Setting: $F = \mathrm{Mon}\langle X \mid \mathrm{FGRel}(X)\rangle$

$u \in X^* \Rightarrow [u] \in F$

$\overline{u} \in \mathcal{C}$ can be computed using $\mathcal{R}$.
(free reduction)

$$\mathcal{A}_S = \mathcal{A}_S(H) = (\Sigma_S, X, E_S, \{H\}, \{H\})$$

right cosets

$(H[u], x, H[u][x])$

Schreier Automaton

$$\mathcal{L}_S(H) = \{ u \in X^* : [u] \in H \}$$

Proposition: • $\mathcal{A}_S$ is complete & trim.

• $\mathcal{L}(\mathcal{A}_S) = \mathcal{L}_S(H)$ i.e. $\mathcal{L}_S$ is rational iff H is of finite index.

$$\mathcal{L}_C = \mathcal{L}_C(H) = \mathcal{L}_S(H) \cap \mathcal{C}$$

the set of canonical forms for $h \in H$
(here: freely reduced ones).

Proposition: If H is finitely generated, then $\mathcal{L}_C$ is rational.

Proof: If H - f.g as a grp $\Rightarrow$ H f.g. as a monoid

pick $\mathcal{U} \subset X^*$ finite, s.t. $\langle[\mathcal{U}]\rangle = H$.
"monoid generating set".

Every elt of H contains an element of $\mathcal{U}^*$ $\Rightarrow$

$\mathcal{L}_C(H) = \overline{\mathcal{U}^*} = \{ \overline{w} : w \in \mathcal{U}^* \}$ is rational

Proposition:

If $\mathcal{U} \subset X^*$ s.t. $H = \langle [\mathcal{U}] \rangle$, finite,

then we may assume:

- $u \in \mathcal{U}$ is freely reduced
- no $u \in \mathcal{U}$ is a proper prefix of $u' \in \mathcal{U}$.

$\mathcal{A} = (\Sigma, X, E, \{\varepsilon\}, \{\varepsilon\})$ recognizes $\mathcal{U}^*$ where

- Σ - set of proper prefixes of words in $\mathcal{U}$
- $E = \{ (u, x, ux)$ s.t. $u, ux \in \Sigma \} \cup$

  $\cup \{(u, x, \varepsilon)$ s.t. $u \in \Sigma$, $ux \in \mathcal{U}\}$.

To proceed further with $\overline{\mathcal{U}^*}$

for every pair of states σ, τ of $\mathcal{A}$ if $\exists$ P - path in $\mathcal{A}$ from σ to τ that is the lhs from R add $(\sigma, \varepsilon, \tau)$ to edges of $\mathcal{A}$ (there are only f. many edges to add

$\mathcal{A}'$ recognizes $\mathcal{U}^*$ and every derived (w.r.t R) word from $\mathcal{U}^*$

$\Rightarrow$ $\mathcal{L}(\mathcal{A}') \supset$ all freely reduced words that are in H.

Proposition: Suppose that $\mathcal{L}_c(H)$ is rational.

we can perform the membership test for H and the coset equality test.

Proof:

Given $u \in X^*$

- compute $\overline{u} \in \mathcal{C}$ - freely reduced
- check if $\overline{u} \in \mathcal{L}_c(H)$.

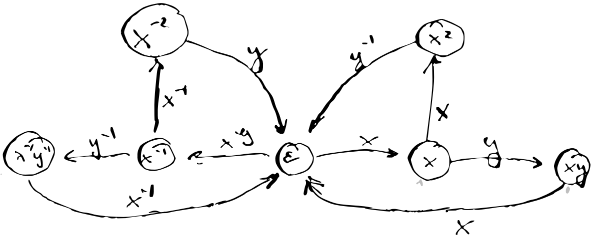
  ↳ construct an automaton for $\mathcal{L}_c(H)$

$$H[u] = H[v] \Leftrightarrow \overline{uv^{-1}} \in \mathcal{L}_c(H). \square$$

$\underline{\text{Ex:}}$ $X = \{x^{\pm 1}, y^{\pm 1}\}$ $\quad \mathcal{R} = \text{FGRel}(X)$

$H = \langle [xyx], [x^{-2}y] \rangle$

$\mathcal{U} = \{ xyx, \; x^{-1}y^{-1}x^{-1}, \; x^{-2}y, \; x^{2}y^{-1} \}$



consider:

$\underline{\underline{\boxed{\varepsilon}}}$ $\quad (x^{-1}y^{-1}) \xrightarrow{x^{-1}} \boxed{\varepsilon} \xrightarrow{x} (x) \Rightarrow (x^{-1}y^{-1}, \varepsilon, x)$

$(xy) \xrightarrow{x} \boxed{\varepsilon} \xrightarrow{x^{-1}} (x^{-1}) \Rightarrow (xy, \varepsilon, x^{-1})$

$ms = [xy, x^{-1}y^{-1}]$

$(xy) : \boxed{x} \xrightarrow{y} \boxed{xy} \xrightarrow{\varepsilon} (x^{-1}) \xrightarrow{y^{-1}} (x^{-1}y^{-1})$

$\Rightarrow (x, \varepsilon, x^{-1}y^{-1})$

$ms = [x^{-1}y^{-1}, x]$

$(x^{-1}y^{-1}) : \boxed{x^{-1}} \xrightarrow{y^{-1}} (x^{-1}y^{-1}) \xrightarrow{\varepsilon} (x) \xrightarrow{y} (xy)$

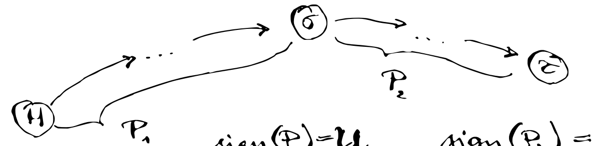
$ms = [x, x^{-1}]$

$\Rightarrow (x^{-1}, \varepsilon, xy)$

processing x, x^{-1} will not give any other new edges.

