Automorphism Groups of Partially Ordered Sets

Ricardo Garcia, Mehek Mehra, Joelle Ocheltree

Pomona Research in Mathematical Experience

August 2, 2022

Table of Contents

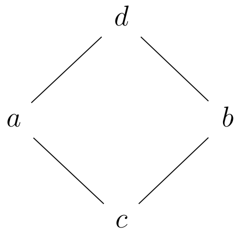
Preliminaries

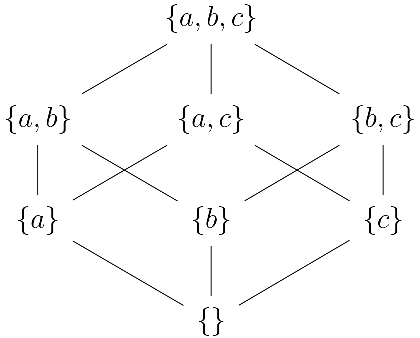
Poset

Pomona College

Definition

A ***poset*** $(P, \leq_P)$ *is a set* P *with a relation* $\leq_P$ *such that for all* $x, y, z \in P$*, we have* $x \leq_P x$; $x \leq_P y$ *and* $y \leq_P x \implies y = x$; *and* $x \leq_P y$ *and* $y \leq_P z \implies x \leq_P z$.





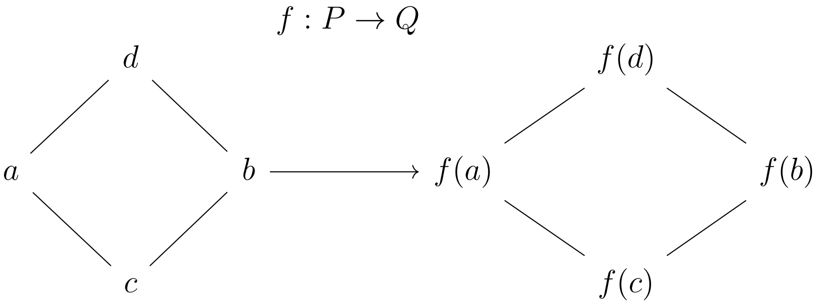
Poset examples

Poset Map

Pomona College

Definition

If P and Q are posets, then $f : P \to Q$ is a ***poset map*** *if for all $x, y \in P$, $x \leq_P y \implies f(x) \leq_Q f(y)$.*



Poset Isomorphism

Definition

A poset map $f : P \to Q$ is said to be ***order-reflexive*** *if for all $x, y \in P$, $f(x) \leq_Q f(y) \implies x \leq_P y$.*

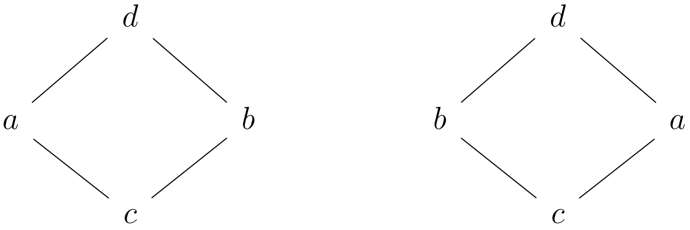
Definition

A surjective, order-reflexive map $f : P \to Q$ is called a ***poset isomorphism***.

Automorphism

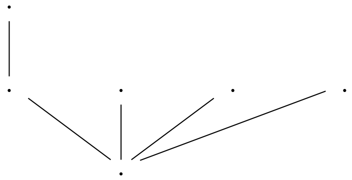
Definition

An isomorphism $f : P \to P$ is called an ***automorphism****. The set of all automorphisms is denoted as $\operatorname{Aut} P$ and is a group under composition. For any poset P with p points, $\operatorname{Aut} P \leq p!$*

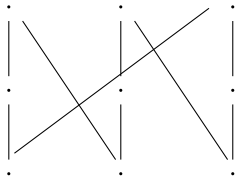


$$\operatorname{Aut} P = Z_2$$

Automorphism Examples



$\mathrm{Aut}\, Q = S_3$



$\mathrm{Aut}\, R = Z_3$
(Barmak '09)

Barmak and Minian 2009

Pomona College

Theorem (Barmak, Minian '09)

If G is a finite group of order n with r elements, then there exists a poset P with $n(r+2)$ points such that $\mathrm{Aut}\, P \cong G$.

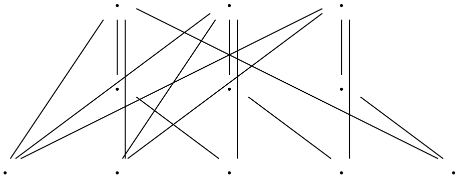
Theorem (Barmak '20)

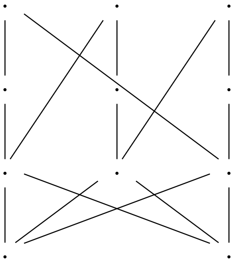
If G is a finite group, then there exists a poset P of $4|G|$ points such that $\mathrm{Aut}\, P \cong G$.

History

Pomona College

- 1939 - Robert Frucht proves that if G is a finite group of order n with r generators, then there exists a poset P with $n(r+2)$ points such that $\operatorname{Aut} P \cong G$. His paper was translated into English in 1949.
- 1946 - George Birkhoff proves that any finite group G is the automorphism group of some poset P.
- 1972 - M.C. Thornton proves that if G is a finite group of order n with r generators, then there exists a poset P with $n(2r+1)$ points such that $\operatorname{Aut} P \cong G$.
- 2009 - Barmak and Minian unknowingly reprove Frucht's theorem
- 2020 - Barmak proves that the poset P has $4|G|$ points





Beta Values of Groups

Definition

$\boldsymbol{\beta(G)} = \min\{|P| : P$ *is a poset with* $\operatorname{Aut} P \cong G\}$.

