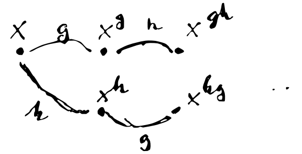
Transversals:

we know that $x^G \leftrightarrow \mathrm{Stab}_G(x) \backslash G$.

can we not only find the orbit of x, but also representatives for cosets?



$[x,\ x^g,\ x^h,\ x^{gh},\ x^{hg},\ \dots]$

$[1,\ g,\ h,\ gh,\ hg,\ \dots]$

Defn:

<u>Transversal</u> is the list of coset representatives associated to orbit x^G.

Notation:

- We will blend the notions of an orbit and a Transversal.
- If T is a transversal for x^G, then $T[y] = g$ s.t. $x^g = y$.

ALGORITHM: ORBIT-TRANSVERSAL

<u>INPUT:</u>
- S - set of generators of $\mathcal{G}$
- x - point to act on

<u>OUTPUT:</u>
- Δ - orbit $x^{\mathcal{G}}$
- T - associated transversal

```
begin
    Δ = [x]
    T = [e]   <- group identity
    for δ ∈ Δ
        for s ∈ S
            γ = δ^s
            if γ ∉ Δ
                push γ to Δ
                T[γ] = T[δ]·s   <- group element which sends x to δ.
            end
        end
    end
    return Δ, T
end
```

Problem with this algorithm: storage.

1 permutation of length 2^{10} is 8KiB

10^6 such permutations $\sim$ 10 GiB

1 perm of length 2^{20} is $1024 \cdot 8$ KiB $= 8$ MiB

512 of such is already 8 GiB

Solution: Only store the minimal information required to reconstruct elements from the transversal!

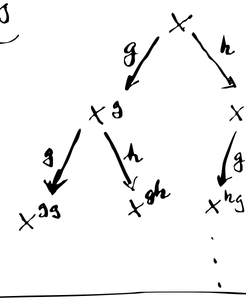
Definition:

Let $\Delta = x^G$ (as ordered set).

We say that $V = (v_1, \dots, v_k) \in (S \cup \{e\})^k$ Schreier vector for Δ iff

1) $k = |\Delta|$ $\qquad \Delta = [x, x^g, x^h, x^{gh}, \dots]$
2) $v_1 = e$ $\qquad V = [e, g, h, h, \dots]$
3) whenever $y \in \Delta$ and $\underbrace{v[y] = s}$ then $y^{s^{-1}}$ occurs in Δ before y.

the entry of v associated to y.



INPUT:
- S – a generating set for $G = \langle S \rangle$
- x – a point $\in \Omega$

OUTPUT:
- Δ – x^G, the orbit of x
- v – Schreier vector / tree

```
Δ = [x], v = [e]
for δ ∈ Δ
    for s ∈ S
        γ = δ^s
        if γ ∉ Δ
            push γ to Δ; push s to v
        end
    end
end
return Δ, v    end
```

ALGORITHM: RECONSTRUCT REPRESENTATIVE

INPUT: Δ – orbit of x

v – associated Schreier vector/tree

y – point in Δ

begin

  $\gamma = y$ // current point

  $r = e$ // coset representative

  while $\gamma \neq x$

    $s = v[\gamma]$

    $r = s \cdot r$ // since we're moving up the tree we multiply on the left

    $\gamma = \gamma^{s^{-1}}$

  end

  return r // r takes $x \curvearrowright y$

end

Generating set for the Stabilizer.

Let $G = \langle S \rangle = \langle s_1, \dots, s_k \rangle$ and let

$H < G$ be of finite index with

coset representatives $\{r_1, \dots, r_n\}$ (we assume that $r_1 = e$).

for $g \in G$ there exists a unique coset s.t

$g \in H \cdot r_j$. we will write $\overline{g}$ for r_j.

Let $U = \{ r_i s_j (\overline{r_i s_j})^{-1} \}_{\substack{i=1,\dots,n \\ j=1,\dots,k}}$

__Lemma:__ $H = \langle U \rangle$.

U is called the set of __Schreier generators__.