Defn:

right coset $\sigma = H[\mathcal{U}]$ is called important if $\mathcal{U}$ is a prefix of element of $\mathcal{L}_c(H)$.



$\mathrm{sign}(P) = \mathcal{U}$ $\quad$ $\mathrm{sign}(P_2) = \mathcal{V}$

$\mathcal{U} \cdot \mathcal{V} \in \mathcal{L}_c(H)$

- $(\mathcal{U}, \mathcal{V})$ - defining pair for σ
- $\mathcal{A}_I = (\Sigma_I, X, E_I, \{H\}, \{H\})$

  the important coset automaton:

  restriction of $\mathcal{A}_s$ to $\Sigma_I \subset \Sigma_s$

Proposition: $\mathcal{A}_I(H)$ is fin and

$$\mathcal{L}_c(H) \subseteq \mathcal{L}_I(H) \subseteq \mathcal{L}_s(H).$$

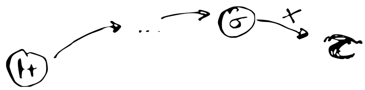
Proposition: $\Sigma_I(H)$ is finite iff H is f.g.

Proof:

(⇒) Suppose Σ_I - finite; for each $\sigma \in \Sigma_I$ let $(\mathcal{U}_\sigma, \mathcal{V}_\sigma)$ be a defining pair.

for each $x \in X$ s.t. $\tau = \underbrace{\mathrm{trace}(\Sigma_I, x, \sigma)}_{\sigma x}$ is defined

Let $Y(\sigma, x) = \mathcal{U}_\sigma x (\mathcal{U}_\tau)^{-1}$



Since $\sigma = \text{trace}(\Sigma_I, U_\sigma, H)$

$$\text{trace}(\Sigma_I,\ x,\ \sigma) = \text{trace}(\Sigma_I,\ U_\sigma \cdot x,\ H)$$

$$\Rightarrow \text{trace}(\Sigma_I,\ Y(\sigma,x),\ H) = H.$$

Choose $U_H = \varepsilon$, and let $u \in \mathcal{L}_c(H)$,

$u = x_1 \cdot \ldots \cdot x_t \qquad x_i \in X \qquad [u] \in H \Rightarrow U_{\sigma_t} = \varepsilon$

$\sigma_0 = H; \quad \sigma_i = \text{trace}(\Sigma_I,\ x_i,\ \sigma_{i-1}) \quad i = 1, \ldots, t.$

$$Y(\sigma_0, x_1) = U_{\sigma_0} x_1 (U_{\sigma_0 x_1})^{-1} = \varepsilon \cdot x_1 (U_{\sigma_1})^{-1}$$

$$Y(\sigma_1, x_2) = U_{\sigma_1} x_2 (U_{\sigma_2})^{-1}$$

$$Y(\sigma_0, x_1) \cdot Y(\sigma_1, x_2) = \varepsilon\, x_1 (U_{\sigma_1})^{-1} \cdot U_{\sigma_1} x_2 (U_{\sigma_2})^{-1}$$
$$= x_1 x_2 \cdot (U_{\sigma_2})^{-1}$$

$$Y(\sigma_0 \cdot x_1) \cdot \ldots \cdot Y(\sigma_{t-1}, x_t) =$$
$$= \varepsilon \cdot x_1 \cdot x_2 \cdot \ldots \cdot x_t (U_{\sigma_t})^{-1} = x_1 \cdot x_2 \cdot \ldots \cdot x_t.$$

Since every $[u] \in H$ contains $\bar{u} \in \mathcal{L}_c(H)$ we've written all elts in H as a product of $\{ Y(\sigma, x) \mid \sigma \in \Sigma_I$ and $\text{trace}(\Sigma_I, x, \sigma)$ is defined $\}$.

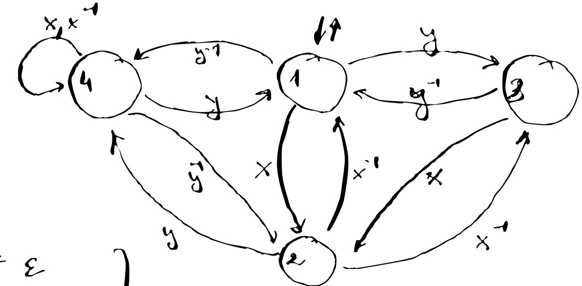
Corollary:

$\mathcal{L}_c(H)$ is rational iff H is f.g.

Example:

$F = FGrp(\{x,y\})$, $X = \{x^{\pm 1}, y^{\pm 1}\}$

Let $\mathcal{A} =$



$$\left.\begin{array}{ll} u_1 = \varepsilon , & v_1 = \varepsilon \\ u_2 = x , & v_2 = xy^{-1} \\ u_3 = y , & v_3 = x^{-1}x^{-1} \\ u_4 = y^{-1} , & v_4 = xy \end{array}\right\} \text{defining pairs}$$

$Y(1,x) = \varepsilon \cdot x \cdot (u_2)^{-1} = xx^{-1}$	$Y(3,x^{-1}) = yx^{-1}x^{-1}$
$Y(1,y) = \varepsilon \cdot y \cdot (u_3)^{-1} = yy^{-1}$	$Y(3,y^{-1}) = yy^{-1}\varepsilon$
$Y(1,y^{-1}] = \varepsilon \cdot y^{-1} \cdot (u_4)^{-1} = y^{-1}y$	$Y(4,x) = y^{-1}xy$
$Y(2,x) = u_2 \cdot x \cdot (u_3)^{-1} = x \cdot xy^{-1}$	$Y(4,y) = y^{-1}y\varepsilon$
$Y(2,x^{-1}) = u_2 \cdot x^{-1}(u_1)^{-1} = x \cdot x^{-1} \cdot \varepsilon$	$Y(4,x^{-1}) = y^{-1}x^{-1}y$
$Y(2,y) = u_2 \cdot y\,(u_4)^{-1} = xy^2$	$Y(4,y^{-1}) = y^{-1}y^{-1}x^{-1}$

If $\mathcal{A} = \mathcal{A}_I(H)$ then $H < F$,

$H = Mon\langle x^2y^{-1}, xy^2, yx^{-2}, y^{-1}xy, y^{-1}x^{-1}y, y^{-2}x^{-1}\rangle$

$H = Grp\langle x^2y^{-1}, xy^2, y^{-1}xy\rangle$.

