G - large (but finite) group

$G = \langle S \rangle$ elements of S - permutations
(of large degree).

Aims: • Compute the order of G.

• find out if given permutation σ actually belongs to G (membership test).

usually hard when G is given abstractly.

sometimes easy when G is given by property

Anti-aims:

• enumerating / storing all of elements of G.

In general we may want to store $O(|S|)$ additional elements to speed up the computations

(Note: usually $|G| \sim O(2^{|S|})$).

Basis and stabilizer chains.

let (G, S) be given as previously.

We want to find a sequence/vector/tuple/list of points $(\beta_1, \dots, \beta_m) \in \mathbb{N}^m$ s.t. every $\sigma \in G$ can be uniquely determined by $(\sigma(\beta_1), \dots, \sigma(\beta_m))$.

Ex:

$\sigma = (1,2)(3,4) \dots (999, 1000)$

$\tau = (1,2)(3,4), \dots, (999, 1000, 1001)$

$G = \langle \sigma, \tau \rangle \subset Sym(1001)$ but it's enough to observe the action of $g \in G$ on $(\beta_1, \dots, \beta_3) = (999, 1000, 1001)$.

Suppose that such $(\beta_1, \dots, \beta_m)$ is given and $(\alpha_1, \dots, \alpha_m)$ is supplied.

Can we determine the permutation $\sigma \in G$ that takes $(\beta_1, \dots, \beta_m) \longrightarrow (\alpha_1, \dots, \alpha_m)$?

Consider

$$G = G^{(0)} > G^{(1)} > \dots > G^{(m)} = \{id\}$$

where $G^{(i)} = Stab_{G^{(i-1)}}(\beta_i)$.

$(\beta_1, \dots, \beta_m)$

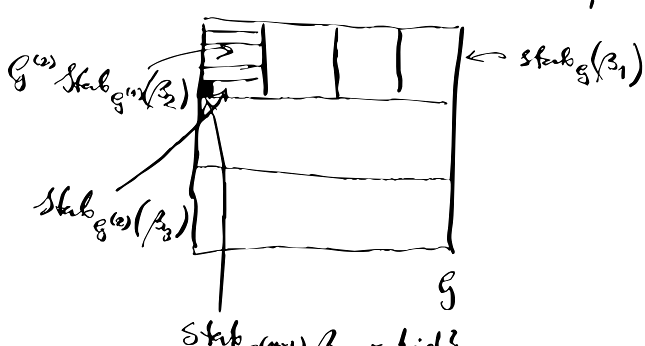
- only id stabilizes all of them.
- pick β_1 and let $G^{(1)} < G$ be its stabilizer

By Orbit - Stabilizer we can divide G into $\mathrm{Stab}_G(\beta_1)$ - cosets - given by the orbit β_1^G

$$\beta_1 \to \beta_1^{g_1} \to \dots \to \beta_1^{g_k}$$

$G^{(1)} = \mathrm{Stab}_G(\beta_1)$ $\quad \mathrm{Stab}_G(\beta_1)g_1$

Inside $\mathrm{Stab}_G(\beta_1)$ find the stabilizer of β_2



- every element of $\mathrm{Stab}_{G^{(1)}}(\beta_2)$ fixes β_1 <u>and</u> β_2
- all elements that fix β_1 can be divided into subsets based on where do these send β_2.

Thus given $\sigma \in G$ we can find $\sigma(\beta_1)$ and observe that $(\sigma \cdot r_1^{-1})(\beta_1) = \beta_1$ where r_1 is a representative of the coset corresponding to $\sigma(\beta_1)$.

Let $\sigma_1 = \sigma \cdot r_1^{-1} \in Stab_G(\beta_1) = G^{(1)}$ and find $\sigma_1(\beta_2)$, identify the corresponding coset of $Stab_{G^{(1)}}(\beta_2)$ in $G^{(1)}$.

We play the same game and see that

$$\sigma_2 = \sigma_1 \cdot r_2^{-1} = \sigma \cdot r_1^{-1} \cdot r_2^{-1}$$

stabilizes both β_1 and β_2.