Beta Values of Groups

Pomona College

Definition

$\boldsymbol{\beta(G)} = \min\{|P| : P$ *is a poset with* $\operatorname{Aut} P \cong G\}$.

- $p = 2$ $\quad 2^{k+1} \leq \beta(Z_{2^k}) \leq 2^{k+1} + 12$ if $k \geq 2$
- $p = 3, 5$ $\quad 2p^k \leq \beta(Z_{p^k}) \leq 2p^k + 3p$
- $p \geq 7$ $\quad 2p^k \leq \beta(Z_{p^k}) \leq 2p^k + p$

Realizing Finite Groups G with a Minimal Poset P

$\beta(Z_3)$

Theorem (Barmak, '20)

$\beta(Z_3) = 9$

Definition

*An **orbit** of an element x is all possible destinations of x under group action.*

$\beta(Z_3)$

Theorem (Barmak, '20)

$\beta(Z_3) = 9$

Definition

*An **orbit** of an element x is all possible destinations of x under group action.*

2 orbits of 3 $\implies$ $\mathrm{Aut}\, P \neq Z_3$

3 orbits of 3 $\implies$ $\mathrm{Aut}\, P = Z_3$

$\beta(S_n)$

Proposition (C., G., M., O., S., '22)

$\beta(S_n) = n$ for all $n \in \mathbb{N}$

$\beta(S_n)$

Proposition (C., G., M., O., S., '22)

$\beta(S_n) = n$ for all $n \in \mathbb{N}$

- If $\operatorname{Aut} P = S_n$, $n \geq 1$ and $|P| = k$

$\beta(S_n)$

Pomona College

Proposition (C., G., M., O., S., '22)

$\beta(S_n) = n$ for all $n \in \mathbb{N}$

- If $\mathrm{Aut}\, P = S_n$, $n \geq 1$ and $|P| = k$
- $\mathrm{Aut}\, P \leq S_K$

$\beta(S_n)$

Proposition (C., G., M., O., S., '22)

$\beta(S_n) = n$ for all $n \in \mathbb{N}$

- If $\mathrm{Aut}\, P = S_n$, $n \geq 1$ and $|P| = k$
- $\mathrm{Aut}\, P \leq S_K$
- $S_n = \mathrm{Aut}\, P \leq S_K \implies n! \leq k! \implies n \leq k$

$\beta(S_n)$

Pomona College

Proposition (C., G., M., O., S., '22)

$\beta(S_n) = n$ for all $n \in \mathbb{N}$

- If $\mathrm{Aut}\, P = S_n$, $n \geq 1$ and $|P| = k$
- $\mathrm{Aut}\, P \leq S_K$
- $S_n = \mathrm{Aut}\, P \leq S_K \implies n! \leq k! \implies n \leq k$
- $\mathrm{Aut}\, P \cong S_n$

$\beta(S_n)$

Proposition (C., G., M., O., S., '22)

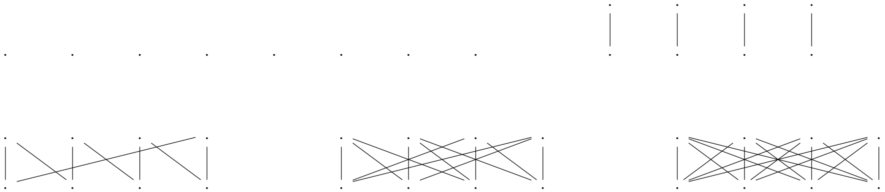
$\beta(S_n) = n$ for all $n \in \mathbb{N}$

- If $\mathrm{Aut}\, P = S_n$, $n \geq 1$ and $|P| = k$
- $\mathrm{Aut}\, P \leq S_K$
- $S_n = \mathrm{Aut}\, P \leq S_K \implies n! \leq k! \implies n \leq k$
- $\mathrm{Aut}\, P \cong S_n$
- $\beta(S_n) = n$ for all $n \in \mathbb{N}$

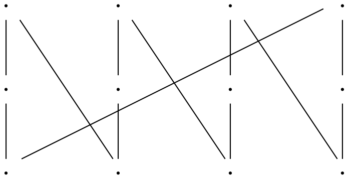
Conjectures

Conjecture

There is a nice proof of $\beta(Z_4) = 12$



$\beta(Z_4) = 12$



Conjectures

Conjecture

$\beta(Z_p) = 3p$ for any prime p

Conjecture

$\beta(Z_{p^k}) = 2p^k + p$ for all primes $p \geq 7$

Realizing Finitely Generated Abelian Groups

Realizing Infinite Groups

Previous work has realized all finite groups (Barmak '09)

Motivating Question

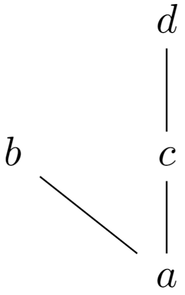
Can we realize finitely generated abelian groups?

The simplest relevant example is $\mathbb{Z}$

Covers

Definition

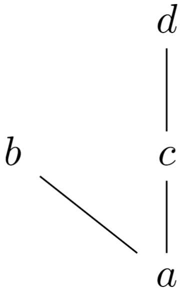
Let $x, y \in P$. We say that y covers x if $x \leq_P y$ and there is no $z \in P$ such that $x <_P z <_P y$.