Proposition:

$[F:H] < \infty$ iff $\mathcal{A}_I(H)$ is finite and complete.

Proof:

$(\Rightarrow)$ $[F:H] < \infty \Rightarrow \Sigma_s(H)$ finite $\Rightarrow \Sigma_I(H)$ finite.

Aim: $\Sigma_I(H) = \Sigma_s(H)$.

Let $u \in \mathcal{C}$. we need to show that
$H \cdot [u] \in \Sigma_I$.

For each $\sigma \in \Sigma_s$ let u_σ be s.t. $H \cdot [u_\sigma] = \sigma$.

Let $m = \max_\sigma |u_\sigma|$.

let $W = u \cdot V \in \mathcal{C}$ s.t. $|V| \geq m$

let $\sigma = H[u \cdot V]$; $S = \overline{u \cdot V \cdot W_\sigma}$ (note $H[S] = H$).

Note : since $|V| \geq |W_\sigma|$ u is a prefix of S.

Write: $S = u \cdot T \Rightarrow (u, T)$ is a defining pair for $H[u] \Rightarrow H[u] \in \Sigma_I$.

Proposition:

Let $u, v \in \mathcal{C}$; Let $u = BR$, $v = CS$ s.t. B, C - the longest prefixes of u, v such that $H[B]$ and $H[C]$ belong to $\Sigma_I(H)$

Then $H[u] = H[v]$ iff $H[B] = H[C]$ & $R = S$.

Proof:

Assume $H[u] = H[v]$ i.e. $[uv^{-1}], [vu^{-1}] \in H$.

If u and v don't end with the same element $\Rightarrow$ $\left.\begin{matrix} uv^{-1} - \text{irreducible} \\ vu^{-1} - \text{"} \text{---} \end{matrix}\right\} \in \mathcal{C}$

$\Rightarrow uv^{-1} \in \mathcal{L}_c(H) \Rightarrow (u, v^{-1})$ is the defing pair for $H[u]$ i.e. $H[u] \in \Sigma_I$;

i.e. $B = u$, $C = v$, $R = S = \varepsilon$.

Suppose that $u = u_1 x$ & $v = v_1 x$

$\Rightarrow H[u_1] = H[u][x^{-1}] = H[v][x^{-1}] = H[v_1]$.

$\left.\begin{matrix} u_1 = B_1 R_1 \\ v_1 = C_1 S_1 \end{matrix}\right\}$ analogously as above.

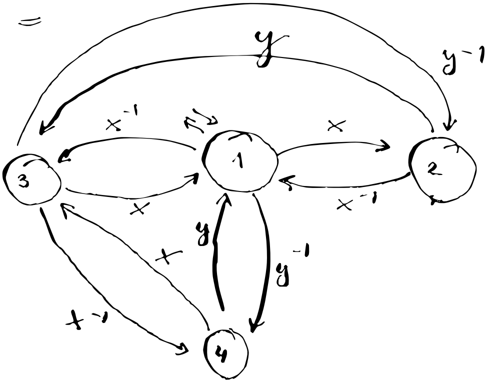
Claim: Either $B = B_1$ or $B_1 = u_1$ and $B = u$

In either case the conclusion follows.

Ex: $F = \mathrm{FGrp}\langle x, y\rangle$

$H = \mathrm{Grp}\langle [xyx], [x^{-2}y]\rangle$

$A_I(H) =$



$$\left.\begin{array}{ll} u_1 = \varepsilon, & s_1 = \varepsilon \\ u_2 = x, & s_2 = yx \\ u_3 = x^{-1}, & s_3 = x^{-1}y \\ u_4 = y^{-1}, & s_4 = x^2 \end{array}\right\}\text{ defining pairs}$$

Let $u = \underbrace{xyx^{-1}yx^{-1}y^{-1}}_{B},\underbrace{y^{-1}xy}_{R},$

$v = \underbrace{y^{-1}xy^{-1}x^{-1}x^{-1}x^{-1}yx}_{C}\underbrace{xy}_{S},$

both traces end at $\sigma = 2$, so

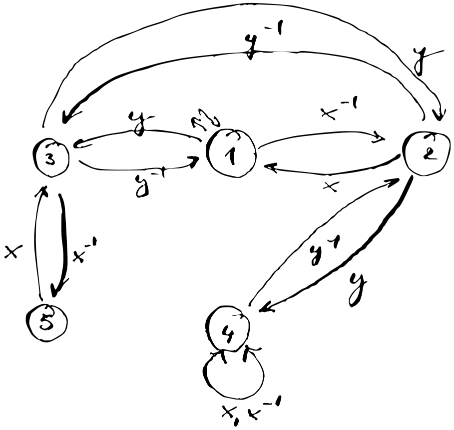
$H[B] = H[C]$, but $R \neq S \Rightarrow$

$H[u] \neq H[v]$.