By following this procedure we find out that $\sigma_m = \sigma \cdot r_1^{-1} \cdot r_2^{-1} \cdots r_m^{-1}$ stabilizes all β_i, hence is the identity. From this we recover

$$\sigma = r_m \cdot \ldots \cdot r_1 .$$

ALGORITHM: Sift / membership test

INPUT:
- $(\beta_1, \dots, \beta_m)$ — basis for $\mathcal{G} \subset Sym(d)$
- $g \in Sym(d)$

OUTPUT:
- $L = [b_1, \dots, b_m]$ of coset representatives for $\mathcal{G} = \mathcal{G}^{(0)} > \mathcal{G}^{(1)} > \dots > \mathcal{G}^{(m)} = \{1\}$
- $r \in \{Sym(d) \setminus \mathcal{G}\} \cup \{e\}$ s.t. $g = r \cdot b_m \cdot \dots \cdot b_1$

```
begin
  L = [ ]
  G^0 = G
  r = g
  for i in 1:m
      T = transversal (β_i, G^{i-1})
      δ = β_i^r
      if δ ∉ T
          return L, r          // r ≠ e ; length(L) = i-1
      end
      push b_i to L
      r = r · b_i^{-1}
      if r = e
          return L, r          // length(L) = i
      else
          G^i = Stab_{G^{i-1}}(β_i)
  end
  return L, r          // happens only when g ∉ G
end                    //       and then r ≠ e
                       //  note: length(L) = m here.
```

Notes:

- basis, transversals and stabilizers are interconnected, so we will be building them together at the same time as a Stabilizer Chain structure.
- We shouldn't use Schreier generators though: by the time we finish we'll end up with $\mathcal{O}(2^{|S|})$ of them!
- we will usually take $\beta_i = \text{first}(T_i)$ (the first element on the orbit)

A partial stabilizer chain is a sequence

$$\mathcal{C} = \{ G^{(0)} > G^{(1)} > \dots > G^{(n)} = \{id\} \}$$

such that $\text{Stab}_{G^{(i-1)}}(\beta_i) \geq G^{(i)}$.

A stabilizer chain (proper, complete) is a similar sequence where $\text{Stab}_{G^{(i-1)}}(\beta_i) = G^{(i)}$.

Note: partial stabilizer chain is proper iff $|G| = \prod_i \left(\prod_{1 \le j \le i} |\Delta_j| \right) \cdot |G^{(i)}|$ for every i.

$$|G| = |\Delta_1| \cdot |G^{(1)}| = |\Delta_1| \cdot |\Delta_2| \cdot |G^{(2)}| =$$
$$= \prod_i |\Delta_i|$$

How to complete a partial stabiliser chain?

Gien a new generator g (Schreier generator)

we sift it through the chain:

ALGORITHM: extend(C, g)

INPUT: • C – a (partial) stabiliser chain for G

• g – an element in G

OUTPUT: • C – containing the g (possible without modifications).

```
begin
    L, r = sift(C, g)
    if r ≠ id    \\ g is not in the group <C>
        d = length(L)    ← the depth where this was recognized
        push!(C, r, d)
    end
end
```

ALGORITHM: push! ($\mathcal{C}$, g, d)

INPUT:
- $\mathcal{C}$ – (partial) stabilizer chain
- g – permutation
- d – depth (a non-negative integer)

OUTPUT:
- $\mathcal{C}$ – (partial) stabilizer chain with g added

```
begin
    assert β_i^g = β_i  for all  i < d
    if d = length(basis(C))  // add new layer to C
        β = first_moved(g) ; S = [g]
        push  (β, S, transversal(β,S)) to C.

    else
        push!(S_d, g)
        // since we extended the generator set at
        // level d we need to
        // 1) update the transversal
        T_d = transversal(β_d, S_d)
        //  sift any new schreier generator
        //    that arises from g down the chain
        for s in schreier_generators(T_d, S_d)
            push!(C, s, d+1)
        end
    end
    return C
end
```

Defn: A strong generating set (sgs) for G is a set S such that $G = \langle S \rangle$ and

$$G^{(i)} = \langle S \cap G^{(i)} \rangle.$$

If $\mathcal{C}$ is a completed stabilizer chain for G, then $S = \bigcup_{i=1}^{d} S_i$ is a sgs.