__Proof:__

1) $r_i s_j (\overline{r_i s_j})^{-1} \in H$ ✓

2) let $x \in H$, $x = g_1 \cdot \dots \cdot g_m$, $g_i \in S$.

$$x = g_1 \cdots g_m = \overbrace{1 \cdot g_1 \cdot (\overline{1 \cdot g_1})^{-1}}^{u_1 \in U} (\overline{1 \cdot g_1}) \cdot g_2 \cdots g_m$$

$$= u_1 \cdot \underbrace{\overline{1 \cdot g_1} \cdot g_2 \cdot (\overline{\overline{1 \cdot g_1} g_2})^{-1}}_{u_2 \in U} (\overline{\overline{1 \cdot g_1} g_2}) g_3 \cdots g_m$$

$$\vdots$$

$$= \dots = u_1 \cdot \dots \cdot u_{m-1} \underbrace{\overline{1 \cdot g_1 \cdot g_2 \cdot g_3 \cdot \dots \cdot g_{m-1}}}_{t} \cdot g_m =$$

Claim: $\overline{t \cdot g_m} = 1$

$x \in H$ & $u_i \in H \Rightarrow t \cdot g_m \in H \Rightarrow \overline{t \cdot g_m} = 1$

$$= u_1 \cdot \dots \cdot u_{m-1} \cdot t \cdot g_m \, \overline{t g_m}^{-1} \in \langle U \rangle.$$

Application:

$H = \mathrm{Stab}_{\mathcal{G}}(x) = \{ g \in \mathcal{G} : x^g = x \}$

$\Delta \leftarrow$ orbit of x $\longleftrightarrow$ cosets of $H \backslash \mathcal{G}$.

transversal

$\underbrace{T[x^g]}_{=g} \longleftrightarrow Hg$ ← coset

ALGORITHM: Orbit/stabilizer

<u>INPUT</u>: • S – set of generators for $\mathcal{G} = \langle S \rangle$

• x – a point in Ω

<u>OUTPUT</u>: • Δ – the orbit of x

• T – transversal for Δ (Schreier or not)

• U – set of Schreier generators

$|\Delta| \cdot |S| - |\Delta| \cdot |S| \cdot (|\Delta| - 1)$

```
begin
    Δ = [x]
    T = [e]
    U = [e]
    for δ ∈ Δ
        for s ∈ S
            γ = δ^s
            if γ ∉ Δ
                push γ to Δ  } we'll enter this |Δ|-1 times
                push s to T  }
            else
                push T[δ]·s·T[γ]^-1 to U
            end
        end
    end
    return Δ, T, U
end
```

$\underbrace{T[\gamma]^{-1}}_{(T[\delta'])^{-1}}$

$\overline{T[\delta] \cdot s}$

Performance: (problem)

- $\mathcal{U}$ will have $|\Delta|\cdot|S|-(|\Delta|-1) = |\Delta|(|S|-1)+1$ Schreier generators
  that's plenty, but sometimes all are needed
- remove duplicates and identity
- instead of collecting all those generators simply form a group $H = \langle \mathcal{U} \rangle$, and check if the new one already belongs to H.
  ↳ we need membership test.

Further applications:

- Normal closure of $H < \mathcal{G}$ i.e.

$$\langle\langle H \rangle\rangle_{\mathcal{G}} = \bigcap \{ N \triangleleft \mathcal{G} \; : \; H \leq N \}$$

Idea:
Start the orbit algorithm with $\Delta = \mathcal{U}$ ← generating set for H
under the action $g^h = h^{-1} g h$.

ALGORITHM: Normal closure:

INPUT : • S - generating set for $\mathcal{G}$
• $\mathcal{U}$ - generating set for H

OUTPUT : • N - generating set for $\langle\langle H \rangle\rangle_{\mathcal{G}}$

```
begin
    N = copy(U)
    for n ∈ N
        for s in S
            g = n^s        // n^s := s^-1 n s
            if g ∉ ⟨N⟩     // is g in the subgroup generated by N?
                push g to N
        end end
    end
    return N
end
```

<u>Proof:</u> 1) termination – If G is finite it's clear.

