Computing Normalisers in Permutation Groups

Mun See Chang

University of St Andrews

Supervised by Dr Chris Jefferson and Dr Colva Roney-Dougal

11 August 2017

Digression: Base

Definition (Base points)

$G \leq Sym(\Omega)$. A **base** of G is $B = [\beta_1, \beta_2, \ldots, \beta_k] \in \Omega^k$ such that $G_{\beta_1,\beta_2,\ldots,\beta_k} = 1$.

Digression: Base

Definition (Base points)

$G \leq Sym(\Omega)$. A **base** of G is $B = [\beta_1, \beta_2, \ldots, \beta_k] \in \Omega^k$ such that $G_{\beta_1,\beta_2,\ldots,\beta_k} = 1$.

Example

$G = \langle (1,2,3),(4,5) \rangle$. $G_1 = \{(),(4,5)\}$. $G_{1,4} = \{()\}$. So $[1,4]$ is a base of G.

Digression: Base

Definition (Base points)

$G \leq Sym(\Omega)$. A **base** of G is $B = [\beta_1, \beta_2, \ldots, \beta_k] \in \Omega^k$ such that $G_{\beta_1, \beta_2, \ldots, \beta_k} = 1$.

Example

$G = \langle (1,2,3), (4,5) \rangle$. $G_1 = \{(), (4,5)\}$. $G_{1,4} = \{()\}$. So $[1,4]$ is a base of G.

Definition

$g \in G \leq Sym(\Omega)$.The **base image** of g relative to a base B is $B^g := [\beta_1^g, \beta_2^g, \ldots \beta_k^g]$.

Digression: Base

Definition (Base points)

$G \leq Sym(\Omega)$. A **base** of G is $B = [\beta_1, \beta_2, \ldots, \beta_k] \in \Omega^k$ such that $G_{\beta_1, \beta_2, \ldots, \beta_k} = 1$.

Example

$G = \langle (1,2,3), (4,5) \rangle$. $G_1 = \{(), (4,5)\}$. $G_{1,4} = \{()\}$. So $[1,4]$ is a base of G.

Definition

$g \in G \leq Sym(\Omega)$.The **base image** of g relative to a base B is $B^g := [\beta_1^g, \beta_2^g, \ldots \beta_k^g]$.

Lemma (Uniqueness of base image)

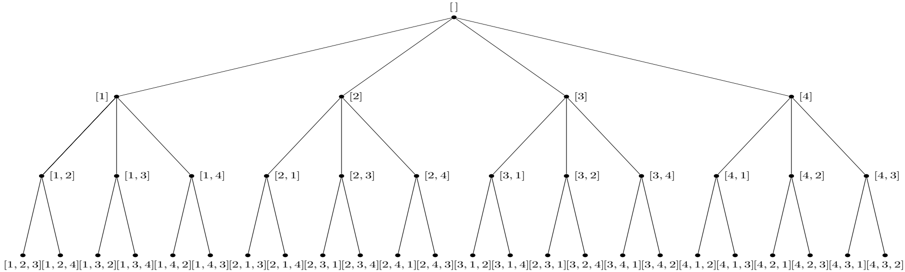
The base image B^g of g uniquely determines $g \in G$.

