



The Group Isomorphism Problem

and some other stuff...

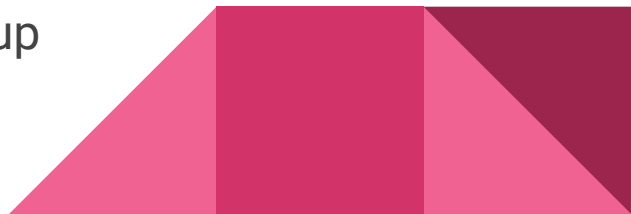
by Daniel Hathcock

What is a Group?

Definition 2.1.1. Group - Let G be a *nonempty* set with a binary, well-defined operation $*$. G is a group if (and only if) G has

1. *Closure* - $\forall a, b \in G, a * b \in G$
2. *Associativity* - $\forall a, b, c \in G, (a * b) * c = a * (b * c)$
3. *Existence of Identity* - $\exists e \in G$ such that $\forall a \in G, e * a = a * e = a$
4. *Existence of Inverse* - $\forall a \in G, \exists a^{-1} \in G$ such that $a * a^{-1} = e = a^{-1} * a$

- If the operation is also commutative, we call the group “Abelian”.



Examples of Groups

- Integers under addition? Yes!
- Integers under multiplication? No! Missing inverses.
- Natural numbers under addition $\{0, 1, 2, \dots\}$? No! Missing inverses.
- Examples of finite groups?...
- Set $\{0, 1, 2, \dots, n - 1\}$ under addition modulo n ?
- Set $\{1, 2, \dots, n - 1\}$ under multiplication modulo n ? Sometimes!



More Interesting Groups

- All of the previous examples have been abelian, since integer addition and multiplication commutes.
- What about the set of permutations (bijections) on a set of size 3?
 - $R = \{1, 2, 3\}$
 - $$S_3 = \left\{ \begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \end{pmatrix} \right\}$$
 - Call this $\{e, a, b, c, d, f\}$ (in the same order) for simplicity.
 - What operation might we use to make this a group?



S_3 (The Symmetric Group of Degree 3).

- Define the operation as composition of permutations (i.e. do one permutation, then do the other).
- This forms a nonabelian group!
 - e (the trivial permutation) is the identity element.
 - We could test to see that $a * b = d$, $b * a = f$, and $c * c = e$.
- Let's make a multiplication table!
- We can do this for any set (say of size n). This creates the symmetric group of degree n .
 - We know from permutations that the number of elements in this group is $n!$ (factorial).

When are Two Groups the Same?

- $\{0, 1\}$ under addition modulo 2 vs. $\{-1, 1\}$ under multiplication?
- $\{0, 1, 2\}$ under addition modulo 3 vs. Rotations of a triangle?
- S_3 vs. $\{0, 1, 2, 3, 4, 5\}$ under addition modulo 6?
- How many groups are there of order (size) 6?? How about of order 2?
- Are they the same when they have the same multiplication table?...



Isomorphisms

- Two groups are “the same” (isomorphic) when there is an isomorphism between them.

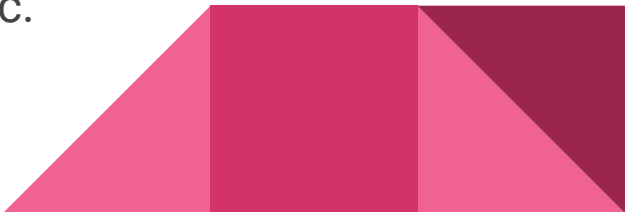
Definition 2.1.2. Isomorphism - Let G and G' be groups, and suppose $\varphi : G \rightarrow G'$ is a function from G to G' . Then φ is an isomorphism if it satisfies both

1. $\varphi(a * b) = \varphi(a) \cdot \varphi(b)$ for any $a, b \in G$. We say φ preserves the group operation.
2. φ is a bijective mapping. i.e. it is both injective (1-1) and surjective (onto). For finite groups, this means that G and G' have the same size, and every element of G maps to a unique element of G' .