2) Suppose that $K = \langle N \rangle$; since $U \subseteq N \Rightarrow H \leq K$

3) every elt of N can be written as $g^{-1}ug \Rightarrow K \leq \langle\langle H \rangle\rangle_G$

4) claim: $K \trianglelefteq G$

Let $k \in K$ and $g \in G$;

$$k^g = g^{-1}kg = g^{-1}n_1 \cdots n_i g = \ldots = n_1^g \cdot \ldots \cdot n_i^g$$

if $n^g \in K$ for every $n \in N \Rightarrow k^g \in K \Rightarrow K \trianglelefteq G$.

$$n^g = n^{s_1 \cdots s_l} = (n^{s_1})^{s_2 \cdots}$$

$\Rightarrow$ It's enough to prove that $n^s \in K = \langle N \rangle$ for any $n \in N$ and any $s \in S$.

$\Rightarrow$ but that's what we precisely do in the orbit algorithm!

$\square$

<u>Application:</u>

The commutator subgroup:

$$G' = \langle\langle a^{-1}b^{-1}ab \mid a, b \in S \rangle\rangle_G$$

Finally: Pseudorandom elements

To find truly random elts in G one would need to

1) access all elts of G as a list (at arbitrary locations)

2) generate random number from $1{:}n$

and then pick the corresponding elt.

1) is infeasible (too many of them)

2) is impossible (no such hardware exist).

ALGORITHM : Pseudo-random

INPUT :
- X – a list of "sufficiently" random elts from G
- s – the seed

OUTPUT :
- X – a list of ______ . ______
- g – a random element

```
begin
    i, j – two distinct integers from 1:|X| chosen at random
    sgn = rand((-1, 1))
    if rand(Bool)
        X[i] = X[i] · X[j]^sgn
        g = s · X[i]
    else
        X[i] = X[j]^sgn · X[i]
        g = X[i] · s
    end
    return X, g
end
```

ALGORITHM: initialize

INPUT: • S - a set of generators for $G = \langle S \rangle$

OUTPUT: • X - a list of "sufficiently random" elements of G
• a - a pseudorandom group element

```
begin
  X = concatenate S with itself as long as |X| < 11
  a = e        // the accumulator //        // 11 is heuristic //
  for _ in 1:50      // also a heuristic
    X, a = pseudorandom(X, a)
  end
  return X, a
end
```

Note: This procedure corresponds to a random walk on the Schreier graph of $Aut^+(Free(n))$ action on n-generating tuples of G.

$$\underset{\langle x_1, \dots, x_n \rangle}{Aut(\overset{\shortparallel}{Free(n)})} = \langle L_{ij}^{\pm 1}, R_{ij}^{\pm 1} \mid \dots \rangle_{\substack{i=1\dots n \\ j=1,\dots,n \\ i \neq j}} =: \mathcal{S}$$

$$sgn = \pm 1 \quad L_{ij}^{sgn}(x_k) = \begin{cases} x_j^{sgn} x_i, & k = i \\ x_k, & \text{otherwise} \end{cases}$$

$$R_{ij}^{sgn}(x_k) = \begin{cases} x_i x_j^{sgn}, & k = i \\ x_k, & \text{otherwise} \end{cases}$$

analogous to E_{ij} in $SL(n, \mathbb{Z}) = Aut^+(\mathbb{Z}^n)$.

$$\Gamma_n(G) = \begin{cases} \mathcal{V} = \{n\text{-generating tuples of } G\} \\ \mathcal{E} = \{(v_1, s, s(v_2)), \text{ where } s \in \mathcal{S}\} \end{cases}$$

Schreier graph

Fast mixing property means that after relatively few steps starting from any vertex (here: X) we arrive close to the stationary measure

→ uniform measure on the vertices (assuming aperiodicity, ergodicity, ...)

→ uniformly chosen element of an "essentially uniformly chosen" n-generating tuple is close (in distribution) to uniformly chosen group element.

To learn more about this:

A. Lubotzky & I. Pak
The product replacement algorithm and Kazhdan Property (T)
J. of Amer Math Soc. 2000 vol 14, 347-363.