Finding Normalisers by Backtrack search

- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search

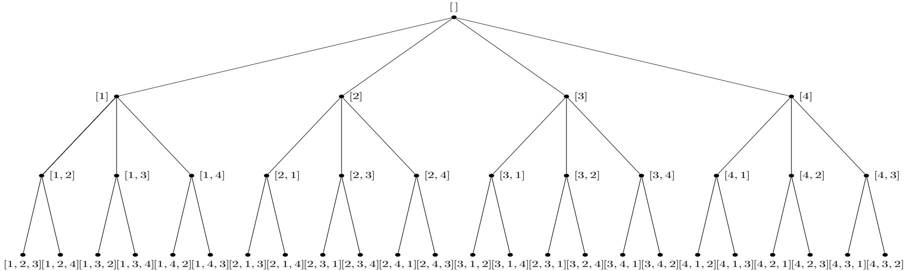
Finding Normalisers by Backtrack search

- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



Finding Normalisers by Backtrack search

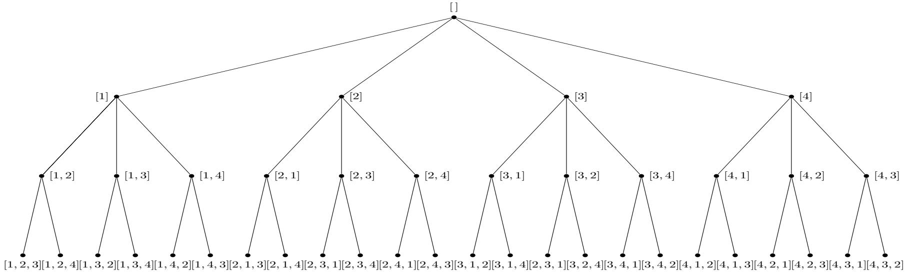
- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$

Finding Normalisers by Backtrack search

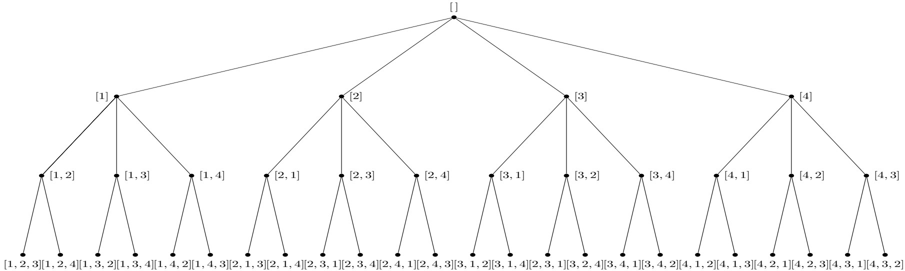
- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$
- At each node, ask: Could there be a solution under here?

Finding Normalisers by Backtrack search

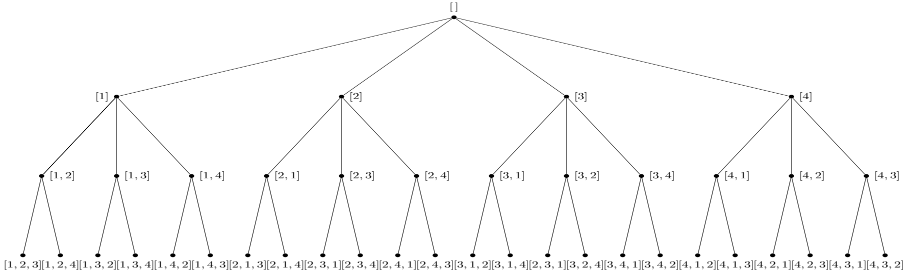
- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$
- At each node, ask: Could there be a solution under here?
- If not, backtrack;

Finding Normalisers by Backtrack search

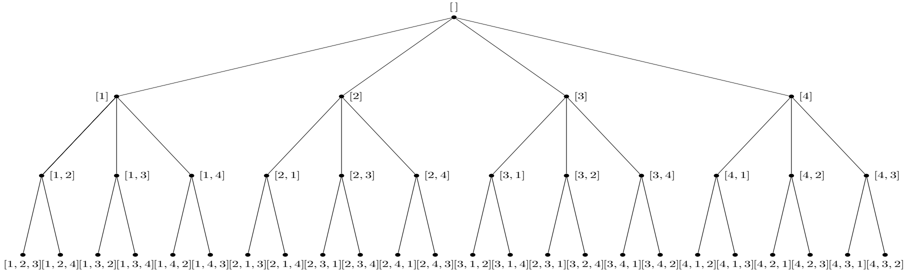
- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$
- At each node, ask: Could there be a solution under here?
- If not, backtrack; if yes, descend

Finding Normalisers by Backtrack search

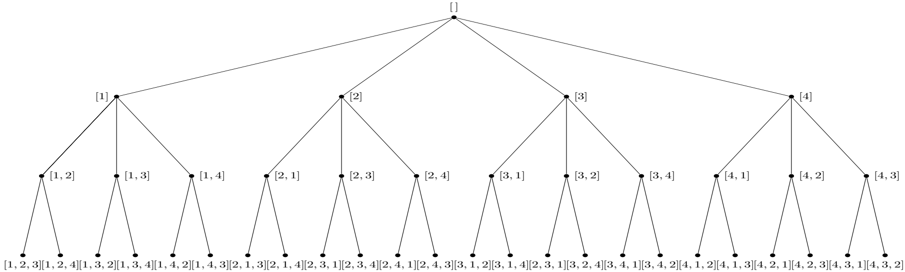
- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$
- At each node, ask: Could there be a solution under here?
- If not, backtrack; if yes, descend
- If find $g \in N_G(H)$, update $N = \langle N, g \rangle$

