The general case for Coset Enumeration:

$N \triangleleft F$; $G = F/N$, $K < G$ finitely generated

$p : F \longrightarrow F/N$

Let $H = p^{-1}(K)$ (H is clearly a subgroup and $N < H$).

Note: $[F:H] = [F/N : K]$

However: Even if K is finitely generated, H doesn't have to be!

Still, we can write $H = \mathrm{Grp}\langle N, U\rangle$ for some finite set $U \subset F$.

If $N = \langle\langle V \rangle\rangle$ ← normal closure of a finite set V

$\Rightarrow (U, V)$ provides a finite description of H.

Proposition: Suppose that H is finitely generated and $[F:H] = \infty$. Then H contains no non-trivial normal subgroup of F.

$\square$

If $[F:H] < \infty \Rightarrow H$ is f.g. // Schreier generators

Since $\underset{\underset{1}{\neq}}{N} < H \Rightarrow$ either $[F:H] < \infty$, or H is not f.g.

Let finite $\mathcal{U}, \mathcal{V} \in X^*$ be given

$H = \mathrm{Grp}\langle \{[u] : u \in \mathcal{U}\}, \{[svs^{-1}] : s \in X^*, v \in \mathcal{V}\}\rangle$.

Global assumption:

- $u \in \mathcal{U}$ is freely reduced ($u \in \mathcal{C}$)
- $v \in \mathcal{V}$ is cyclicly reduced (all cyclic permutation of v are in $\mathcal{C}$).

Question: Is it possible to decide if the index of H in F is finite?

If it were, we'd set $\mathcal{U} = \emptyset$, $H = N$

$\Rightarrow [F : H]$ is the order of f. p. G

$\Rightarrow$ we'd be able to decide whether G is finite

$\Rightarrow$ but that is an algorithmically undecidable

i.e. we can only verify that $[F : H] < \infty$ (and compute it), or our algorithm will not stop.

Definition: $\mathcal{A}$ - coset automaton is compatible with $\mathcal{V}$ iff

$$\mathrm{trace}(\mathcal{A}, v, \sigma) = \sigma \quad \text{for all } \sigma \in \Sigma, v \in \mathcal{V}$$

Proposition: Let $L < F$, $N \triangleleft F$, $N = \langle\langle \mathcal{V} \rangle\rangle$. Then $N \subseteq L$ iff $\sigma[v] = \sigma$ for every coset $\sigma \in F/L$ and every $v \in \mathcal{V}$.

Proof:

$N \subseteq L$ iff L contains all conjugates of $[v]$ where $v \in V$.

Let $w \in X^*$, $v \in V$ then

$$[w][v][w]^{-1} \in L \quad \text{iff} \quad L[w][v][w^{-1}] = L, \text{ or}$$

$$\underbrace{L[w]}_{\sigma}[v] = \underbrace{L[w]}_{\sigma}.$$

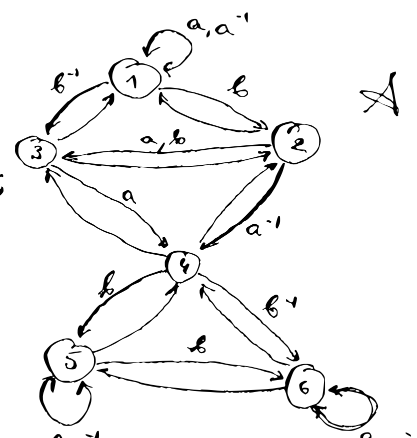
$\square$

Corollary / Restatement:

$N \subset L$ iff $\mathrm{trace}(\mathcal{A}_S(L), v, \sigma) = |v|, \sigma$ for all $v \in V$ & all $\sigma \in \mathcal{A}_S(L)$

iff $\mathcal{A}_S(L)$ is compatible with V.

Example:



$\mathcal{A}$

$F = \mathrm{FreeGrp}\langle\{a, b\}\rangle$

$L = K(\mathcal{A})$ has index 6 in F; and $\mathcal{A} = \mathcal{A}_I(L) = \mathcal{A}_S(L)$.

