MT10/P19 - TP1 : Groupes d'ordre 4

Table des matières

Question 0: (à rédiger !) Rappeler la définition de « groupe » et celle de « morphisme de groupes ».

1 Test d'une loi

1.1 Codage d'une loi sur un ensemble

Bien que `SageMath` reconnaisse le type « ensemble », il sera préférable de coder les ensembles par des listes, afin de repérer sans ambiguïté ses éléments. Par exemple, l'ensemble à 4 éléments

$$E = \Big\{ a; b; c; d \Big\}$$

sera représenté par la liste

```
E=[a, b , c, d]
```

de sorte que les éléments de E soient numérotés : $E[0] = a$, ..., $E[3] = d$.

Une loi $*$ sur E,

$$\begin{array}{rcl} * : E \times E & \longrightarrow & E \\ (x, y) & \longmapsto & x * y \end{array}$$

peut donc se coder par un tableau [1] d'entiers $t[i][j] \in [\![0, 3]\!]$ tel que

$$(\forall i, j \in [\![0, 3]\!]) \qquad E[t[i][j]] = E[i] * E[j]$$

1. En fait une liste de listes.

Question 1: Former (à la main) la liste des éléments E et le tableau correspondant de loi t pour les groupes $(\mathbb{Z}/4\mathbb{Z}, +, 0)$ et $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$, puis faire la même chose à l'aide de `SageMath`.

Indications : Avec `SageMath`, on forme d'abord le groupe souhaité, puis on lui applique des méthodes idoines pour récupérer la liste E et le tableau t souhaités. `SageMath` permet de coder des groupes cycliques comme sous-groupes de groupes de permutation. Par exemple :

```
C4=groups.permutation.Cyclic(4)
```

Taper ensuite `C4.` suivi de TAB pour obtenir la liste des méthodes associées au groupe `C4` que vous venez d'instancier. On récupère la table d'opération par la méthode `cayley_table()`. Celle-ci n'est pas au format souhaité, mais on peut lui appliquer

— la méthode `table()` pour récupérer le tableau t,
— la méthode `column_keys()` pour récupérer la liste E.

Construire de même le groupe `C2` puis son produit cartésien avec lui-même à l'aide de la commande `cartesian_product([C2,C2])`.

1.2 Elément neutre

S'il existe, l'élément neutre est $E[\alpha]$ où $\alpha \in [\![0, 3]\!]$ vérifie

$$(\forall i \in [\![0, 3]\!]) \qquad t[i][\alpha] = i \quad \text{et} \quad t[\alpha][i] = i$$

Question 2: Ecrire une procédure `SageMath` qui teste si une loi donnée par un tableau t admet un élément neutre, et qui, s'il existe, le renvoie. Tester sur les tables d'opération des groupes $(\mathbb{Z}/4\mathbb{Z}, +, 0)$ et $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$, et sur une autre table choisie au hasard. Faire le même test directement à l'aide de méthode `SageMath`.

1.3 Elément symétrique

L'élément $E[i]$ admet $E[j]$ comme symétrique si

$$t[i][j] = \alpha \quad \text{et} \quad t[j][i] = \alpha$$

Question 3: Ecrire une procédure `SageMath` qui, pour chaque élément, recherche son élément symétrique, et qui, s'il existe, le renvoie. Tester sur les tables d'opération des groupes $(\mathbb{Z}/4\mathbb{Z}, +, 0)$ et $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$, et former la table des symétriques. Faire la même chose directement à l'aide de méthode `SageMath`.

1.4 Associativité

La loi codée par le tableau t est associative si

$$(\forall i, j, k \in [\![0, 3]\!]) \qquad t[t[i][j]][k] = t[i][t[j][k]]$$

Question 4: Ecrire une procédure `SageMath` qui teste si une loi donnée par un tableau t est associative. Tester sur les tables d'opération des groupes $(\mathbb{Z}/4\mathbb{Z}, +, 0)$ et $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$. Faire la même chose directement à l'aide de méthode `SageMath`.