Coset automata:

Defn: $\mathcal{A} = (\Sigma, X, E, A, \Omega)$ automaton over alphabet X; $\mathcal{A}$ is a coset automaton relative to $\mathcal{R}$ - rws if

- $\mathcal{A}$ - accessible & deterministic
- $A = \Omega \neq \emptyset$
- If $(\sigma, x, \tau) \in E \Rightarrow (\tau, x^{-1}, \sigma) \in E$



Ex: $\mathcal{A}_s(H)$, $\mathcal{A}_I(H)$ are coset automata

Prop:
- Coset automata are trim.
- In coset automaton if $\sigma^{u} = \tau$ then $\sigma^{\bar{u}} = \tau$.

Let $K(\mathcal{A}) = \{[u] : u \in \mathcal{L}(\mathcal{A})\} \subset F$

Proposition: If $\mathcal{A}$ is a coset automaton, then $K(\mathcal{A}) < F$ is a subgroup; If $\mathcal{A}$ is finite $\Rightarrow$ $K(\mathcal{A})$ is f.g.

__Defn:__ Let $\mathcal{A} = (\Sigma, X, E, A, \Omega)$ be a coset aut.

$\mathcal{A}$ is __reduced__ iff every $\sigma \in \Sigma \setminus A$

σ has at least two out-neighbours.

Proposition:

Let $\mathcal{A}$ be a finite coset automaton;

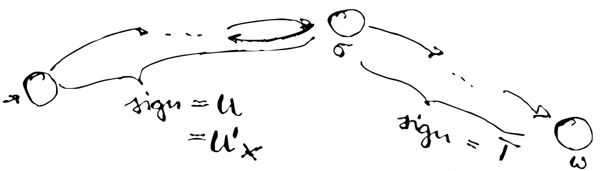
$\mathcal{A} = \mathcal{A}_I(K(\mathcal{A}))$ iff $\mathcal{A}$ is reduced.

Let's see that $\mathcal{A}_I(K(\mathcal{A}))$ is reduced.

(Aim: every state has at least two outgoing edges)

Let $\sigma \in \Sigma_I$, (U, T) its defing pair;

if $\sigma \notin A \Rightarrow U \neq \varepsilon \neq T$



Since $UT = U'xyT' \in C$

$x^{-1} \neq y$

$\Rightarrow (\sigma, y, \tau), \ (\sigma, x^{-1}, \tau')$ are edges starting at σ.

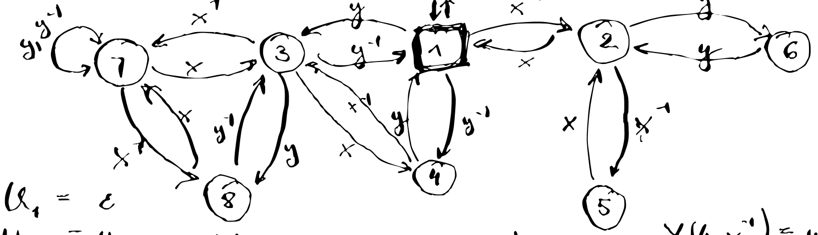
Algorithm: reduce

Input : $\mathcal{A}$ – finite coset automaton

Output: $\mathcal{B}$ – a finite, reduced cosetautomaton

$K(\mathcal{A}) = K(\mathcal{B})$.

```
begin:
    B = A
    reduced = false
    while !reduced
        reduced = true
        for σ in states(B)
            if !isinitial(σ)
                if outdegree(σ) = 1
                    # (σ, x, τ) is the only edge
                    delete!(B, (σ, x, τ))
                    delete!(B, (τ, x^{-1}, σ))
                    delete!(B, σ)
                    reduced = false
                end
            end
        end
    end
    return B
end
```



$U_1 = \varepsilon$

$U_3 = y$

$U_4 = y^{-1}$

$U_7 = y x^{-1}$

$U_8 = y y$

$Y(1, y) = \varepsilon \cdot y \cdot U_3^{-1} = \varepsilon$

$Y(1, y^{-1}) = \varepsilon \cdot y^{-1} \cdot U_4^{-1} = \varepsilon$

$Y(3, x) = U_3 \cdot x \cdot U_4^{-1} = y x y$

$Y(3, x^{-1}) = U_3 \cdot x^{-1} \cdot U_7^{-1} = y x^{-1} x y = \varepsilon$

$Y(3, y) = U_3 \cdot y \cdot U_8^{-1} = y y y^{-2} = \varepsilon$

$Y(3, y^{-1}) = U_3 \cdot y^{-1} \cdot U_1^{-1} = \varepsilon$

$Y(4, x^{-1}) = y^{-1} x^{-1} y^{-1}$

$Y(4, y) = y^{-1} y \varepsilon$

$Y(7, x) = y x^{-1} x y^{-1} = \varepsilon$

$Y(7, x^{-1}) = y x^{-1} x^{-1} y^{-1} y^{-1}$

$Y(7, y) = y x^{-1} y x y^{-1}$

$Y(7, y^{-1}) = y x^{-1} y^{-1} x y^{-1}$

$Y(8, x) = y y x x y^{-1}$

$Y(8, y^{-1}) = y y y^{-1} y^{-1} = \varepsilon$

Coset enumeration:

F - free group

$H < F$ - subgroup

Task: Compute $\mathcal{A}_I(H)$

If $[F:H]$ is finite then every coset is important $\Rightarrow$ we need to list them all.

Algorithm: Define!

Input : $\mathcal{A}$ - coset automaton
σ - state in Σ
l - letter of the alphabet X

Output: $\mathcal{A}$ with added edges (σ, l, τ), (τ, l^{-1}, σ) for new state τ.

```
begin
    $\tau$ = addstate!($\mathcal{A}$)
    add edge! ($\mathcal{A}$, ($\sigma$, $l$, $\tau$))
    add edge! ($\mathcal{A}$, ($\tau$, $l^{-1}$, $\sigma$))
    return $\mathcal{A}$
end
```

Proposition: Suppose that σ^l is not defined in $\mathcal{A}$.

then $K(\mathcal{A}) = K(\text{Define}(\mathcal{A}, \sigma, l))$.