Let $V = \{\underbrace{abab}_{v}\}$

$\mathrm{trace}(\mathcal{A}, v, 1) = 1$

$\mathrm{trace}(\mathcal{A}, v, 2) = 2$

$\mathrm{trace}(\mathcal{A}, v, 3) = 6$

$\Longrightarrow N = \langle\langle (ab)^2 \rangle\rangle \not\subset L.$

$b^{-1} v b \in N$ but $b^{-1} v b \notin L$.

Assume $[F:H]$ is finite.

(which means that $\mathcal{A}_I(H) = \mathcal{A}_s(H)$)

Since H is f.g. (Schreier generators!)

there exists a finite

$$W' \subset W = \mathcal{U} \cup \{ s v s^{-1} : s \in X^*, v \in V \}$$

s.t. $H = Grp\langle \{ [w'] : w \in W' \} \rangle$.

Suppose that we exhaust W by finite sets W_i

- $\mathcal{U} \subseteq W_1 \subseteq W_2 \subseteq \dots \subseteq W_n \subseteq \dots \subseteq W$
- $\bigcup_i W_i = W$

Since $W' \subset W_i$ for sufficiently large i and for $L_i = Grp\langle [w] : w \in W_i \rangle$ we have $H = L_i$.

If we compute $\mathcal{A}_I(L_1), \mathcal{A}_I(L_2), \dots$ at some point $\mathcal{A}_i$ will be complete and compatible with V. Then we know that $H = L_i$.

[If $[F:H] = \infty$ we will be computing $\mathcal{A}_I(L_i)$ forever].

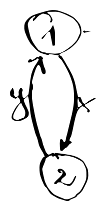
Algorithm: coset_enumeration_naive

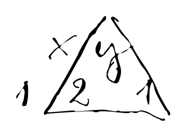
Input:
- X – alphabet with Inverses
- $\mathcal{U}$ – set of words over X^*
- $\mathcal{V}$ – ——

Output:
- $\mathcal{A}$ – important coset automaton for $H = \mathrm{Grp}\langle \mathcal{U}, \langle\langle \mathcal{V} \rangle\rangle \rangle$

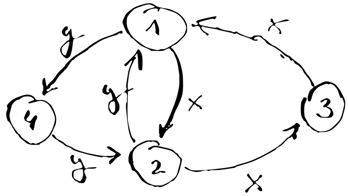
```
begin
    i = 0
    while true
        T = U ∪ { s v s⁻¹ : s ∈ X*, v ∈ V, |s| ≤ i }
        A = coset_enumeration(X, T)
        if A is complete and compatible with V
            return A
        end
        i += 1
    end
end
```

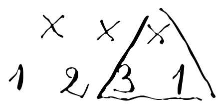
Ex: $X = \{x^{\pm 1}, y^{\pm 1}\}$; $\mathcal{U} = \{xy\}$, $\mathcal{V} = \{x^3, y^3, (xy)^3, (xy^{-1})^3\}$



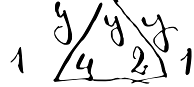


call join!($\mathcal{A}$, 2, 1, y)

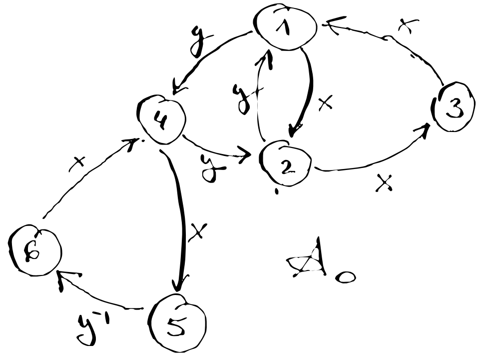




join!($\mathcal{A}$, 3, 1, x).



join!($\mathcal{A}$, 4, 2, y)



