Computational Group Theory

The study of algorithms for groups,

Aim: answer questions for concrete groups

What kind of questions?

Given G, determine its size,

find its subgroup, Sylow subgroups, quotients

inspect orders of elements

find normal forms of words,

small generating sets

construct composition series

decide if group is simple / nilpotent, solvable.

→ maybe even recognize G as a "std group".

construct groups w/ particular properties

For most of these questions, there are NO algorithm that answer these. (Halting problem)

for particular groups, there are procedures that "more or less work"

↙ we will look into what that means.

What kind of groups?
(he means these are well-implemented)

There are 4 most common types:

1. Permutations on a finite set S_n.
2. Matrices
3. Finitely presented groups
4. Polycyclic group — groups w/ cyclic decomposition

THIS IS WHAT PYLAH did! Simpler=better

see Urban's lecture

see Primer's lectures

⇒ Given a group, good to choose between these 4 types & implement algo appropriate to that type

↑ what we mean by concrete group

In this course, we will look at how these are implemented in GAP

groups algorithms programming

I gotta look at reference manual & tutorial.

Packages can download.

These are peer-reviewed!

Should submit yours!!!

It's not clear how GAP does things internally

↖ look at documentation.

Demo

- Asked questions about group
  derived subgroup because alternating

=> GAP saves properties

Known Properties of Object (),
Know Attributes Of
and changes which algorithm it uses depending on the answer!

→ there no need to ask the same Q twice, always saved!

- pc group = polycyclic group.
- map = "" List (l, t → IsAbelian(t)) ""
- ^G : means all the G orbit.
- Id Group (g) : Gap might know precisely what the group is.
  +
  StructureDescription(g)

In this course:

- Will look at what GAP actually does when we ask it questions (ex: size).
- "Handbook of Computational Group Theory"