Covers

Definition

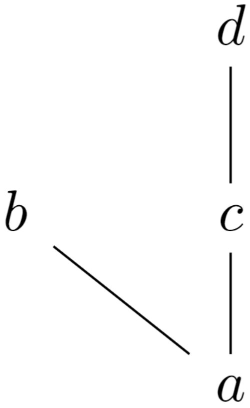
Let $x, y \in P$. We say that y covers x if $x \leq_P y$ and there is no $z \in P$ such that $x <_P z <_P y$.



b and c cover a, but d does not cover a

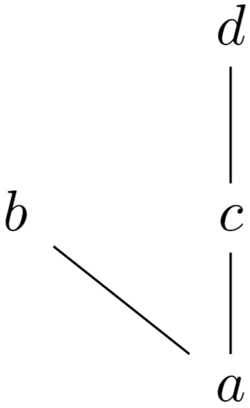
Covers

Covers are preserved under automorphisms



Covers

Covers are preserved under automorphisms

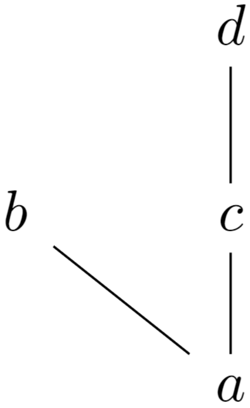


Proposition

Let $f \in \mathrm{Aut}\, P$. If b covers a then $f(b)$ covers $f(a)$

Covers

Covers are preserved under automorphisms



Proposition

Let $f \in \mathrm{Aut}\, P$. If b covers a then $f(b)$ covers $f(a)$

Proof Sketch: Since f is order reflexive, $f(a) < z < f(b)$ implies $a < f^{-1}(z) < b$

Realizing the Integers

$$\begin{array}{c} \vdots \\ \uparrow \\ 1 \\ \uparrow \\ 0 \\ \uparrow \\ -1 \\ \uparrow \\ \vdots \end{array}$$

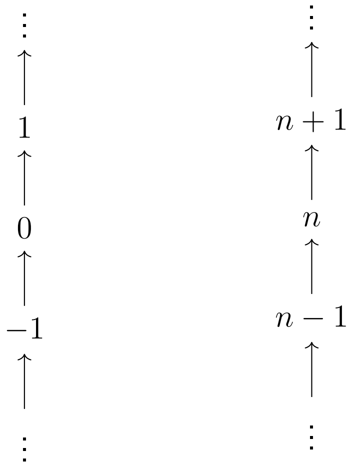
Realizing the Integers

$$\vdots$$
$$\uparrow$$
$$1$$
$$\uparrow$$
$$0$$
$$\uparrow$$
$$-1$$
$$\uparrow$$
$$\vdots$$

Result

Let $P = (\mathbb{Z}, \leq)$ with the usual order $\leq$. Then $\operatorname{Aut} P \cong \mathbb{Z}$

Realizing the Integers



Result

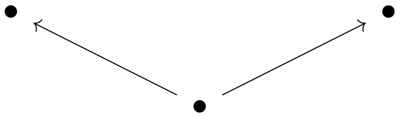
Let $P = (\mathbb{Z}, \leq)$ with the usual order $\leq$. Then $\operatorname{Aut} P \cong \mathbb{Z}$

The only automorphisms of P are $f(x) = x + n$ for some $n \in \mathbb{Z}$

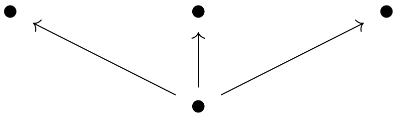
Going Further

Motivating Question

Having realized G and H, how do we realize $G \times H$?



Aut $P \cong \mathbb{Z}_2$



Aut $Q \cong S_3$

Examples

Aut $P \cong \mathbb{Z}_2$

Aut $Q \cong \mathbb{Z}_2$

Examples

Aut $P \cong \mathbb{Z}_2$

Aut $Q \cong \mathbb{Z}_2$

What is $\mathrm{Aut}\, P \sqcup Q$?

Examples

Aut $P \cong \mathbb{Z}_2$

Aut $Q \cong \mathbb{Z}_2$

$$\mathrm{Aut}\, P \sqcup Q \cong S_4 \ncong \mathbb{Z}_2 \times \mathbb{Z}_2$$

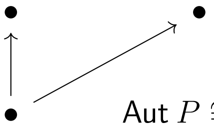
Connected Posets

Pomona College

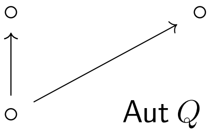
Definition

A poset P is connected if for any $x, y \in P$, there exists a sequence of points $(x = \gamma_1, \gamma_2, ..., \gamma_n = y)$ in P such that γ_i is comparable to γ_{i+1} for all $1 \leq i < n$.

Examples

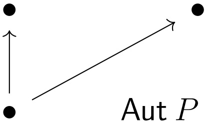


$\text{Aut } P \cong \mathbb{Z}_2$

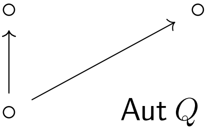


$\text{Aut } Q \cong \mathbb{Z}_2$

Examples



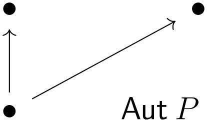
Aut $P \cong \mathbb{Z}_2$



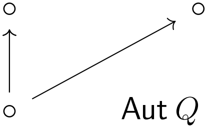
Aut $Q \cong \mathbb{Z}_2$

What is $\mathrm{Aut}\, P \sqcup Q$?