$\mathcal{A}_0$

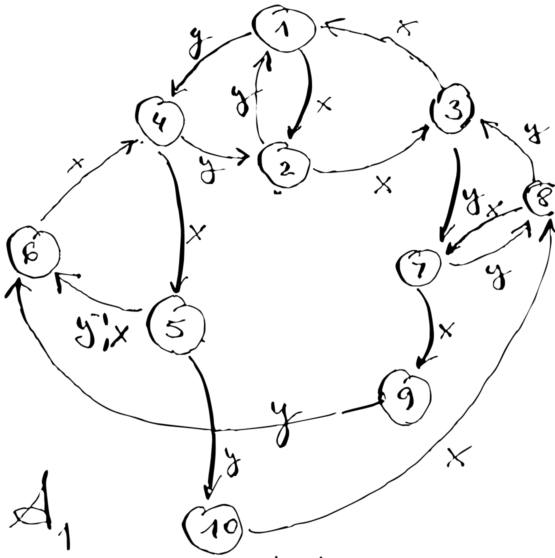
$\cancel{x^2}, \cancel{y^3}, (xy)^3, (xy^{-1})^3$

$(xy)^3$ does not change $\mathcal{A}$

$x \; y^{-1} \; x \; y^{-1} \; x \; y^{-1}$
1 2 4 5 6 4 1

$\mathcal{A}_0$ - not complete we continue with $\mathcal{A}_0$ tracing

xy^3x^{-1}, $x^{-1}y^3x$, yx^3y^{-1}, $y^{-1}x^3y$, $x(xy)^3x^{-1}$, $x^{-1}(xy)^3x^{-1}$, $y(xy)^3y^{-1}$, $y^{-1}(xy)^3y$



$\mathcal{A}_1$

$x \; x \; y^{-1} \; x \; y^{-1} \; x \; y^{-1} \; x^{-1}$
1 2 3 8 $\boxed{4}$ 3 1 2 1
7 coincidence nodes
(8, x, 7)

$x^{-1} \; x \; y^{-1} \; x \; y^{-1} \; x \; y^{-1} \; x$ traces
1 3 1 2 3 8 7 3 1

$y(xy^{-1})^3y$ traces

$y^{-1}(xy^{-1})^3y$ traces

$x \; y \; y \; y \; x^{-1}$
1 2 1 4 2 1

$x^{-1} \; y \; y \; y \; x$
1 3 7 8 3 1

$y \; x \; x \; x \; y^{-1}$
1 4 5 9 4 1
6 + call to coincidence

$y^{-1} \; x \; x \; x \; y$
1 2 3 1 2 1 ✓

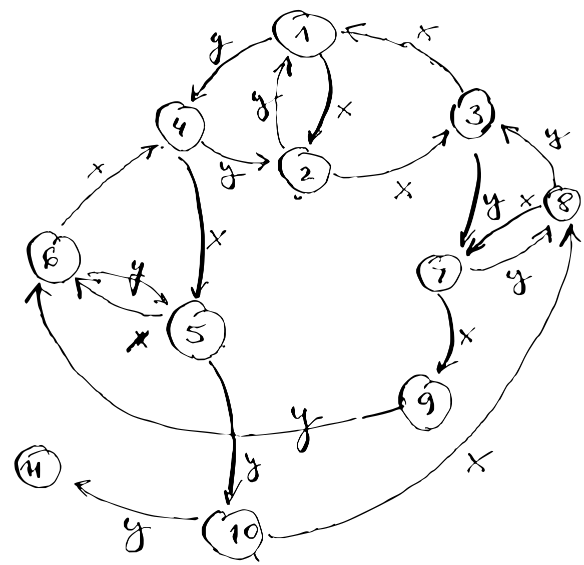
$x \; x \; y \; x \; y \; x \; y \; x^{-1}$
1 2 3 7 9 6 4 2 1

$x^{-1} \; x \; y \; x \; y \; x \; y \; x$
1 3 1 4 5 10 8 3 1

$y \; x \; y \; x \; y \; x \; y \; y^{-1}$
1 4 5 10 8 3 1 4 1 ✓

$y^{-1} \; x \; y \; x \; y \; x \; y \; y$
1 2 3 7 9 6 4 2 1 ✓

A_1 is not complete so we continue...