Finding Normalisers by Backtrack search

- $G, H \leq S_n$. No known polynomial time algorithm (in general) to compute $N_G(H)$: use backtrack search



- N will become $N_G(H)$. Initialise $N = H$
- At each node, ask: Could there be a solution under here?
- If not, backtrack; if yes, descend
- If find $g \in N_G(H)$, update $N = \langle N, g \rangle$
- Motto: fail early to avoid traversing bigger subtree

Current tests

[1] Gregory Butler. "Computing normalizers in permutation groups". In: *Journals of Algorithms* 4 (1983), pp. 163–175. DOI: https://doi.org/10.1016/0196-6774(83)90043-3.

[2] Heiko Theissen. "Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen". PhD thesis. RWTH Aachen, 1997. PhD thesis.

Current tests

Lemma

If $g \in N_G(H)$ *and* $[\beta_1, \beta_2, \ldots, \beta_i]^g = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ $(i \leq k)$, *then* $g^{-1} H_{\beta_1, \beta_2, \ldots, \beta_i} g = H_{\alpha_1, \alpha_2, \ldots, \alpha_i}$.

[1] Gregory Butler. "Computing normalizers in permutation groups". In: *Journals of Algorithms* 4 (1983), pp. 163–175. DOI: https://doi.org/10.1016/0196-6774(83)90043-3.

[2] Heiko Theissen. "Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen". PhD thesis. RWTH Aachen, 1997. PhD thesis.

Current tests

Lemma

If $g \in N_G(H)$ and $[\beta_1, \beta_2, \ldots, \beta_i]^g = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ $(i \leq k)$, then $g^{-1} H_{\beta_1,\beta_2,\ldots,\beta_i} g = H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.

- At each node $[\alpha_1, \alpha_2, \ldots, \alpha_i]$, compare $H_{\beta_1,\beta_2,\ldots\beta_i}$ and $H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.

[1] Gregory Butler. "Computing normalizers in permutation groups". In: *Journals of Algorithms* 4 (1983), pp. 163–175. DOI: https://doi.org/10.1016/0196-6774(83)90043-3.

[2] Heiko Theissen. "Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen". PhD thesis. RWTH Aachen, 1997. PhD thesis.

Current tests

Lemma

If $g \in N_G(H)$ *and* $[\beta_1, \beta_2, \ldots, \beta_i]^g = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ $(i \leq k)$, *then* $g^{-1} H_{\beta_1,\beta_2,\ldots,\beta_i} g = H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.

- At each node $[\alpha_1, \alpha_2, \ldots, \alpha_i]$, compare $H_{\beta_1,\beta_2,\ldots\beta_i}$ and $H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.
- $N_G(H)$ permutes H-orbits.[1]

[1] Gregory Butler. "Computing normalizers in permutation groups". In: *Journals of Algorithms* 4 (1983), pp. 163–175. DOI: https://doi.org/10.1016/0196-6774(83)90043-3.

[2] Heiko Theissen. "Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen". PhD thesis. RWTH Aachen, 1997. PhD thesis.

Current tests

Lemma

If $g \in N_G(H)$ *and* $[\beta_1, \beta_2, \ldots, \beta_i]^g = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ $(i \leq k)$*, then* $g^{-1} H_{\beta_1,\beta_2,\ldots,\beta_i} g = H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.

- At each node $[\alpha_1, \alpha_2, \ldots, \alpha_i]$, compare $H_{\beta_1,\beta_2,\ldots\beta_i}$ and $H_{\alpha_1,\alpha_2,\ldots,\alpha_i}$.
- $N_G(H)$ permutes H-orbits.[1]
- $N_G(H)$ permutes orbital graphs of H.[2]

[1] Gregory Butler. "Computing normalizers in permutation groups". In: *Journals of Algorithms* 4 (1983), pp. 163–175. DOI: https://doi.org/10.1016/0196-6774(83)90043-3.

