Backtrack:

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

Aims: find all/one elements satisfying certain property.

Ex: • Centralizer and Normalizer in permutation groups.

- Conjugating element
- Set stabilizer
- Graph isomorphism

The tree of a Stabilizer Chain C

- root – empty node
- first layer – the orbit of transversal(C, 1)
- children of a node at level/depth d –
  – the orbit of transversal (C, d+1) shifted by the element corresponding to the node.

ALGORITHM: Backtrack!

INPUT:
- L - an (empty) list
- $\mathcal{C}$ - stabilizer chain for $G = \langle S \rangle$
- $g = e$ - an element of G.
- $d = 1$ - depth

OUTPUT:
- L - a list of all elements of G.

```
begin
    T = transversal(C, d)
    for δ ∈ T
        if length(C) = d        // we're in a leaf node
            push g·T[δ] to L
        else
            Backtrack!(L, C, g·T[δ], d+1)
        end
    end
    return L
end
```

- We can add a predicate p and only push when $p(g \cdot T[\delta])$ is satisfied.
- <u>Problem:</u> this runs over all leafs when sometimes whole branches can be discarded by the predicate
- <u>Solution:</u> Add a <u>problem specific</u> for $\delta \in T$ and avoid descendig into the whole branch.

- General procedure:

Given a group $\mathcal{G}$ and P, a problem to solve

- find the optimal basis β for the problem
- use the existing SC to complete $SC(\beta)$ (e.g. knowing the order of $\mathcal{G}$ helps, there are algorithms for transforming one basis to another)
- use backtrack + check for P to prune the search tree.

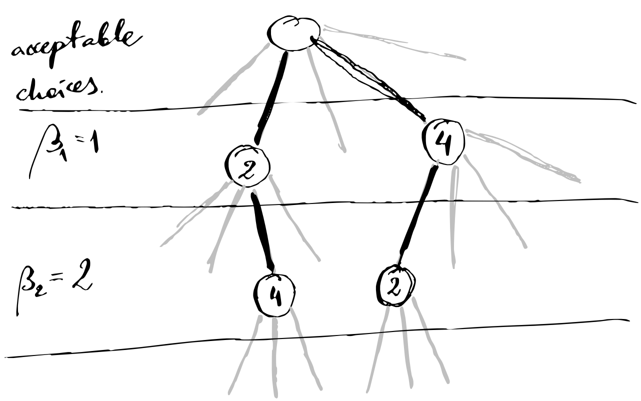
Ex, Searching in $Sym(5)$ for g such that

$$(1,2)(3,4,5)^g = (2,4)(1,5,3)$$

$\beta = (1, 2, \dots)$

We immediately know that $(1,2) \longrightarrow (2,4)$

$1 \longmapsto 2$
$1 \longmapsto 4$ are the only acceptable choices.



now

Ex 2:

$$\mathcal{G} = \langle \overset{a}{(1,3,5,7)(2,4,6,8)},\ \overset{b}{(1,3,8)(4,5,7)} \rangle$$

find $C_{\mathcal{G}}(x)$ where $x = (1,2,4)(5,6,8)$.

$$C_{\mathcal{G}}(x) = \{ g \in \mathcal{G} : g^{-1} x g = x \text{ i.e. } xg = gx \}$$