In the other direction: Given a sgs (and the corresponding basis) we can rebuild the stabilizer chain by simply computing the transversals.

Performance notes:

- There is no need to compute all Schreier generators when recomputing the transversal happens.
- Unfortunate choice for generators may lead to very long T_i's on each level.

Ex: $G = \langle a = (1, \dots, 100), \overset{b=}{(1,2)} \rangle$

$\beta_1 = 1$, representative $= a^i$, $-50 < i < 50$.

better generating set:

$$\langle \qquad \rangle$$

How?

- Expensive operations:
  - permutation multiplication:

    every $a \cdot b$ allocates!

    → store the products as words in generators.

→ If $H < G$ and basis β for G is known we can always store β^g instead of g!

then $g \cdot h$ is $(\beta^g)^h$.

the cost of multiplication:

$$O(\mathrm{degree}(G)) \to O(\mathrm{length}(\mathrm{basis})).$$

If $|G|$ is known beforehand (eg. we're recomputing the chain) then we could quickly terminate as soon as $\prod_{i=1}^{d} |T_i| = |G|$.
this usually avoids sifting of most of the generators.

Lemma: If $\mathcal{C}$ is a partial stabilizer chain for G then chosen uniformly at random $g \in G$ fails the membership test with $\mathcal{C}$ with probability at least $\frac{1}{2}$.

Corollary:
If elements g are chosen uniformly at random from G, then the probability of n of them passing the membership test with an incomplete chain is at most $(1-\frac{1}{2})^n$.

Summary:
Consequences of Schreier-Sims / base & stabilizer chain:

- membership test for G
- compute $|G|$ as $\prod_{i=1}^{d} |T_i|$
- Given $\gamma = (\gamma_1, \dots, \gamma_d)$ fing $g \in G$ s.t. $\beta^g = \gamma$.
- Normal closure as the stabilizer of H under the action $(g, H) \mapsto g^{-1} H g$.
- derived series: $D_0 = G$; $D_i = D_{i-1}'$ ← the commutator subgroup
- lower central series: $L_0 = G$; $L_i = [G, L_{i-1}]$
- test whether two elements are in the same coset of a subgroup

- Determine the permutation action on the cosets of a subgroup

- Determine point-wise stabilizer of a set
- enumerate G
- Obtain random elements from G with guaranteed uniform distribution.

Factorisation into generators.

$$g = r_1 \cdot \ldots \cdot r_k = \underbrace{s_{11} s_{12} \cdots s_{1m_1}}_{r_1} \cdot \underbrace{s_{21} \ldots s_{2m_2}}_{r_2} \cdot \ldots \cdot \underbrace{s_{k1} \cdots s_{km}}_{r_k}$$

this is usually very far from minimal.

solution: minimize r_i by e.g. flattening the schreier trees.
(but this still will not give you minimality).

Homomorphisms:

If we know (sgs, basis) for $G = \langle S_g \rangle$
have a homomorphism $\varphi: G \to H$
we can quickly evaluate it by
- starting with $\{(s, \varphi(s))\}_{s \in S_g} \subset G \times H$
- doing the computation of sgs in G and mirroring the group operations on H
- If $g \in G$, $g = r_1 \cdot \ldots \cdot r_k \Rightarrow$ the computations gives us $\varphi(r_1), \ldots, \varphi(r_k)$

If H is a permutation group then

$G \times H$ is also $\Rightarrow$

$$G \times H \overset{i}{\hookrightarrow} \mathrm{Sym}(\mathrm{degree}(G) + \mathrm{degree}(H))$$

$$\underbrace{i(\Omega_G)}_{\deg(G)} \quad \underbrace{i(\Omega_H)}_{\deg(H)}$$

then $\underbrace{i(\langle\langle (s, \varphi(s)) \rangle\rangle)}_{J}$ acts on $i(\Omega_G) \cup i(\Omega_H)$

$\ker \varphi \cong$ pointwise stabilizer of $i(\Omega_H)$.

<u>Backtrack:</u>

An algorithm to traverse the tree formed by a stabilizer Chain.

<u>Aims:</u> find all/one elements satisfying certain property.

<u>Ex:</u>
- Centralizer and Normalizer in permutation groups.
- Conjugating element
- Set stabilizer
- Graph isomorphism