[2] Heiko Theissen. "Eine Methode zur Normalisatorberechnung in Permutationsgruppen mit Anwendungen in der Konstruktion primitiver Gruppen". PhD thesis. RWTH Aachen, 1997. PhD thesis.

Our groups

Our groups

- V permutation isomorphic to $\langle (1,2), \ldots, (2m-1, 2m) \rangle \leq S_{2m}$, $(n = 2m)$

Our groups

- V permutation isomorphic to $\langle (1,2), \ldots, (2m-1, 2m) \rangle \leq S_{2m}$, $(n = 2m)$
- $H \leq V$, generated by $\sim m/2$ elements

Our groups

- V permutation isomorphic to $\langle (1,2),\ldots,(2m-1,2m)\rangle \leq S_{2m}$, $(n = 2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this

Our groups

- V permutation isomorphic to $\langle (1,2), \ldots, (2m-1, 2m) \rangle \leq S_{2m}$, $(n = 2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this
- Current tests don't do much

Our groups

- V permutation isomorphic to $\langle(1,2),\ldots,(2m-1,2m)\rangle \leq S_{2m}$, $(n=2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this
- Current tests don't do much

Example

$H = \langle(1,7)(2,8)(5,11),(1,7)(2,8)(3,9)(6,12),(2,8)(3,9)(4,10)\rangle \leq S_{12}$

Our groups

- V permutation isomorphic to $\langle (1,2),\ldots,(2m-1,2m)\rangle \leq S_{2m}$, $(n=2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this
- Current tests don't do much

Example

$H = \langle (1,7)(2,8)(5,11),(1,7)(2,8)(3,9)(6,12),(2,8)(3,9)(4,10)\rangle \leq S_{12}$

$$
\begin{array}{cc}
 & \begin{array}{cccccc} 1,7 & 2,8 & 3,9 & 4,10 & 5,11 & 6,12 \end{array} \\
M = & \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}
\end{array}
$$

Our groups

- V permutation isomorphic to $\langle (1,2),\dots,(2m-1,2m)\rangle \leq S_{2m}$, $(n=2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this
- Current tests don't do much

Example

$H = \langle (1,7)(2,8)(5,11), (1,7)(2,8)(3,9)(6,12), (2,8)(3,9)(4,10)\rangle \leq S_{12}$

$$M = \begin{array}{c} \begin{array}{cccccc} 1,7 & 2,8 & 3,9 & 4,10 & 5,11 & 6,12 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

$$reduced(M) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Our groups

- V permutation isomorphic to $\langle(1,2),\ldots,(2m-1,2m)\rangle \leq S_{2m}$, $(n = 2m)$
- $H \leq V$, generated by $\sim m/2$ elements
- It is conjectured that a random subgroup of S_n is 'close' to this
- Current tests don't do much

Example

$H = \langle(1,7)(2,8)(5,11),(1,7)(2,8)(3,9)(6,12),(2,8)(3,9)(4,10)\rangle \leq S_{12}$

$$M = \begin{array}{c} \begin{array}{cccccc} 1,7 & 2,8 & 3,9 & 4,10 & 5,11 & 6,12 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

$$reduced(M) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Both M and $reduced(M)$ represent H.

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits
- The image of a point determines the image of the H-orbit

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits
- The image of a point determines the image of the H-orbit

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1,7 & 2,8 & 3,9 & 4,10 & 5,11 & 6,12 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits
- The image of a point determines the image of the H-orbit

Example

$$\begin{array}{cc} & \begin{array}{cccccc} 1,7 & 2,8 & 3,9 & 4,10 & 5,11 & 6,12 \end{array} \\ M = & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

$K \cong S_6$ acts on the columns of M, base of $K = [1, 2, 3, 4, 5]$.

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits
- The image of a point determines the image of the H-orbit

Example

$$M = \begin{array}{c} \begin{array}{cccccc} \color{red}{1},7 & \color{red}{2},8 & \color{red}{3},9 & \color{red}{4},10 & \color{red}{5},11 & \color{red}{6},12 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

$K \cong S_6$ acts on the columns of M, base of $K = [1, 2, 3, 4, 5]$.
$\sigma \in K$, then $M' = M^\sigma$.