$|\mathcal{G}| = 24$. $\quad \beta = (\beta_1, \beta_2) = (1,2)$

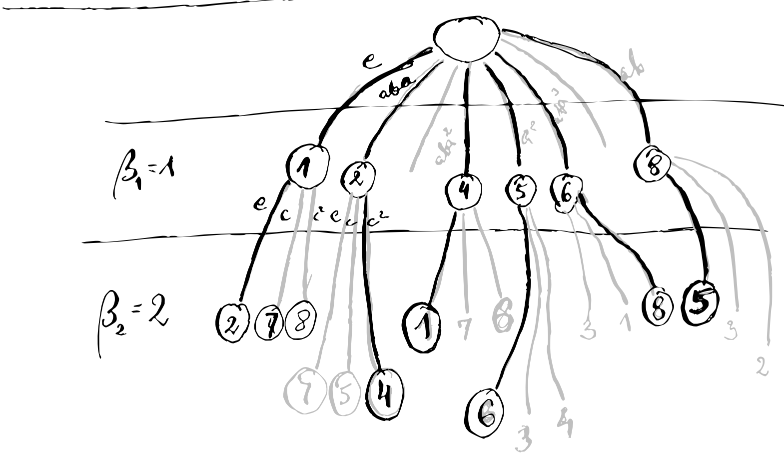
$$1^{\mathcal{G}} = \{1, 3, 5, 8, 7, 2, 4, 6\}$$

$$T_1 = \{e, a, a^2, ab, a^3, aba, aba^2, aba^3\}$$

$$\mathcal{G}^{(1)} = \mathrm{Stab}_{\mathcal{G}^{(0)}}(\beta_1) = \langle ab^{-1} = (1)(2,7,8)(3,4,6), \dots \rangle$$

$$2^{\mathcal{G}^{(1)}} = \{2, 7, 8\}$$

$$T_2 = \{e, c, c^2\}$$



$\begin{cases} 1 \to 1 \\ 2 \to 2 \end{cases}$ $\begin{cases} 1 \to 5 \\ 2 \to 6 \end{cases}$

$\begin{cases} 1 \to 2 \\ 2 \to 4 \end{cases}$ $\begin{cases} 1 \to 6 \\ 2 \to 8 \end{cases}$

$\begin{cases} 4 \to 4 \\ 2 \to 1 \end{cases}$ $\begin{cases} 1 \to 8 \\ 2 \to 5 \end{cases}$

Ex 3: Setwise Stabilizer.

$X \subset \Omega$

$(\beta_1, \dots, \beta_k)$ chosen from X

$\Rightarrow \quad \mathrm{Stab}_G(X) \geq G^{(k+1)}$.

Finish the basis and do the backtrack search for $\beta_i^g \in \{\beta_1, \dots, \beta_k\}$ for $i \leq k$.

Partition backtrack

→ nodes maintain information about cycle mappings.

Groups / elements defined by properties:

- Centralizer, Normalizer
- Set stabilizer
- given $a, b \in G$ are they conjugated? If so find $g \in G$ s.t. $g^{-1} a g = b$.

Note: $(a' g b')^{-1} a (a' g b') =$

$$= b'^{-1} g^{-1} a'^{-1} a a' g b' \underset{\substack{\uparrow \\ \text{if } a' \in C_G(a)}}{=} b'^{-1} g^{-1} a g b' \underset{\substack{\uparrow \\ \text{since } g^{-1} a g = b}}{=}$$

$$= b'^{-1} b b' \underset{\substack{\uparrow \\ \text{if } b' \in C_G(b)}}{=} b$$

Hence: any solution g comes with $C_G(a) \cdot g \cdot C_G(b)$ of other ones.

Finding a subgroup: P

- start with $K = \langle \emptyset \rangle$, whenever a new elt g satisfying the predicate is found update $K := \langle K, g \rangle$ ← build the stabilizer chain for K along!
- descend down the SC and find $P \cap G^{(i+1)}$ before considering elts from $G^{(i)}$

Lemma:

Suppose that $K = G^{(i)} \cap P$ and N is a node prescribing $(\beta_1, \dots, \beta_i)$.

If a leaf g, below N belongs to P, then we can update $K = \langle K, g \rangle$ and prune the whole branch below N from the search.

Proof: any element below N that is in P is also in Kg.

Corollary:

If $g \notin K$ and $g \notin P$ $\Rightarrow$ none of elements from KgK is in P.

Problem: How to test that $h \in KgK$ (for every g found as above)?

Solution: given h find a "canonical representative" of KhK and compare it to the known representatives.

this is hard.