$xx\,yyy\,x^{-1}x^{-1}$ traces

$x^{-1}x^{-1}\,yyy\,xx$ traces

$yy\,xxx\,y^{-1}y^{-1}$ traces

$y^{-1}y^{-1}\,xxx\,yy$ traces

$xy\,xxx\,y^{-1}x^{-1}$ traces

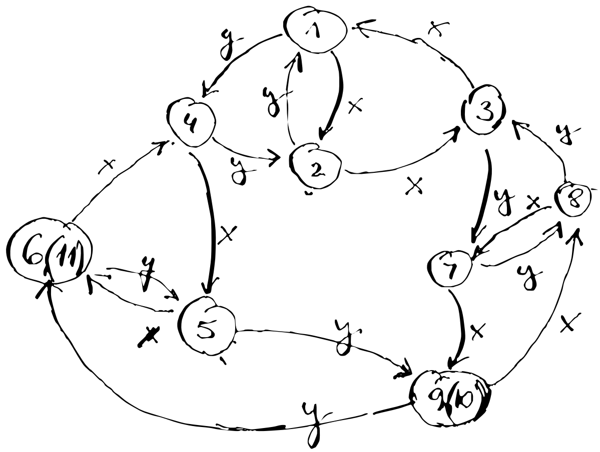
$y^{-1}x^{-1}\,xxx\,xy$ traces

$yx\,yyy\,x^{-1}y^{-1}$

$\begin{matrix} 1 & 4 & 5 & 10 & 11 & 5 & 4 & 1 \\ & & & & 6 & & & \end{matrix}$

coincidence identifies 11 and 6

$\begin{pmatrix} 11 & y^{-1} & 10 \\ 6 & y^{-1} & 9 \end{pmatrix}$ $10 \approx 9$



A_2.

$xx\,xyxyxy\,x^{-1}x^{-1}$ traces

$x^{-1}x^{-1}\,xyxyxy\,xx$ traces

$yy\,xyxyxy\,y^{-1}y^{-1}$ traces

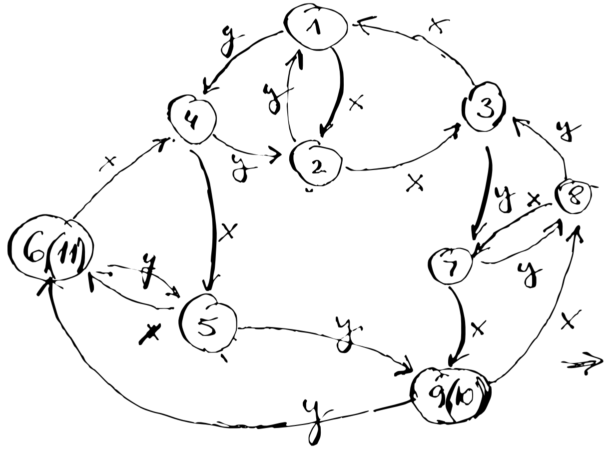
$y^{-1}y^{-1}(xy)^3\,yy$ traces

$xy(xy)^3\,y^{-1}x^{-1}$ traces

$yx(xy)^3\,x^{-1}y^{-1}$ traces

xy^{-1}

$y^{-1}x$ and their inverses as well...



This guy is complete and compatible with σ.

$\Rightarrow$ we can compute with cosets of H!

there are 9 states/cosets $\Rightarrow$ $[F : H] = [G : K] = 9$

the order of K is at most 3 since

$K = \langle xy \rangle < G$, $(xy)^3 = 1$ in G.