Group Isomorphism Problem(s)

There are two related problems:

1. Let G and G' be groups, given by their (finite) group presentations. We want to determine if these two groups are isomorphic.
 - a. This problem is undecidable! There is no algorithm which can solve every instance of the problem.
 - b. Group presentations are complicated. A finite presentation can define an infinite group.
 2. Let G and G' be groups, given by their Cayley table (multiplication table). We want to determine if these two groups are isomorphic.
 - a. Decidable but difficult. I will talk about this problem.
 - b. Reduces to the **Graph Isomorphism Problem**!
- 

The Naive Approach

- Can you think of a simple (inefficient) brute force algorithm to solve the problem?
- Try everything!



Improvements?

- Subgroups
- Lagrange's Theorem
- Generating set of a group



Better Algorithms?

- Can we use some of the basic ideas in group theory to improve our algorithm?
- What if we are given a generating set A for group G with $|A| < |G|$?
- How big will A be? Can we compute some sufficiently small A quickly?



A Quasipolynomial Time Algorithm (Tarjan, 70s)

1. Compute a generating set A for G of size at most $\log_2(n)$ where $n = |G|$ (polynomial time)
2. Check all bijections between A and a subset of G' of size $|A|$.
 - a. For each bijection, “expand” to test whether the bijection is also an isomorphism (polynomial time)
 - b. Total number of bijections to check is $nC|A| * |A|! = n! / (n - |A|)! = O(n^{\log(n)})$
3. So, total running time is $O(n^{\log(n) + O(1)})$



Recent Improvements

- Lipton gave a slightly stronger result: Group Iso in $O(\log^2(n))$ space (1976).
- Few improvements between 70s and 2010s.
- The classification of finite simple groups (2004) provides a polynomial time algorithm for groups which are known to be simple
 - They have a generator set with only 2 generators.
- In 2012, Babai gave a polynomial time algorithm for groups with no abelian normal subgroups
- In 2013 Rosenbaum gave a slightly improved general algorithm with running time $O(n^{0.5\log(n) + o(\log(n))})$



A Related Problem: Graph Isomorphism

- **Graph:** a set of vertices V , and edges between them E .
 - Directed graph - edges have an order. i.e. edge (x, y) is drawn as $x \rightarrow y$.
 - Undirected graph - edges have no order. i.e. (x, y) is drawn as $x - y$.
- Two graphs (V, E) and (V', E') are isomorphic if there is a bijection (permutation) s from V to V' such that $(x, y) \in E$ if and only if $(s(x), s(y)) \in E'$
 - One graph's vertices is just a relabeling of the other's vertices.
- Groups have much more structure than graphs, so we might conjecture that the group isomorphism problem is “easier” than graph Isomorphism. How can we show this?



Reductions

- If problem A is “harder” than problem B, we might hope that an algorithm which efficiently solves A can be used as a subroutine to solve B with the same efficiency.
- In complexity theory, this is called a “reduction”. Intuitively, it is an algorithm that converts an easy problem to a hard problem efficiently (constant or polynomial time).



Groups and Graphs

- Can we come up with a way to reduce the group isomorphism problem to the graph isomorphism problem? (Hint: Yes we can...)
- It isn't immediately obvious how though.
- Instead we will first look at how to reduce directed graph isomorphism to graph isomorphism.
 - This might give us an idea...



Isomorphism Problem Difficulty

- The reduction is interesting...but not very useful.
- The best currently accepted algorithm for graph iso has time complexity $2^{O(\sqrt{n} \log(n))}$
 - Babai's $2^{O(\log(n)^3)}$ has not yet been peer reviewed, but is still worse than current group iso state of the art algorithms.
- Babai thinks that the group isomorphism problem is the only isomorphism problem of many which is expected to be solvable in polynomial time.



Sources

<https://www.cs.cmu.edu/~glmiller/Publications/Papers/Mi79.pdf>

<https://www.cs.cmu.edu/~glmiller/Publications/Papers/Mi78.pdf>

<http://people.cs.uchicago.edu/~joshuag/grochow,qiao-gpiso.pdf>

Wikipedia :)

