Computational Group Theory

Ethan Padden, Jack Lynch, Anthony Horgan

Introduction

Computational Group Theory (CGT) involves *“the design, implementation, and analysis of algorithms for groups and related objects”*.
Software such as GAP and MAGMA have been developed to assist research in algebra. When using systems such as these, the groups are usually represented as matrices, because large and even infinite groups can be represented in such a way, using a small input.

History

Almost immediately after the birth of group theory, the first algorithmic questions about groups appeared.
The origin of computational group theory can be traced back to 1911 when Max Dehn, an American mathematician, posed three questions concerning groups defined by finite presentations: the word problem, conjugacy problem and Isomorphism problem. These questions prompted the idea of using algorithms to solve problems related to group theory.
Alan Turing in 1945 first proposed the use of computers to answer group theory questions. Up to this point computers were primarily used for numerical computation. In the 1950’s attempts to implement coset enumeration began. This made coset enumeration one of the first non-numerical procedures to be programmed.
In 1981 the classification of finite simple groups was completed. This had an important impact on computational group theory as mathematicians found it difficult to show the existence of these large groups. This prompted an effort to provide computer proofs which in turn provided motivation for improving algorithmic techniques.
Up until the 1970’s the contributors to computational group theory were mainly group theorists. This changed in the 1980’s with the discovery that permutation group algorithms and the isomorphism problem were closely connected. With the graph isomorphism problem being one of the most famous problems in computer science, it drew computer scientist into the field. These computer scientists created new data structures and algorithms with the goal of improving algorithm efficiency and complexity. This collaboration between group theorists and computer scientists proved important as they used their distinct skills to further the field of computational group theory. Since the late 1970’s the field of computational group theory has been growing rapidly.

Algorithms

A large number of algorithms in CGT involve choosing random elements from groups, and for the level of generality required, algorithms designed to work in black box groups are the most suitable. A polynomial time algorithm for doing this was described by Babai in 1991, but it is too slow for practical purposes. The Product Replacement Algorithm of Charles Leedham Green provides a satisfactory compromise between true randomness and speed of execution.

Black Box Group

A group G whose elements are encoded by bit strings of length N, and group operations are performed by an oracle (the "black box"). These operations include:

- taking a product g·h of elements g and h
- taking an inverse g1 of element g
- deciding whether g = 1

The Product Replacement Algorithm

Input: A black box group $G = \langle X \rangle$ and parameters n=10 and c=50.

1. Define list Y: Let Y be an ordered list with $|Y| = max(|X|, n)$ containing the elements of X, with elements repeated if $|X| \leq n$.
   Base Move: Choose random distinct i,j with $1 \leq i, j \leq n$ and replace $Y[i]$ by one of the four elements
   $$Y[j]Y[i], \quad Y[j]^{-1}Y[i], \quad Y[i]Y[j], \quad Y[i]Y[j]^{-1}$$
   chosen at random
2. Initialization: Perform the basic move c times.
3. Generate a random element: Perform the basic move and then return $Y[i]$.

The Schreier–Sims algorithm

Many early algorithms in CGT, such as the Schreier–Sims algorithm, require a permutation representation of a group and thus are not black box. Named after mathematicians Otto Schreier and Charles Sims. It allowed a O(n) time of the order of a finite permutation group. Timing was subsequently improved by Donald Knuth. Later, an even faster randomized version of the algorithm was developed.

Sub-areas

There are 4 major subjects that CGT is focusing on:

Permutation Groups

Let G be a group of permutations of a finite set. Research in CGT has led to algorithms for answering questions such as:

What is the order of G?

Is G solvable?

"Let is denote G by H_0. We say that G is a solvable group if there exists a sequence of subgroups $H_1, \ldots, H_k$ of G such that, for each j, $1 \leq j \leq k$, H_j is a normal subgroup of $\frac{H_{j-1}}{H_j}$ is abelian, and $H_k = id_G$"

Is G nilpotent?

“A group G Is nilpotent if $Z_i(G) = G$ for some i.”

What are the Sylow subgroups of G?

“Let G be a group, and let p be a prime number. A group of order p^k for some $k \geq 1$ is called a p-group. A subgroup of order p^k for some $k \geq 1$ is called a p-subgroup. If $|G| = p^\alpha m$ where p does not divide m, then a subgroup of order $p^\alpha m$ is called a Sylow p-subgroup of G.”

Finitely-Presented (FP) Groups

Revisiting word problems from before, there is a theorem that states that, in general, the word problem for fp-group is undecidable: *“A problem is said to be* ***Decidable*** *if we can always construct a corresponding* ***algorithm*** *that can answer the problem correctly”* This means that computing with FP-groups is quite different with groups in a more concrete form. Questions that CGT has helped answering are:

- Is G the trivial group?
- Is G finite?
- If G is infinite, is it free?
- What are the abelian/nilpotent quotients of G?
- Is G abelian/nilpotent?
- Can we construct a permutation representation for G?
- Can we construct a matrix representation for G?

Polycyclic Groups

Let G be a group. *"G is polycyclic if it has a descending chain of subgroups $G = G_1 \geq G_2 \geq \ldots \geq G_{n+1} = 1$ in which $G_{i+1} \triangleleft G_i$ and $\frac{G_i}{G_{i+1}}$ is cyclic."*

Represetation Theory

Representation theory is a field of study that enables us to represent groups as a group of linear maps on a vector space: *A representation of G is a group homormophism: $\rho : G \longrightarrow GL(V)$*
Computer algebra systems are powerful tools when it comes to studying polycyclic groups and representations and throughout the years have been integral to this research.

The Matrix Group Recognition Project

“The aim is to produce efficient algorithms for solving problems with matrix groups over finite fields, as well as making efficient implementations of these algorithms.”
Often, one needs to gather structural information about a given group (and other tools for computation). The project’s main focus is to create an algorithm to provide this information by computing a composition tree of a matrix group using a set of generators in order to establish the composition series of the group.

Composition Series

"Let G be a group. A Composition Series for G is a (finite) chain of successive subgroups of G, denoted by $e = G_0 \leq G_1 \leq \ldots \leq G_n = G$ with the following properties:

1. *G_i is a normal subgroup of $G_{i+1} forall 0 \leq i \leq n-1$.*
2. *G_{i+1}/G_i is a simple group for all $0 \leq i \leq n-1$.*

The Length of the composition series is the number n, and the Composition Factors of the composition series are the quotient groups $\frac{G_{i_1}}{G_i}$."

Achievements

- A listing of all finite groups of order less than 2000.
- Computation of representations for all the sporadic groups.
- Enumeration of regular maps up to genus 100.
- Computing cohomology of some finite groups.

National University of Ireland, Galway
College of Engineering and Informatics
SCHOOL OF MATHEMATICS

NUI Galway
OÉ Gaillimh