$F \xrightarrow{\varphi} \mathrm{Sym}(\mathcal{A}_2)$

$x \longmapsto (1,2,3)(4,5,6)(7,9,8)$

$y \longmapsto (1,4,2)(3,7,8)(5,9,6)$

$\varphi(xy) = (1)(2,7,6)(3,4,9)(5)(8)$

has order 3

hence xy in G has order at least 3.

$\Rightarrow |G| = [G : K] \cdot |K| = 27.$

Problems with coset_enumeration_naive:

- The size of $\{[s v s^{-1}] : v \in V, |s| < k\}$ grows exponentially with k

- We throw away each automaton when we start anew

Suppose $\mathcal{A} = (\Sigma, X, E, \{\alpha\}, \{\alpha\})$ is a coset automaton

If $\text{trace}(\mathcal{A}, s, \alpha) = \sigma$ and $\text{trace}(\mathcal{A}, v, \sigma) = \sigma$

then $[s v s^{-1}] \in K(\mathcal{A})$.

$\Rightarrow$ $K(\mathcal{A})$ contains all $[s v s^{-1}]$ for $v \in V$, $|s| \leq i$ when

- s is traceable in $\mathcal{A}$
- $\text{trace}(\mathcal{A}, v, \sigma) = \sigma$ for every state $\sigma \in \mathcal{A}$ which can be reached from α by a path of length $\leq i$.

General coset enumeration scheme:

1) $\mathcal{A}$ = CosetAutomaton($\mathcal{X}$)

2) for $u \in \mathcal{U}$ trace_and_reverse!($\mathcal{A}$, u)

Execute any sequence of those steps

- pick $\sigma \in \Sigma$, $x \in X$
  if !hasedge($\mathcal{A}$, σ, x)
    define!($\mathcal{A}$, σ, x)
  end

- pick $\sigma \in \Sigma$, $v \in \mathcal{V}$
  call trace_and_reverse($\mathcal{A}$, v, define = true/false)

- if $\mathcal{A}$ is complete and compatible with $\mathcal{V}$
  return $\mathcal{A}$

We want the sequence to satisfy three conditions:

1) if termination is possible, then it happens
2) either a state is σ deleted from $\mathcal{A}$, or
3) it becomes complete at some point, and
   trace($\mathcal{A}$, v, σ) = σ for all $v \in \mathcal{V}$.

Proposition: If our general coset enumeration terminates, then $K(\mathcal{A}) = H$.

Proof: We begin with $K(\mathcal{A}) = \mathrm{Grp}\langle [u] : u \in \mathcal{U} \rangle < H$.

- defines don't change $K(\mathcal{A})$
- trace_and_reverse adds $[s v s^{-1}]$ to generators of $K(\mathcal{A})$
  $\in H$.

If we terminate, then
$\mathcal{A}$ is compatible with $\mathcal{V}$, then $N < K(\mathcal{A})$
$\Rightarrow K(\mathcal{A}) = H$. $\square$

Proposition: If $[F:H] < \infty$ then any general coset enumeration terminates.

Proof:

Suppose that $[F:H]$ is finite, but coset enumeration doesn't terminate.

Claim: for every $s \in X^*$ $trace(\mathcal{A}, s, \alpha)$ is eventually successful.

Induction on $|s|$:

0) $trace(\mathcal{A}, \varepsilon, \alpha)$ is defined

n) suppose that $trace(\mathcal{A}, s, \alpha)$ is defined for all $|s| < n$

n+1) consider $w = sx$, $x \in X$

- neither define nor join change $\sigma = trace(\mathcal{A}, s, \alpha)$
- coincidence! may change σ, but only finitely many times
- afterwards it eventually becomes complete making $trace(\mathcal{A}, sx, \alpha)$ successful.

Note: every $s \in X^*$ $[s v s^{-1}]$ eventually belongs to $K(\mathcal{A})$

Once $\sigma = trace(\mathcal{A}, s, \alpha)$ is successful property 3) implies that $trace(\mathcal{A}, v, \sigma) = \sigma$ so $[s v s^{-1}] \in K(\mathcal{A})$.