Algorithm: join!

Input:
- $\mathcal{A}$ - coset automaton
- σ - state
- l - letter
- τ - state

Output: $\mathcal{A}$ where (σ, l, τ); (τ, l^{-1}, σ) are defined.

```
begin
    assert !hasedge($\mathcal{A}$, $(\sigma, l)$) & !hasedge($\mathcal{A}$, $(\tau, l^{-1})$).
    add $(\sigma, l, \tau)$ to edges of $\mathcal{A}$
    if $\sigma \neq \tau$ || $l \neq l^{-1}$        this is always true in free group!
        add $(\tau, l^{-1}, \sigma)$ to edges of $\mathcal{A}$
    end
    return $\mathcal{A}$
end
```

Proposition:

Let $H = K(\mathcal{A})$; $u \in X^*$ s.t. $\alpha^u = \sigma$

$v \in X^*$ s.t. $\tau^v = \alpha$

$$K(\text{join!}(\mathcal{A}, \sigma, x, \tau)) = Grp\langle H, [u \times v]\rangle$$

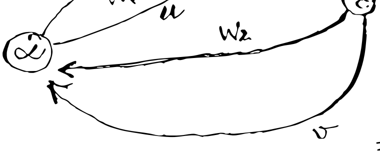
Proof:

After join! $\alpha^{u \times v} = \sigma^{\times v} = \tau^v = \alpha$

$\Rightarrow u \times v \in L(\mathcal{A})$ and $u \times v \in K(\mathcal{A})$.

Now suppose $\alpha^W = \alpha$ for some W

- if none of edges is equal to $(\sigma, x, \tau) \Rightarrow [W] \in H$.



Let $W = W_1 \times W_2$ (single occurrence)

$[W_1 u^{-1}] \in Grp\langle H, [u \times v]\rangle$

$[v^{-1} W_2] \in Grp\langle H, [u \times v]\rangle$

$\Rightarrow W_1 \times W_2 = \underline{W_1 u^{-1}}\,\overline{u \times v}\,\underline{v^{-1} W_2} \in Grp\langle H, [u \times v]\rangle$ + induction.

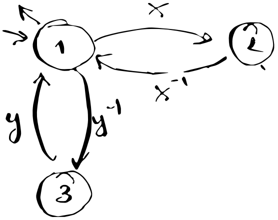
Example:

$$\boxed{\begin{array}{l} xX \to \varepsilon \\ Xx \to \varepsilon \end{array}} \quad x^2 \to \varepsilon$$

$yY \to \varepsilon$

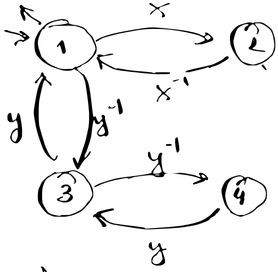
$Yy \to \varepsilon$

$\mathcal{A} =$



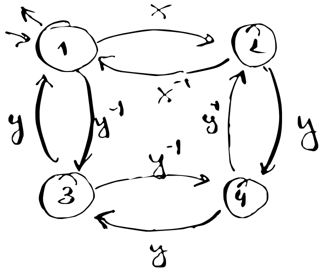
define!$(\mathcal{A}, 3, y^{-1})$

$\mathcal{A} =$



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

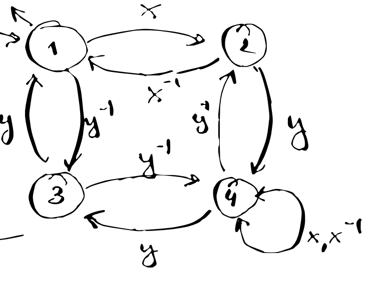
$\mathcal{A} =$



we've added $xyyy$ to $K(\mathcal{A})$!

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

$\mathcal{A} =$



we've added $xyxy^{-1}x^{-1}$ to $K(\mathcal{A})$.

Congruence on $\mathcal{A} = (\Sigma, X, E, \{\alpha\}, \{\alpha\})$
(coset automaton)

is a $\approx$ on Σ s.t. if

$(\sigma, x, \tau), (\varphi, x, \psi) \in E$ & $\sigma \approx \varphi$, then

$\tau \approx \psi$

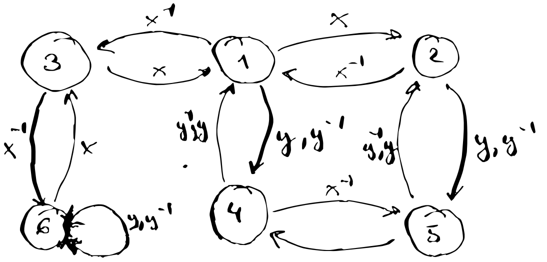
(Aim: Quotient automaton!).

Λ - set of $\approx$-classes

$\Delta = \{([\sigma], x, [\tau]) : (\sigma, x, \tau) \in E\}$

$\mathcal{B} = (\Lambda, X, \Delta, \{[\alpha]\}, \{[\alpha]\})$

Proposition: $\mathcal{B}$ is a coset automaton

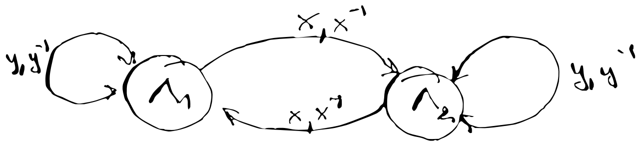


Let's begin by proclaiming $1 \approx 4$

$\left.\begin{matrix}(1, x, 2)\\ (4, x, ???)\end{matrix}\right\} \Rightarrow$ nothing $\quad \left.\begin{matrix}(1, y, 4)\\ (4, y, 6)\end{matrix}\right\} \Rightarrow 4 \approx 6$

$\left.\begin{matrix}(1, x^{-1}, 3)\\ (4, x^{-1}, 5)\end{matrix}\right\} \Rightarrow 3 \approx 5 \quad \left.\begin{matrix}(5, y, 2)\\ (2, y, 5)\end{matrix}\right\} \Rightarrow 2 \approx 5$

$\Lambda_1 = \{1, 4, 6\}, \ \Lambda_2 = \{2, 3, 5\}$



Algorithm: coincidence!