Examples



Aut $P \cong \mathbb{Z}_2$



Aut $Q \cong \mathbb{Z}_2$

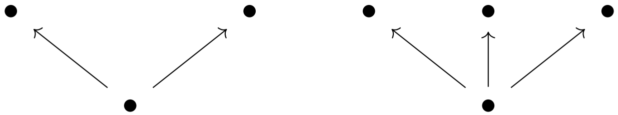
$$\text{Aut } P \sqcup Q \ncong \mathbb{Z}_2 \times \mathbb{Z}_2$$

A Useful Result

Theorem (C., G., M., O., S., '22)

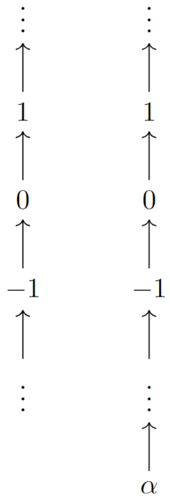
Let $\mathcal{P} = \sqcup P_i$ be a disjoint union of nonempty posets, where each P_i is connected, and they are pairwise not isomorphic. Then,

$$\operatorname{Aut}\mathcal{P} \cong \prod \operatorname{Aut} P_i$$



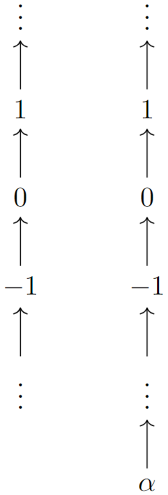
$$\text{Aut} P \sqcup Q \cong \mathbb{Z}_2 \times S_3$$

Realizing $\mathbb{Z}^r$



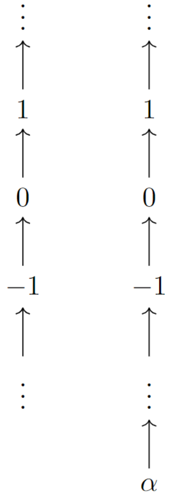
Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.

Realizing $\mathbb{Z}^r$



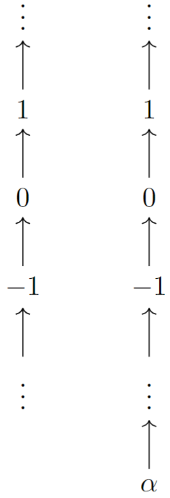
Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.
We know that $\operatorname{Aut} P_0 \cong \operatorname{Aut} P_1 \cong \mathbb{Z}$.

Realizing $\mathbb{Z}^r$



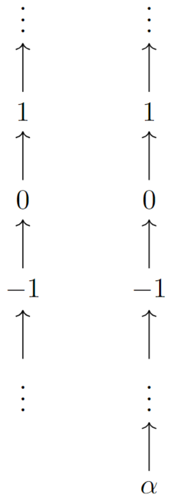
Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.
We know that $\mathrm{Aut}\, P_0 \cong \mathrm{Aut}\, P_1 \cong \mathbb{Z}$.
They are not isomorphic, $P_0 \ncong P_1$

Realizing $\mathbb{Z}^r$



Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.
We know that $\operatorname{Aut} P_0 \cong \operatorname{Aut} P_1 \cong \mathbb{Z}$.
They are not isomorphic, $P_0 \ncong P_1$
Therefore, $\operatorname{Aut} P \sqcup Q \cong \mathbb{Z} \times \mathbb{Z} \cong \mathbb{Z}^2$

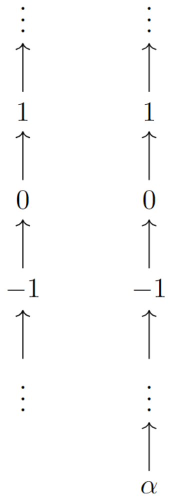
Realizing $\mathbb{Z}^r$



Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.
We know that $\operatorname{Aut} P_0 \cong \operatorname{Aut} P_1 \cong \mathbb{Z}$.
They are not isomorphic, $P_0 \ncong P_1$
Therefore, $\operatorname{Aut} P \sqcup Q \cong \mathbb{Z} \times \mathbb{Z} \cong \mathbb{Z}^2$
We can extend this result to

$$\operatorname{Aut} \bigsqcup_{0<i<r} P_i \cong \mathbb{Z}^r$$

Realizing $\mathbb{Z}^r$



Let $P_0 = (\mathbb{Z}, \leq)$ and $P_1 = P_0 \cup \{\alpha\}$.
We know that $\mathrm{Aut}\, P_0 \cong \mathrm{Aut}\, P_1 \cong \mathbb{Z}$.
They are not isomorphic, $P_0 \ncong P_1$
Therefore, $\mathrm{Aut}\, P \sqcup Q \cong \mathbb{Z} \times \mathbb{Z} \cong \mathbb{Z}^2$
We can extend this result to

$$\mathrm{Aut} \bigsqcup_{0<i<r} P_i \cong \mathbb{Z}^r$$

Result

$\mathbb{Z}^r$ is realizable for all $r \in \mathbb{N}$.

Realizing All F.G.A.G.

Theorem

Every finitely generated abelian group G can be decomposed as

$$G \cong \mathbb{Z}^r \times \mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2} \times \ldots \times \mathbb{Z}_{n_k}$$

Main Result

- We know all F.G.A.G are isomorphic to $\mathbb{Z}^r \times T$.