Normalisers in vector spaces

Lemma

Let M be a matrix over $GF(2)$ representing H and let $g \in S_n$. Let M' be the matrix representing H^g. Then $g \in N_{S_n}(H) \iff row(M) = row(M')$.

- Elements of $N_G(H)$ that fix each H-orbit: easy. Only search for those that permutes the orbits
- The image of a point determines the image of the H-orbit

Example

$$\begin{array}{cc} & \begin{array}{cccccc} \color{red}{1},7 & \color{red}{2},8 & \color{red}{3},9 & \color{red}{4},10 & \color{red}{5},11 & \color{red}{6},12 \end{array} \\ M = & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

$K \cong S_6$ acts on the columns of M, base of $K = [1, 2, 3, 4, 5]$.
$\sigma \in K$, then $M' = M^\sigma$.

- Centralisers are determined identical columns - assume none

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \multicolumn{6}{c}{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$M^\sigma = \begin{array}{ccc} 3 & 2 & 1 \\ \multicolumn{3}{c}{\begin{bmatrix} 0 & 0 & 1 & & & \\ 0 & 1 & 0 & & & \\ 1 & 0 & 0 & & & \end{bmatrix}} \end{array}$$

.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$M^\sigma = \begin{array}{c} \begin{matrix} 3 & 2 & 1 & & & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & & & \\ 1 & 0 & 0 & & & \end{bmatrix} \end{array}$$

.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1,2,3]$.
Base of $K = [1,2,3,4,5]$.

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3,2,1]$.

$$M^\sigma = \begin{array}{c} \begin{matrix} 3 & 2 & 1 & & & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & & & \end{bmatrix} \end{array}$$

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$\begin{array}{cc} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ M = & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$\begin{array}{cc} & \begin{array}{cccccc} 3 & 2 & 1 & & & \end{array} \\ M^\sigma = & \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$M^\sigma = \begin{array}{c} \begin{matrix} 3 & 2 & 1 & & & \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Columns of M^σ must be columns of M.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$M^\sigma = \begin{array}{c} \begin{matrix} 3 & 2 & 1 & 5 & 4 & 6 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Columns of M^σ must be columns of M. Extend to base image $[3, 2, 1, 5, 4]$.

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Partial base image $= [3, 2, 1]$.

$$M^\sigma = \begin{array}{c} \begin{array}{cccccc} 3 & 2 & 1 & 5 & 4 & 6 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Columns of M^σ must be columns of M. Extend to base image $[3, 2, 1, 5, 4]$.

- If pass, we can extend the partial base image to full base image, add the corresponding group element to N; else: backtrack

Limiting the depth of the search tree

- If H-base images under g is known, then s^g is known $\forall$generator s of H.

Example (Extending from base H)

Base of $H = [1, 2, 3]$.
Base of $K = [1, 2, 3, 4, 5]$.

$$M = \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$
$$M = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Partial base image $= [3, 2, 1]$.

$$\begin{array}{cccccc} 3 & 2 & 1 & 5 & 4 & 6 \end{array}$$
$$M^\sigma = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Columns of M^σ must be columns of M. Extend to base image $[3, 2, 1, 5, 4]$.

- If pass, we can extend the partial base image to full base image, add the corresponding group element to N; else: backtrack
- So only check up to depth length of base of H - which is $n/4$

Tests

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.Try partial base image $:= [1, 3]$.

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.Try partial base image $:= [1, 3]$.

$$M(H_{1,2}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.Try partial base image := $[1, 3]$.

$$M(H_{1,2}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad M(H_{1,3}) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.Try partial base image $:= [1, 3]$.

$$M(H_{1,2}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad M(H_{1,3}) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Matrix has three 1's.

Tests

Lemma

If $\sigma \in N_K(H)$ *s.t.* $[\beta_1, \beta_2, \ldots, \beta_i]^\sigma = [\alpha_1, \alpha_2, \ldots, \alpha_i]$ *then* $row(M(H_{\beta_1,\beta_2,\ldots,\beta_i})^\sigma) = row(M(H_{\alpha_1,\alpha_2,\ldots,\alpha_i}))$

Example

$$M = \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Base of $H = [1, 2, 3]$.Try partial base image $:= [1, 3]$.

$$M(H_{1,2}) = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \qquad M(H_{1,3}) = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Matrix has three 1's. Matrix has four 1's $\implies$ Backtrack!