Input : • $\mathcal{A}$ – coset automaton
: σ } – states of $\mathcal{A}$
: τ }

Output : $\mathcal{A}$ – Qotient automaton of $\approx$ generated by (σ, τ).

```
begin
    deleted = {}        // deleted states
    tmpE = {}           // edges that need to be processed
    $\Lambda$ = partition of $\Sigma$ with singletons
    $\lambda$ = union!($\Lambda$, $\sigma$, $\tau$) // return the representative
    push!(deleted, $\lambda = \sigma ? \tau : \sigma$)
    while !isempty(deleted)
        $\nu$ = pop!(deleted)
        for $x \in X$
            if hasedge($\mathcal{A}$, $\nu$, x)
                $\mu$ = trace($\mathcal{A}$, x, $\nu$)).
                remove ($\nu$, x, $\mu$) from $\mathcal{A}$        also removes the opposite edge!
                push!(tmpE, ($\nu$, x, $\mu$))
            end
        end
        for ($\nu$, x, $\mu$) in tmpE
            $\bar{\nu} = \Lambda[\nu]$ , $\bar{\mu} = \Lambda[\mu]$
            if hasedge($\mathcal{A}$, $\bar{\nu}$, x)
                $\varphi$ = trace($\mathcal{A}$, $\bar{\nu}$, x))
                if $\Lambda[\varphi] \neq \bar{\mu}$
                    $\lambda$ = union!($\Lambda$, $\bar{\mu}$, $\varphi$)
                    push!(deleted, $\lambda = \bar{\mu} ? \varphi : \bar{\mu}$)
                end
            else
                add ($\bar{\nu}$, x, $\bar{\mu}$) to $\mathcal{A}$        also add the opposite edge.
            end
        end
    end
    return $\mathcal{A}$
end
```

Proposition: Let $H = K(\mathcal{A})$ and let S, T be such that $\mathrm{trace}(\mathcal{A}, S, \alpha) = \sigma$; $\mathrm{trace}(\mathcal{A}, T, \tau) = \alpha$

Let $\mathcal{B} = \mathrm{Coincidence}(\mathcal{A}, \sigma, \tau)$ then $K(\mathcal{B}) = \mathrm{Grp}\langle H, [ST] \rangle$.

Proof: $M = \mathrm{Grp}\langle H, [ST] \rangle$; $\mathcal{A}_0$ - automaton before coincidence!. After the call $\mathcal{B}$ is $\cong$ to a quotient of $\mathcal{A}_0$.

Claim: $K(\mathcal{B})$ contains H

Proof: every word accepted by $\mathcal{A}_0$ is also accepted by $\mathcal{B}$: If $W \in X^*$, $W = x_1 \cdots x_t$ and $(\alpha, x_1, \sigma_1), \ldots (\sigma_{t-1}, x_t, \alpha)$ is a path in $\mathcal{A}_0 \Rightarrow ([\alpha], x_1, [\sigma_1]) \in E(\mathcal{B})$

⋮

Note:

$$\mathrm{trace}(\mathcal{B}, S, [\alpha]) = [\sigma] = [\tau]$$

$$\mathrm{trace}(\mathcal{B}, T, [\tau]) = [\alpha]$$

$$\Rightarrow \mathrm{trace}(\mathcal{B}, ST, [\alpha]) = [\alpha]$$

$$\Rightarrow [ST] \in K(\mathcal{B}) \Rightarrow M \subseteq K(\mathcal{B}).$$

We can write a map $g : \mathcal{A}_0 \to \mathcal{A}_s(M)$

$$g(\psi) = g(\text{trace}(\mathcal{A}_0, V, \alpha)) := M[V].$$

(if there are two V_1, V_2 then $[V_1 V_2^{-1}] \in H < M$).

we'll write $\varphi \equiv \psi \iff g(\varphi) = g(\psi)$.

($\equiv$ is an eq. relation).

note: $$g(\sigma) = M[S] = M[ST]^{-1}[S] = M[T^{-1}] = g(\tau),$$

so $\sigma \equiv \tau$.

By definition coincidence! creates the smallest eq. relation $\approx$ s.t. $\sigma \approx \tau$

$$\Rightarrow \approx \subset \equiv.$$

Therefore we can create a map h

$$\begin{tikzcd}
\mathcal{A}_0 \arrow[rr, "g"] \arrow[dr, "{[\,]_\approx}"'] & & \mathcal{A}_s(M) \\
& \mathcal{B} \arrow[ur, "h"'] &
\end{tikzcd}$$

by enlarging $\approx$ to $\equiv$ and everything that is traceable in $\mathcal{B}$ is traceable in $\mathcal{A}_s(M)$

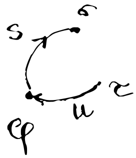
$$\Rightarrow K(\mathcal{B}) \subset K(\mathcal{A}_s(M)) = M.$$

$\square$

Two sided trace:

$W \in X^*$, $\mathcal{A}$ - coset automaton over X^*
(reduced)
φ - state of $\mathcal{A}$

Aim: modify $\mathcal{A}$ so that $\mathrm{trace}(\mathcal{A}, W, \varphi) = \varphi$.



Write W as $S \cdot T \cdot U$ and find
$\sigma \in \Sigma$ s.t. $\mathrm{trace}(\mathcal{A}, S, \varphi) = \sigma$
$\tau \in \Sigma$ s.t. $\mathrm{trace}(\mathcal{A}, U, \tau) = \varphi$

Such that S is as long as possible,
and then U is as long as possible.

