

## 5.1.4

**Claim:** Let  $G$  be any group and  $H$  any subgroup. Then  $G$  acts on the set  $G/H$  of left cosets of  $H$  in  $G$  by left multiplication. Then action is transitive.

**Proof:**  $\forall aH \in G/H, \forall g \in G, g \cdot G/H = gaH = (ga)H$ , since  $ga \in G$  by closure,  $gaH \in G/H$ .  $\forall bH \in G/H, g = ba^{-1} \in G$  so that  $gaH = bH$ , so we proved that the number of orbit  $\mathcal{O}(x)$  is 1.

As a result,  $G$  acts on the set  $G/H$  of left cosets of  $H$  in  $G$  by left multiplication is transitive. ■

## 5.1.6

**Claim:** Let  $G$  act on  $X$ , and let  $x \in X$ . We have  $\text{Stab}(x) \subseteq G$  and if  $x$  and  $y$  are in the same orbit, then the subgroups  $\text{Stab}(x)$  and  $\text{Stab}(y)$  are conjugate subgroups.

**Proof:** Take  $s \in \text{Stab}(x) = \{g \in G : g \cdot x = x\}$ , obviously  $s \in G$ , so  $\text{Stab}(x) \subseteq G$ .

If  $x$  and  $y$  are in the same orbit, then  $\exists g \in G$  that  $y = g \cdot x = gxg^{-1}$ . Then for  $\text{Stab}(x) = \{g \in G : g \cdot x = x\} = \{g \in G : gxg^{-1} = x\}$  and  $\text{Stab}(y) = \{g \in G : g \cdot y = y\} = \{g \in G : gyg^{-1} = y\}$ .  $\forall g' \in G, g'\text{Stab}(x)g'^{-1} = \{g \in G : g'g \cdot xg^{-1}g'^{-1} = x\} = \{g \in G : g'gxg^{-1}g'^{-1} = x\} = \{g \in G : g'gxg^{-1}g'^{-1} = x\} = \{g \in G : g'gxg^{-1}g'^{-1} = x\} = \{g \in G : g \cdot y = y\} = \text{Stab}(y)$ . ■

## 5.2.2

**Claim:** 2 red beads, 2 green beads and 2 violet heads can make 11 necklaces.

**Proof:** Since the symmetries consist 6 rotational symmetries and 6 reflectional symmetries. We choose  $G = D_6$ , so  $|G| = 12$ . And since

$$|X| = \frac{6!}{2!2!2!} = 90,$$

so  $|\text{Fix}(e)| = 90, |\text{Fix}(r^3)| = 6, |\text{Fix}(a)| = |b| = |\text{Fix}(c)| = |\text{Fix}(d)| = |\text{Fix}(e)| = |\text{Fix}(f)| = 6$ .

Thus, by Burnside's Lemma,

$$n = \frac{1}{12}(90 + 7 \cdot 6) = 11. \blacksquare$$

### 5.3.7

**Claim:**  $\text{Aut}(\mathbb{Z}_2 \times \mathbb{Z}_2) \cong S_3$ .

**Proof 1:** Since  $\mathbb{Z}_2 \times \mathbb{Z}_2$  can be grouped into a vector of  $2D$  vector, which is basically a  $2 \times 2$  matrix, in which each elements can only be either 1 or 0. So, obviously, the only way to transform a  $2 \times 2$  matrix to a  $2 \times 2$  matrix is to multiply a  $2 \times 2$  matrix, which means  $\text{Aut}(\mathbb{Z}_2 \times \mathbb{Z}_2) \subseteq \text{GL}(2, \mathbb{R})$ . Of them, only 6 matrices are invertible including:

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

Those matrices can be easily checked to be a group by doing the multiplication.

Since there're only 2 kinds of groups of order 6 up to isomorphism  $S_3$  and  $\mathbb{Z}_6$ . Since we can check that

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

So we know  $\text{Aut}(\mathbb{Z}_2 \times \mathbb{Z}_2)$  is not commutative, which means it isn't isomorphic to  $\mathbb{Z}_6$ , and thus we know that  $\text{Aut}(\mathbb{Z}_2 \times \mathbb{Z}_2) \cong S_3$ .  $\blacksquare$

**Proof 2:** Since in any group mapping, we always send identity to identity in order to preserve the structure. So what really matters is that to which we send  $\{a, b, c\}$  to from the group of  $\mathbb{Z}_2 \times \mathbb{Z}_2 = \{e, a, b, c\}$ . Then for each homomorphism there exists a unique permutation in  $S_3$  to fulfill this requirement.

Then  $\forall g \in S_3$ , by permutation composition,  $g$  acts on  $\{a, b, c\}$  sends bijectively send  $a, b, c$  to  $\{a, b, c\}$  so we can see such permutation determines an automorphism by definition.

As a result,  $\text{Aut}(\mathbb{Z}_2 \times \mathbb{Z}_2) \cong S_3$ .  $\blacksquare$