Since $[F:H]$ is finite, H is finitely generated (Schreier generators!) so eventually

$T_k = \mathcal{U} \cup \{ [s v s^{-1}] : v \in \mathcal{V}, |s| < k \}$ generates H.

Then $K(\mathcal{A}_k) = H$, thus $\mathcal{A}_k$ is complete and compatible with $\mathcal{V}$, so that we terminate. $\square$.

HLT (Haselgrove, Leech, Trotter) strategy:

"Define new states as we go"

Algorithm: coset_enumeration_hlt

Input:
- X – alphabet with inverses
- $\mathcal{U}$ – set of words over X
- $\mathcal{V}$ – ——— " ———

Output: $\mathcal{A}$ – coset automaton for $\mathcal{U}$ compatible with $\mathcal{V}$ (If terminates: $\mathcal{A}_J(H)$, $H = \mathrm{Grp}\langle \mathcal{U}, \langle\langle \mathcal{V} \rangle\rangle \rangle$).

```
begin
    A = CosetAutomaton(X)
    for u in U
        trace_and_reverse!(A, u)
    end
    for σ in states(A)
        for v ∈ V
            trace_and_reverse!(A, σ, σ)
            if find(A.partition, σ) ≠ σ
                break
            end
        end
        if find(A.partition, σ) = σ
            for x in X
                if !hasedge(A, σ, x)
                    define!(A, σ, x)
                end
            end
        end
    end
    return A
end
```

The Felsch strategy:

"As long as we can progress further try not to define new states"

1) Don't define new states tracing elements of $\mathcal{V}$

2) Try to complete states one by one, in the order they were defined, with some fixed order on X

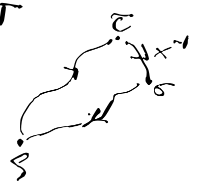
The idea for implementation:

- use a stack of newly defined edges

  if (σ, x, τ) was recently added

  then the only unsuccessful two-sided traces that may be completed are

where • $v = SxT$ and $\text{trace}(\mathcal{A}, S, \rho) = \sigma$.



• $v = Cx^{-1}D$ and $\text{trace}(\mathcal{A}, D, \sigma) = \rho$ // for some ρ.

- instead of tracing SxT from ρ it's enough to trace xTS from σ.

- whenever a new edge (σ, x, τ) is added put it on the stack and try to trace from σ all cyclic perms of $v \in \mathcal{V}$ that begin with x.

Algorithm : deduce!

Input:
- $\mathcal{A}$ – coset automaton
- W – set of cyclic perms of $v \in V$
- stack – stack of newly added edges

Output:
- $\mathcal{A}$ – with deducable traces of $w \in W$ defined

```
begin
    while !isempty(stack)
        (σ, x, τ) = pop!(stack)
        if find!(A.partition, σ) = σ
            for w ∈ W
                if w[begin] = x
                    trace_and_reverse!(A, w, σ,
                                       define=false)
                end
            end
        end
    end
    return A
end
```

this needs to be repeated for (τ, x^{-1})

Algorithm : coset_enumeration_felsch
Input : X, $\mathcal{U}$, $\mathcal{V}$ // as previously
Output : $\mathcal{A}$ // as previously

```
begin:
    W - the set of cyclic perms of σ ∈ V
    stack = []
    for u ∈ U
        trace_and_reverse!(A, u, stack)
        deduce!(A, W, stack)
    end
    for σ in states(A)
        for x in X
            if !hasedge(A, σ, x)
                define!(A, σ, x, stack)
            end
        end
        deduce!(A, W, stack)
    end
    return A
end
```

this version passes stack to define!, join! coincidence!

a new version of define! that pushes (σ, x, τ) onto the stack

Notes:

- define, join, coincidence must be modified to push added edges to stack
- stack is small (1 elt for define! and join!), but may explode in size after coincidence! → in such cases it's better to trace every element of $\mathcal{V}$ on each state $\sigma \in \Sigma$.