Cases:

1) If $T = \varepsilon$ ⇒ identify σ and τ
(Call Coincidence)
2) If $T = x \in X$ ⇒ connect σ and τ via x
(Call to join)
3) If $|T| > 1$ ⇒ add new states following σ via $T[1]$ and predecessing τ via $T[\mathrm{end}]$.

Algorithm: trace_and_reverse!

Input: $\mathcal{A}$ – coset automaton

W – reduced word in X^*

φ [= initial($\mathcal{A}$)] – a state of $\mathcal{A}$.

Output: $\mathcal{A}$ – so that trace($\mathcal{A}$, W, φ) = φ

```
begin
    n, σ = trace(A, W, φ)
    // S = W[begin : n]
    k, τ = trace(A, W[n+1 : end], φ, reverse=true)
    // U = W[end-k+1 : end]
    / T = W[n+1 : end-k]
    while length(W) - (n+k) > 1  // length(T) > 1
        x = W[n+1]
        A = define!(A, σ, x)
        σ = trace(A, x, σ)
        n = n+1
        if length(W) - (n+k) > 1
            x = W[end-k]
            A = define!(A, τ, x^-1)
            k = k+1
        end
    end
    if length(W) - (n+k) = 1
        x = W[n+1]
        A = join!(A, σ, x, τ)
    elseif σ ≠ τ  // length(W) = n+k here i.e. T = ε
        A = coincidence!(A, σ, τ)
    end
    return A
end
```

a special routine for tracing words in reverse; efficient for coset aut.

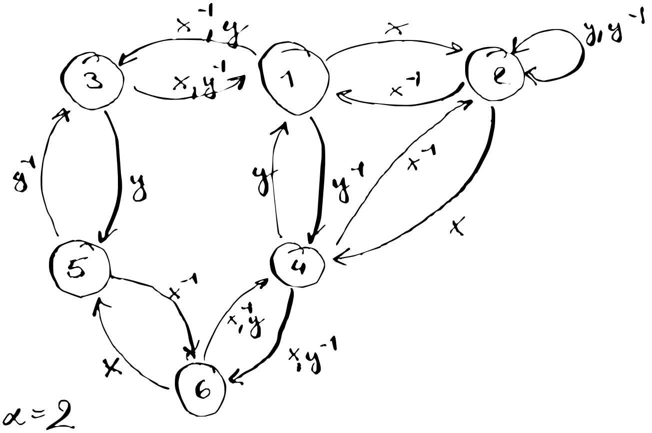
Proposition: Let $H = K(\mathcal{A})$; $\varphi \in \Sigma$

$L, M \in X^*$ s.t. $\alpha^L = \varphi$; $\varphi^M = \alpha$.

After trace_and_reverse($\mathcal{A}$, W, φ)

$$K(\mathcal{A}) = \operatorname{Grp}\langle H, [LWM] \rangle.$$

Ex:



$W = xy^{-2}x^{-1}y \quad , \alpha = 2$

begin:

$${}_2\, x \quad y^{-1} \quad y^{-1} \quad x^{-1} \quad y_2$$

forward:

$${}_2\, x \;{}_4\, y^{-1} \;{}_6\, y^{-1} \quad x^{-1} \quad y_2$$

reverse:

$${}_2\, x \;{}_4\, y^{-1} \;{}_6\, y^{-1} \;{}_4\, x^{-1} \;{}_2\, y_2$$

$$\underline{1}$$

we get $T = \varepsilon$ but $\sigma \neq \tau \Rightarrow$ call coincidence!$(\mathcal{A}, \sigma, \tau)$

$W = x^{-1}yxyxy^{3}$

$\alpha = 1$

// unsuccessful trace

begin:

$${}_1\, x^{-1} \quad y \quad x \quad y \quad x \quad y \quad y \quad y_1$$

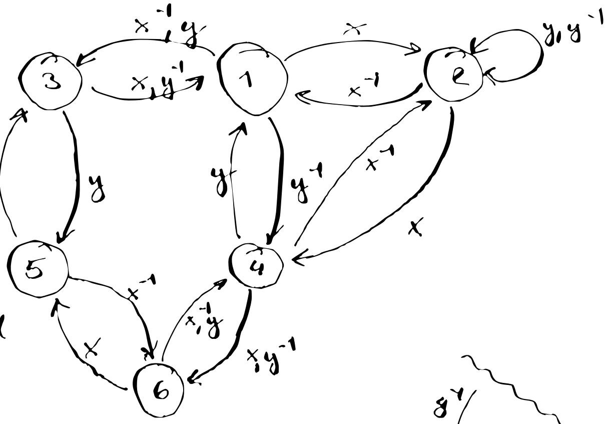
forward trace:

$${}_1\, x^{-1} \;{}_3\, y \;{}_5\, x \quad y \quad x \quad y \quad y \quad y_1$$

backward trace

$${}_1\, x^{-1} \;{}_3\, y \;{}_5\, x \quad y \quad x \quad y \;{}_6\, y \;{}_4\, y_1$$

$\mathcal{A} =$

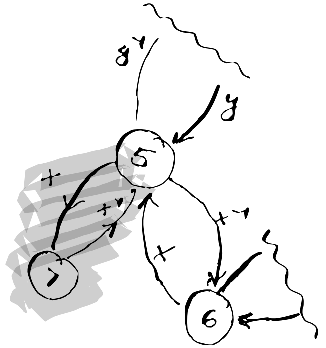


$W = y^2 x y^2 x^{-2} y^2$, $\alpha = 1$

traces:

$$1 \; ^{y} \; 3 \; ^{y} \; 5 \; ^{x} \quad ^{y} \, ^{y} \, ^{x^{-1}} \; 5 \; ^{x^{-1}} \; 6 \; ^{y} \; 4 \; ^{y} \; 1$$