Tests

Lemma

$\sigma \in K$.If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Tests

Lemma

$\sigma \in K$.If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\multicolumn{6}{c}{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}} \\
c_1 & c_2 & c_3 & c_4 & c_5 & c_6
\end{array}$$

$$\begin{aligned}
c_1 + c_2 + c_4 &= [0, 0, 0] \\
c_2 + c_3 + c_5 &= [0, 0, 0] \\
c_1 + c_2 + c_3 + c_6 &= [0, 0, 0]
\end{aligned}$$

Tests

Lemma

$\sigma \in K$. If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0,0,0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0,0,0] \end{aligned}$$

Tests

Lemma

$\sigma \in K$. If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \multicolumn{6}{c}{\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}} \\ c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

Base of $H = [1, 2, 3]$
Try partial base image $:= [1, 3]$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0, 0, 0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0, 0, 0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0, 0, 0] \end{aligned}$$

Tests

Lemma

$\sigma \in K$. If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$
$$\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0,0,0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0,0,0] \end{aligned}$$

Base of $H = [1,2,3]$
Try partial base image $:= [1,3]$

$$c_1^\sigma + c_2^\sigma + c_4^\sigma = [0,0,0]$$

Tests

Lemma

$\sigma \in K$. If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0,0,0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0,0,0] \end{aligned}$$

Base of $H = [1, 2, 3]$
Try partial base image $:= [1, 3]$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_1 + c_3 + c_4^\sigma &= [0,0,0] \end{aligned}$$

Tests

Lemma

$\sigma \in K$.If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$
$$\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0,0,0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0,0,0] \end{aligned}$$

Base of $H = [1,2,3]$
Try partial base image $:= [1,3]$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_1 + c_3 + c_4^\sigma &= [0,0,0] \\ c_4^\sigma &= [1,0,1] \end{aligned}$$

Tests

Lemma

$\sigma \in K$. If L is a set of linearly dependent columns of M and $\sigma \in N_K(H)$, then L^σ is also a set of linearly dependent columns of M^σ.

Example

$$\begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$\begin{array}{cccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 \end{array}$$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_2^\sigma + c_3^\sigma + c_5^\sigma &= [0,0,0] \\ c_1^\sigma + c_2^\sigma + c_3^\sigma + c_6^\sigma &= [0,0,0] \end{aligned}$$

Base of $H = [1,2,3]$
Try partial base image $:= [1,3]$

$$\begin{aligned} c_1^\sigma + c_2^\sigma + c_4^\sigma &= [0,0,0] \\ c_1 + c_3 + c_4^\sigma &= [0,0,0] \\ c_4^\sigma &= [1,0,1] \end{aligned}$$

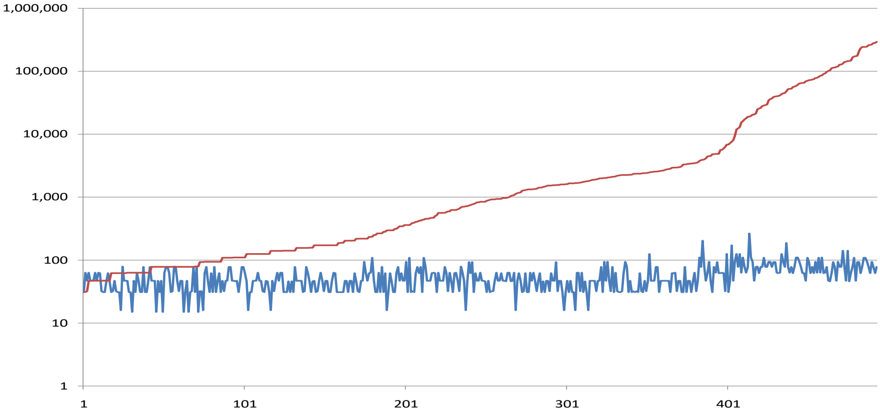
This is not a column in M,
$\implies$ Backtrack!

Test results

Tested on 500 random groups on 20 points in GAP

Test results

Tested on 500 random groups on 20 points in GAP



- Red: Log(time taken by the original algorithm of GAP, in milliseconds);
- Blue: Log(time taken by new algorithm)