conferir se já existe, caso não, adicionamos ele a lista de casos dessa nova iteração ("cube_list") e seguimos o processo.

Como resultado desse código, obtivemos a informação de que existem 32 cenários dentro do subgrupo que necessitam de 13 movimentos, entre os permitidos, para serem resolvidos. Esse resultado foi negativamente surpreendente, pois esperávamos que fossem necessários menos movimentos no pior dos casos. De qualquer forma, resolvemos seguir o estudo do caso.

A próxima etapa consiste em descobrir quantos movimentos são necessários para sair de uma classe lateral e chegar no subgrupo. Para isso, criamos uma versão fake do cubo, em que todas as peças são a mesma (peça 1) e a única informação que varia é a rotação dela.

Dessa forma, conseguimos focar nas rotações das peças, que é o que realmente determina a classe lateral de determinado estado. Com isso, rodamos um código, semelhante ao anterior, que verifica a quantidade de passos necessária para atingir o subgrupo:

```
distance = {}
only_orientation = [[1,0],[1,0],[1,0],[1,0],[1,0],[1,0],[1,0]]
distance[0] = [cube(only_orientation)]

i = 0

all_cube_states = [cube(only_orientation)]

while distance[i] != []:
    cube_list = []
    for state in tqdm(distance[i]):
        for j in range(3):
            state_copy = state.copy()
            state_copy.make_move('f', j + 1)
            if not state_copy.is_in(all_cube_states):
                cube_list.append(state_copy)
                all_cube_states.append(state_copy)

        for j in range(3):
            state_copy = state.copy()
            state_copy.make_move('r', j + 1)
            if not state_copy.is_in(all_cube_states):
                cube_list.append(state_copy)
                all_cube_states.append(state_copy)

        for j in range(3):
            state_copy = state.copy()
            state_copy.make_move('u', j + 1)
            if not state_copy.is_in(all_cube_states):
                cube_list.append(state_copy)
                all_cube_states.append(state_copy)

    distance[i + 1] = cube_list
    i += 1
```

Note que, a única diferença desse caso é de que utilizamos o cubo fake que só possui peças 1 e permitimos utilizar qualquer movimento possível ao cubo, sem a necessidade de ser um movimento do subgrupo.

Com isso, obtivemos o resultado de que o pior caso necessita de 6 movimentos para chegar ao subgrupo, sendo 40 casos como esse. A distância mais comum é de 5 movimentos, sendo 336 casos desse tipo. Vale lembrar que existem 729 classe laterais no total.

Com esses resultados, temos que no pior dos casos, precisamos de 6 movimentos para chegar ao subgrupo e outros 13 movimentos para resolver dentro do subgrupo, totalizando 19 movimentos. Esse resultado reduz nossa cota superior de 20 para 19, no entanto não é animador, visto que esperávamos um número bem menor. Portanto, seguiremos nosso estudo.

4.5.2 Algoritmo Força Bruta

Após realizar o estudo anterior, percebemos que os resultados não foram muito motivadores, porém os tempos de execução dos códigos foram. Ambos os códigos apresentados rodaram sem apresentar dificuldades, o primeiro levando menos de 10 segundos e o segundo em menos de 1 segundo já apresentou seu resultado.

Com isso em mente, resolvemos testar a força bruta e tentar encontrar as soluções de todos cenários possíveis, sem filtro de subgrupo nem filtro de movimentos. Para isso, utilizamos a mesma ideia inicial de resolução dentro do subgrupo, a única alteração foi a de incluir os demais movimentos presentes no grupo completo.

Ao rodar o código na mesma estrutura anterior mas sem limitar os movimentos tivemos problema de tempo de execução, contrariando nossas espectativas. Repensando a quantidade de casos, realmente existe uma grande diferença entre analisar 5.040 casos e analisar 3.674.160 casos.

Analisando o código, conseguimos fazer algumas alterações para otimizar o processo mantendo a mesma ideia. Primeiramente, utilizamos a biblioteca "numpy" pois permitem trabalhar com itens sequenciais de forma mais efetiva. Além disso, alteramos o momento de verificação de estado já visto equivalente, de modo que essa comparação passou a ser feita ao final de cada etapa por meio da retirada de registros repetidos. Com as alterações indicadas fomos capaz de rodar o código em pouco mais de 2 horas. Apesar de ser um tempo elevado, o código nos forneceu boas informações.

Como o código avalia a quantidade de movimentos necessárias para chegar em cada estado do cubo, com seus resultados podemos saber a exata quantidade de estados resolvíveis para cada quantidade de movimentos. Além disso, conseguimos descobrir o nosso tão sonhado número de Deus, que é 11. Os resultados geraram a seguinte tabela 1, indicando o número de movimentos necessários e a quantidade de embaralhamentos.

Número de movimentos	Quantidade de estados
0	1
1	9
2	54
3	321
4	1.847
5	9.992
6	50.136
7	227.536
8	870.072
9	1.887.748
10	623.800
11	2.644

Tabela 1 – Quantidade de estados para cada número de movimentos

Podemos verificar que a soma dos valores da segunda coluna realmente resulta em

3.674.160, indicando que mapeamos todos os casos do cubo.

Como mencionado, esse resultado, além de nos informar que o número de Deus para o 2x2x2 é 11, também nos fornece dados sobre a quantidade de movimentos necessária para resolver cada estado do cubo. Um ponto muito interessante é de que mais da metade dos possíveis embaralhamentos necessitam de 9 movimentos para serem resolvidos. Além disso, é curioso que o resultado obtido anteriormente, de que necessitamos de 13 movimentos para resolver estados dentro do subgrupo, indica que para alguns estados é mais efetivo sair do subgrupo e voltar posteriormente.

Com esse resultado, concluímos nossa busca ao número de Deus para essa versão simplificada do brinquedo.

5 Conclusão

Parte final do trabalho, apresenta as conclusões correspondentes aos objetivos ou hipóteses.

Bibliografia

[1] Tomas Rokicki et al. “The Diameter of the Rubik’s Cube Group Is Twenty”. Em: *SIAM Review* 56.4 (2014), pp. 645–670. DOI: 10.1137/140973499. URL: https://doi.org/10.1137/140973499.

"