Define! ($\mathcal{A}$, 5, x)

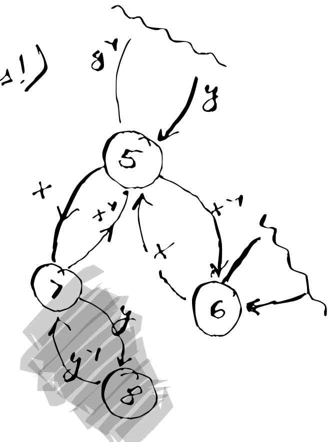


Continue:

$$1 \; ^{y} \; 3 \; ^{y} \; 5 \; ^{x} \; 7 \quad ^{y} \, ^{y} \; 7 \; ^{x^{-1}} \; 5 \; ^{x^{-1}} \; 6 \; ^{y} \; 4 \; ^{y} \; 1$$

(we continue extending trace from both sides!)

Define! ($\mathcal{A}$, 7, y)

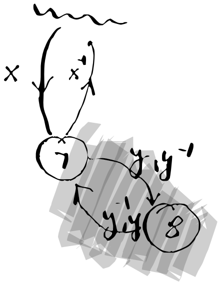


Continue:

$$1 \; ^{y} \; 3 \; ^{y} \; 5 \; ^{x} \; 7 \; ^{y} \; \underline{8 \; ^{y} \; 7} \; ^{x^{-1}} \; 5 \; ^{x^{-1}} \; 6 \; ^{y} \; 4 \; ^{y} \; 1$$

$T = \varepsilon$ but $8 \neq 7$

$\Rightarrow$ join($\mathcal{A}$, 8, y, 7)



Algorithm : coset_enumeration

Input: • X - alphabet with inverses

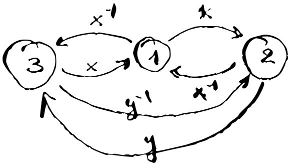
• $\mathcal{U}$ - a finite set of words over X

Output: $\mathcal{A}_I(H)$, where $H = Grp\langle \mathcal{U} \rangle$

```
begin
   $\mathcal{A} = (\{1\}, X, \{\}, \{1\}, \{1\})$
   for u in $\mathcal{U}$
      w = rewrite(u, $\mathcal{U}$)   // freely reduced
      trace_and_reverse!($\mathcal{A}$, w, 1)
   end
   return $\mathcal{A}$
end
```

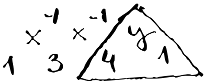
$X = \{x^{\pm 1}, y^{\pm 1}\}$ $\mathcal{U} = \{xyx, x^{-2}y, xy^2x^{-1}y\}$

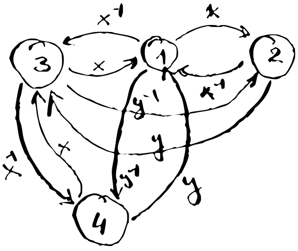
1)



// after tracing xyx

2)



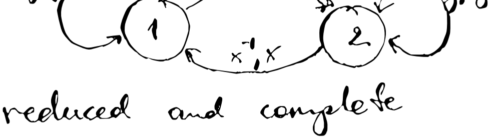


3) $1 \; x \; 2 \; y \; \binom{3}{2} \; y \; 3 \; x^{-1} \; 4 \; y \; 1$

$\Lambda_1 = \{1, 4\}$

$\Lambda_2 = \{2, 3\}$

$\mathcal{A}_I(\langle \mathcal{U} \rangle)$



reduced and complete

$\Rightarrow \langle \mathcal{U} \rangle$ is of finite index $= 2$ in $FGrp(X)$.

Different way of dealing with this problem:

$Y = \{H\} \cup \{X\}, \quad H < x < x^{-1} < y < y^{-1}$

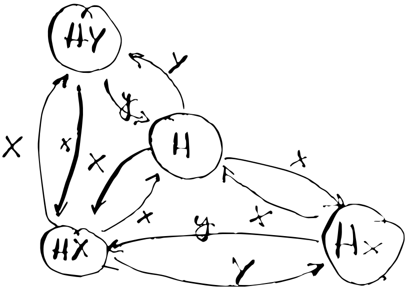
For Knuth-Bendix input:

$$R \left\{ \begin{array}{ll} x x^{-1} \to \varepsilon & x^{-1} x \to \varepsilon \\ y y^{-1} \to \varepsilon & y^{-1} y \to \varepsilon \end{array} \right.$$

$$H xyx \to H \qquad H x^{-2} y \to H$$

Output: $R \cup S$ s.t.

$$S = \left\{ \begin{array}{l} Hxy \to HX \\ HXX \to HY \\ HXY \to Hx \\ HYx \to HX \end{array} \right\}$$



General procedure:

Given $R = FGRel(X)$ choose letter $\ulcorner H \urcorner \notin X$.

Set $Y = \{\ulcorner H \urcorner\} \cup X$, + ordering on Y.

Let $\mathcal{U} \subset (X^{\pm})^*$; $T = \{(\ulcorner H \urcorner \cdot u, \ulcorner H \urcorner) \; u \in \mathcal{U}$

Perform Knuth-Bendix on $(R \cup T, \text{Lenlex}(Y))$.

obtaining $\mathcal{V}$, reduced, confluent rws.

$\mathcal{L} = \{B : (\ulcorner H \urcorner B \to \ulcorner H \urcorner C) \in \mathcal{V}\}$, reps for

$\mathcal{P}$ - proper prefixes of elts from $\mathcal{L}$. ↙ important cosets!

edges: for $P \in \mathcal{P}$ rewrite $\ulcorner H \urcorner P x$ w.r.t. $\mathcal{V}$

$\rightsquigarrow Q$

if $Q \in \mathcal{P} \Rightarrow (P, x\, Q) \in$ edges.