Main Result

- We know all F.G.A.G are isomorphic to $\mathbb{Z}^r \times T$.
- Given G and H, we know how to realize $G \times H$

Main Result

- We know all F.G.A.G are isomorphic to $\mathbb{Z}^r \times T$.
- Given G and H, we know how to realize $G \times H$
- (Barmak '09) gives us a construction for any finite group. So T is realizable

Main Result

- We know all F.G.A.G are isomorphic to $\mathbb{Z}^r \times T$.
- Given G and H, we know how to realize $G \times H$
- (Barmak '09) gives us a construction for any finite group. So T is realizable
- $\mathbb{Z}^r$ is realizable

Main Result

- We know all F.G.A.G are isomorphic to $\mathbb{Z}^r \times T$.
- Given G and H, we know how to realize $G \times H$
- (Barmak '09) gives us a construction for any finite group. So T is realizable
- $\mathbb{Z}^r$ is realizable

Theorem (C., G., M., O., S., '22)

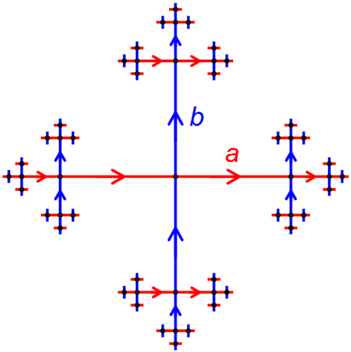
Every finitely generated abelian group is realizable as the automorphism group of some poset

Free Groups

Definition

The free group F_S over the set S is the set of reduced words that can be built from elements of S

- Let $S = \{a, b\}$. Then $ab^2a \in S$
- We have inverses: $a^{-1}, b^{-1} \in S$
- $bbaa^{-1} = b^2$

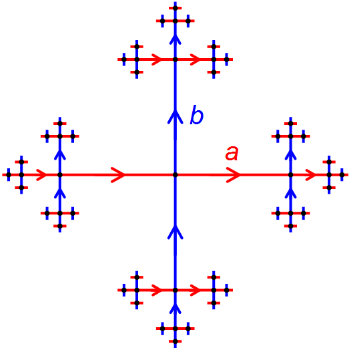


A Partial Order On Free Groups

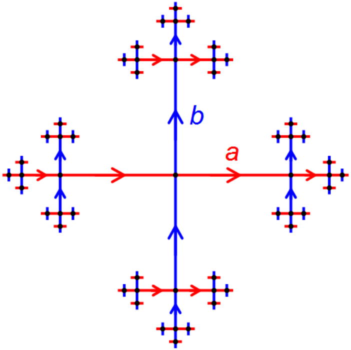
Theorem (C., G., M., O., S., '22)

Let $x, y \in F_S$. The relation $x \leq y \iff y = xg_1g_2...g_k$ where $g_i \in S \cup \{1_{F_S}\}$ makes F_S into a poset.

- $a \leq a$ since $1_{F_S} \in S \cup \{1_{F_S}\}$
- $a \leq ab^2$ since $b \in S \cup \{1_{F_S}\}$
- $a^{-7} \leq a^{-7}a^7 = 1_{F_S}$
- b is not comparable to a



Automorphism Group Of Free Group



Consider the map $f_p(x) = px$ for some $p \in F_S$. We proved that this is a poset automorphism. Take f_a for example:

- $b^{-1} \leq a \iff ab^{-1} \leq a^2$

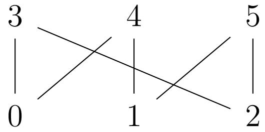
Result

There exists a poset P such that $\operatorname{Aut} P$ contains an isomorphic copy of F_S for any free group F_S

Counting Automorphisms with Python

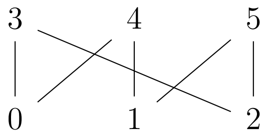
Counting Automorphisms

How many automorphisms does this poset have?



Counting Automorphisms

How many automorphisms does this poset have?



We can use Python to find out

Counting Automorphisms

How does the code work?

Counting Automorphisms

How does the code work?

1. Finds S_P for a poset P

Counting Automorphisms

How does the code work?

1. Finds S_P for a poset P
2. Finds and counts all of the bijections that preserve the structure of the poset

How Do We Count Automorphisms?

Why do we find S_P?

How Do We Count Automorphisms?

Why do we find S_P?

Lemma

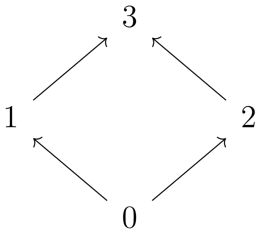
If P is a poset, then Aut P is a subgroup of S_P.

How Do We Count Automorphisms?

Why do we find S_P?

Lemma

If P is a poset, then Aut P is a subgroup of S_P.

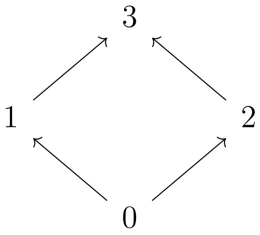


How Do We Count Automorphisms?

Why do we find S_P?

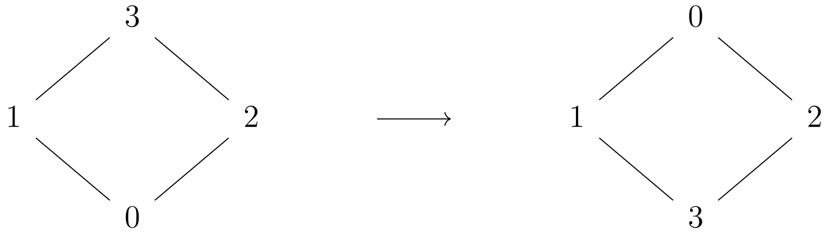
Lemma

If P is a poset, then Aut P is a subgroup of S_P.



Aut$P$ is a subgroup of S_4

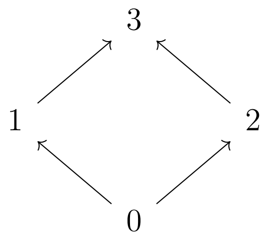
How Do We Count Automorphisms?



Getting Bijections from S_P

Let $P := \{0, 1, 2, 3\}$.
We focus on S_4.
e.g. $(0\ 3) \in S_P$
This is the same as the mapping $0 \mapsto 3$ and $3 \mapsto 0$.
This is not an automorphism

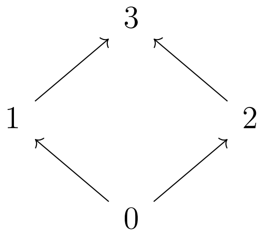
How Do We Count Automorphisms?



Checking to See Which Bijections are Automorphisms

We use matrices.
We construct $C = [c_{ij}]$ where $i, j \in P$. We set $c_{ij} = 1$ if $i = j$ or if j covers i. Otherwise, $c_{ij} = 0$.

How Do We Count Automorphisms?



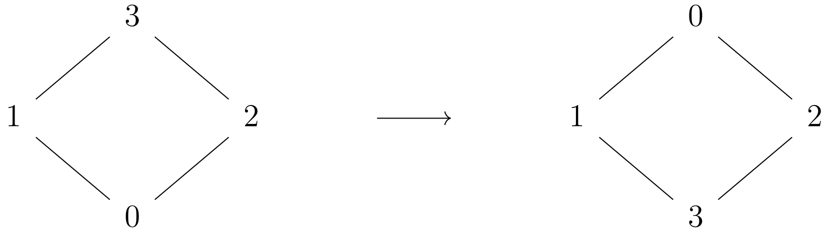
$$
\begin{array}{cccc}
0 & 1 & 2 & 3
\end{array}
$$

$$
\begin{pmatrix}
1 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{array}{c}
0 \\ 1 \\ 2 \\ 3
\end{array}
$$

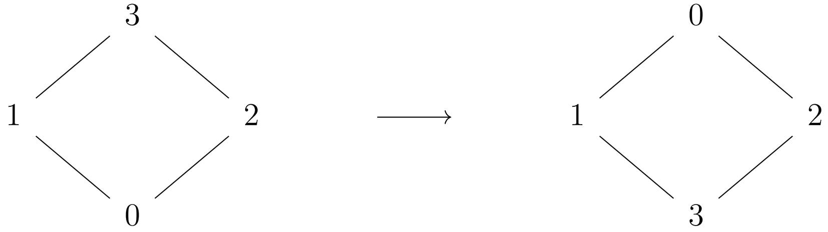
How Do We Count Automorphisms?

Applying a Bijection to the Matrix

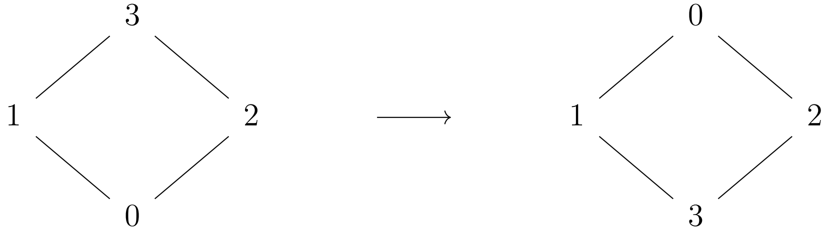
Based on the bijection mappings, we swap the rows and columns of the matrix.
Consider the bijection from earlier
$0 \mapsto 3$ and $3 \mapsto 0$



How do We Count Automorphisms?



How do We Count Automorphisms?



$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\!\!\begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \rightarrow \begin{array}{c} \\ 3 \\ 1 \\ 2 \\ 0 \end{array}\!\!\begin{array}{c} \begin{matrix} 3 & 1 & 2 & 0 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \end{array}$$

How Do We Count Automorphisms?

When is a Bijection an Automorphism?

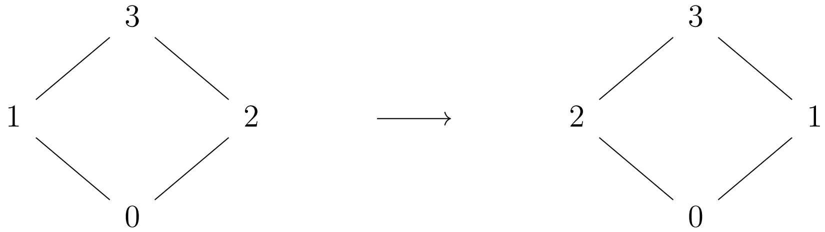
A bijection is an automorphism when the bijection does not change the matrix.

How Do We Count Automorphisms?

When is a Bijection an Automorphism?

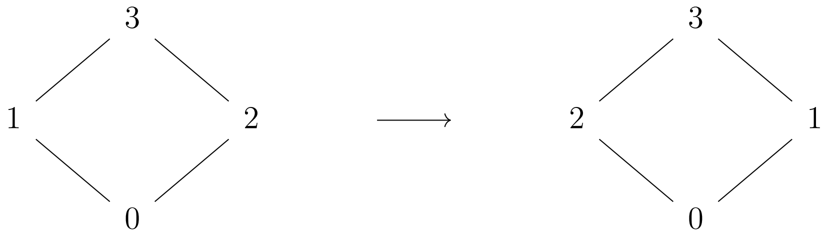
A bijection is an automorphism when the bijection does not change the matrix.

Example:



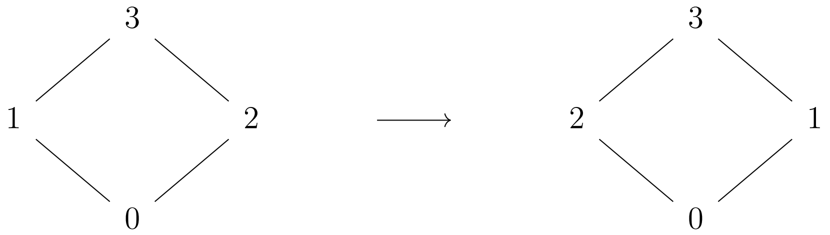
How Do We Count Automorphisms?

Example:



How Do We Count Automorphisms?

Example:



$$
\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}
\begin{array}{c} \begin{matrix} 0 & 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}
\to
\begin{array}{c} \\ 0 \\ 2 \\ 1 \\ 3 \end{array}
\begin{array}{c} \begin{matrix} 0 & 2 & 1 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}
$$

How Do We Count Automorphisms?

Putting it All Together

The program finds $\mathrm{Aut}\, P$ by

How Do We Count Automorphisms?

Putting it All Together

The program finds $\mathrm{Aut}\, P$ by

1. Finding S_P

How Do We Count Automorphisms?

Pomona College

Putting it All Together

The program finds $\mathrm{Aut}\, P$ by

1. Finding S_P
2. Constructing a matrix representation of P

How Do We Count Automorphisms?

Putting it All Together

The program finds $\operatorname{Aut} P$ by

1. Finding S_P
2. Constructing a matrix representation of P
3. For each element of S_P, swapping the rows and columns of the matrix and comparing it to the original matrix

How Do We Count Automorphisms?

Putting it All Together

The program finds $\mathrm{Aut}\, P$ by

1. Finding S_P
2. Constructing a matrix representation of P
3. For each element of S_P, swapping the rows and columns of the matrix and comparing it to the original matrix
4. Returning all of the bijections that do not change the matrix and counting the number of such bijections

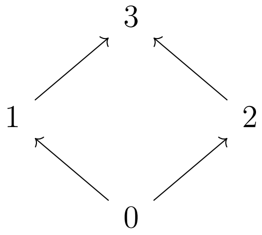
How Do We Count Automorphisms?

Putting it All Together

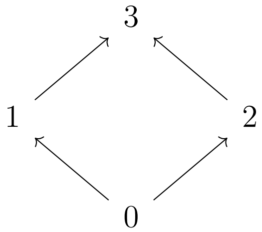
The program finds $\operatorname{Aut} P$ by

1. Finding S_P
2. Constructing a matrix representation of P
3. For each element of S_P, swapping the rows and columns of the matrix and comparing it to the original matrix
4. Returning all of the bijections that do not change the matrix and counting the number of such bijections

Running the Code

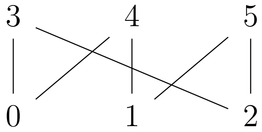


Running the Code

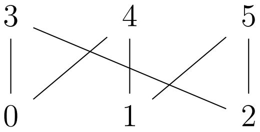



```
This poset has 2 automorphisms

The automorphisms on this poset are :

f0:

0 -> 0
1 -> 1
2 -> 2
3 -> 3


f1:

0 -> 0
1 -> 2
2 -> 1
3 -> 3
```

Running the Code



Running the Code




```
This poset has 6 automorphisms

The automorphisms on this poset are :

f0:              f3:

0 -> 0           0 -> 1
1 -> 1           1 -> 2
2 -> 2           2 -> 0
3 -> 3           3 -> 4
4 -> 4           4 -> 5
5 -> 5           5 -> 3


f1:              f4:

0 -> 0           0 -> 2
1 -> 2           1 -> 0
2 -> 1           2 -> 1
3 -> 4           3 -> 5
4 -> 3           4 -> 3
5 -> 5           5 -> 4


f2:              f5:

0 -> 1           0 -> 2
1 -> 0           1 -> 1
2 -> 2           2 -> 0
3 -> 5           3 -> 3
4 -> 4           4 -> 5
5 -> 3           5 -> 4
```

Running the Code

For posets P of the following form,

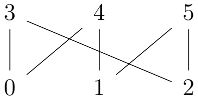


Figure: n = 3

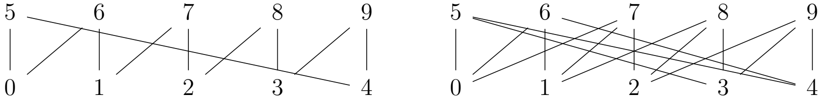


Figure: n = 5

Running the Code

For posets P of the following form,

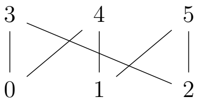


Figure: n = 3

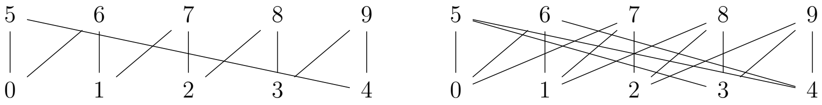
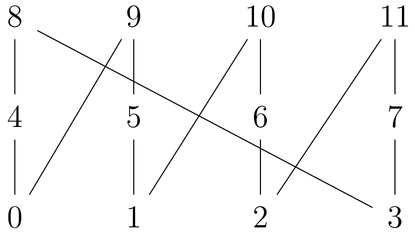


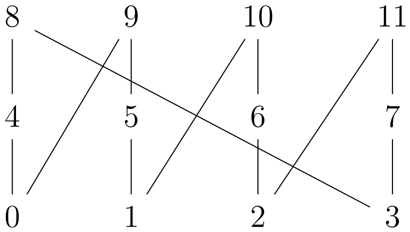
Figure: n = 5

we think that Aut $P = D_{2n}$ for $3 \leq n \leq 7$

Running the Code

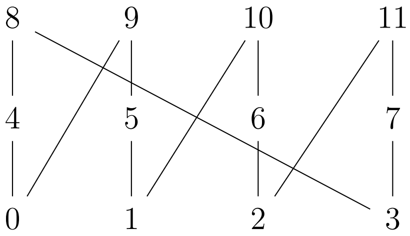


Running the Code



This takes too long and does not provide an output.
Why does this happen and how to we fix it?

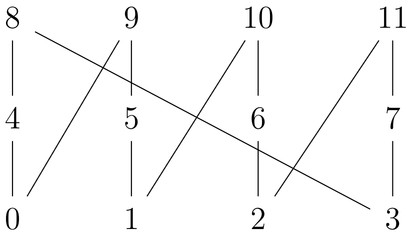
Making the Code More Efficient



$|S_{12}| = 12! = 479001600.$

The program has to check a lot of automorphisms.

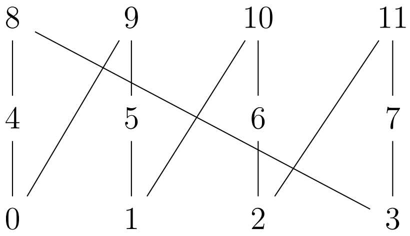
Making the Code More Efficient



$|S_{12}| = 12! = 479001600.$
The program has to check a lot of automorphisms.

How do we reduce the number of things the program must check?

Making the Code More Efficient

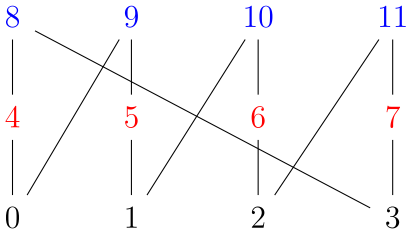


$|S_{12}| = 12! = 479001600.$

The program has to check a lot of automorphisms.

How do we reduce the number of things the program must check?
We can add restrictions.

Making the Code More Efficient

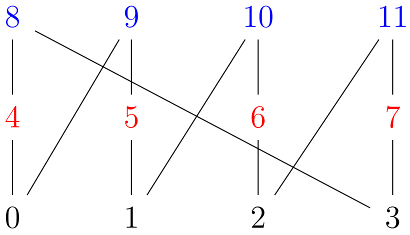


$|S_{12}| = 12! = 479001600.$
The program has to check a lot of automorphisms.

How do we reduce the number of things the program must check?
We can add restrictions.

We find all of the bijections for each height and their combinations

Making the Code More Efficient

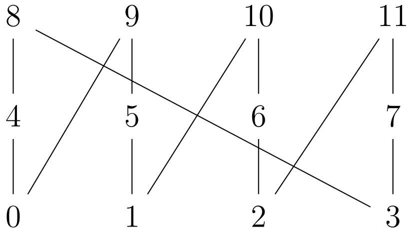


$|S_{12}| = 12! = 479001600.$
The program has to check a lot of automorphisms.

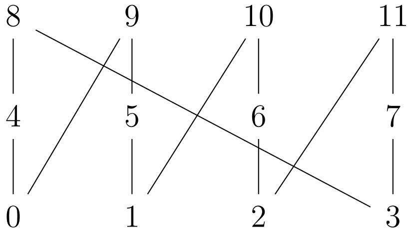
How do we reduce the number of things the program must check?
We can add restrictions.

We find all of the bijections for each height and their combinations
We only need to check $|S_4| \times |S_4| \times |S_4| = 13824$ automorphisms.

Running the Code



Running the Code




```
This poset has 4 automorphisms
The automorphisms on this poset are :

f0:

0 -> 0
1 -> 1
2 -> 2
3 -> 3
4 -> 4
5 -> 5
6 -> 6
7 -> 7
8 -> 8
9 -> 9
10 -> 10
11 -> 11

f1:

0 -> 1
1 -> 2
2 -> 3
3 -> 0
4 -> 5
5 -> 6
6 -> 7
7 -> 4
8 -> 9
9 -> 10
10 -> 11
11 -> 8

f2:

0 -> 2
1 -> 3
2 -> 0
3 -> 1
4 -> 6
5 -> 7
6 -> 4
7 -> 5
8 -> 10
9 -> 11
10 -> 8
11 -> 9

f3:

0 -> 3
1 -> 0
2 -> 1
3 -> 2
4 -> 7
5 -> 4
6 -> 5
7 -> 6
8 -> 11
9 -> 8
10 -> 9
11 -> 10
```

Future Work

- Conjectures of $\beta(Z_{p^k})$
- The free group of rank r is realizable for all $r > 1$
- Construction of posets P such that $\mathrm{Aut}\, P = D_{2P}$
- Extending the code to identity $\mathrm{Aut}\, P$ for a poset P

Acknowledgements

We would like to acknowledge and thank the NSF[1], Cory Colbert, Rodrigo Smith, Edray Goins, Alex Barrios, and all other professors and students who have been part of PRiME '22. We would also like to thank the NSF and Pomona College.

[1]This material is based upon work supported by the National Science Foundation under Grant No. DMS-2113782. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Jonathan A. Barmak.
Automorphism groups of finite posets ii.
arXiv, 2020.

Jonathan Ariel Barmak and Elias Gabriel Minian.
Automorphism groups of finite posets.
Discrete Mathematics, 309(10):3424–3426, 2009.

G. Birkoff.
On groups of automorphisms.
Rev. Un. Math. Argentina, 11:155–157, 1946.

Robert Frucht.
Herstellung von graphen mit vorgegebener abstrakter gruppe.
Compositio Mathematica (German), 6:239–250, 1939.

M.C. Thorton.
Spaces with given homeomorphism groups.
Proceedings of the American Mathematical Society, 33:127–131, 1972.

Thank You!