2 Test d'un morphisme

Considérons deux groupes finis $(G, *, e)$ et $(G', *', e')$ d'ordre n et n' respectivement. Leurs éléments sont codés respectivement par les listes (notées encore) G et G' et leurs lois sont respectivement codées par les tableaux t et t'. Une application

$$\begin{array}{rcl} f : G & \longrightarrow & G' \\ x & \longmapsto & f(x) \end{array}$$

se codera par une liste (notée encore) f de n entiers appartenant à $[\![0, n'-1]\!]$ de sorte que :

$$(\forall k \in [\![0, n-1]\!]) \qquad G'[f[k]] = f(G[k])$$

L'application f est un morphisme (de groupes) si

$$(\forall x, y \in G) \qquad f(x * y) = f(x) *' f(y),$$

qu'on codera par

$$(\forall i, j \in [\![0, n-1]\!]) \qquad f[t[i][j]] = t'[f[i]][f[j]].$$

Question 5:

1. Ecrire une procédure `SageMath` qui teste si une application donnée entre deux groupes est un morphisme.
2. Utiliser cette procédure pour construire tous les automorphismes de $(\mathbb{Z}/4\mathbb{Z}, +, 0)$, puis de $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$.
   (On rappelle qu'un automorphisme de G est une bijection de G dans lui-même qui est aussi un morphisme.)
3. Montrer que les groupes $(\mathbb{Z}/4\mathbb{Z}, +, 0)$ et $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$ ne sont pas isomorphes.
4. Faire la même chose directement à l'aide de méthode `SageMath`.
   <u>Indications</u> : Pour former un morphisme d'un groupe de permutation vers un autre, on peut utiliser la commande `PermutationGroupMorphism`. Pour tester si un groupe est isomorphe à un autre, utiliser la méthode `is_isomorphic`, qui elle aussi ne fonctionne qu'avec des groupes de permutations. $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$ s'obtient comme groupe de permutations par `groups.permutation.KleinFour()`.

3 Où l'on identifie tous les groupes d'ordre 4

Soit un ensemble à 4 éléments : $E = \{a_0; a_1; a_2; a_3\}$. On veut en faire un groupe $(E, *, e)$.

Question 6: Dire pourquoi la table de $*$ doit être un carré latin, *i.e.* chaque élément apparaît une fois et une seule dans chaque ligne et chaque colonne.

Question 7:

Si on choisit $e = a_0$, et en imposant le fait que la table d'un groupe doit être un carré latin, on obtient ces 4 tables :

(I)

$x\backslash x'$	a_0	a_1	a_2	a_3
a_0	a_0	a_1	a_2	a_3
a_1	a_1	a_0	a_3	a_2
a_2	a_2	a_3	a_1	a_0
a_3	a_3	a_2	a_0	a_1

(II)

$x\backslash x'$	a_0	a_1	a_2	a_3
a_0	a_0	a_1	a_2	a_3
a_1	a_1	a_0	a_3	a_2
a_2	a_2	a_3	a_0	a_1
a_3	a_3	a_2	a_1	a_0

(III)

$x\backslash x'$	a_0	a_1	a_2	a_3
a_0	a_0	a_1	a_2	a_3
a_1	a_1	a_2	a_3	a_0
a_2	a_2	a_3	a_0	a_1
a_3	a_3	a_0	a_1	a_2

(IV)

$x\backslash x'$	a_0	a_1	a_2	a_3
a_0	a_0	a_1	a_2	a_3
a_1	a_1	a_3	a_0	a_2
a_2	a_2	a_0	a_3	a_1
a_3	a_3	a_2	a_1	a_0

1. Dire si ces 4 tables sont bien des tables de groupe et si ce sont les seules.
2. Parmi ces tables, déterminer celles qui sont isomorphes à $(\mathbb{Z}/4\mathbb{Z}, +, 0)$, puis celles isomorphes à $(\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}, +, 0)$.

Indication : Pour générer les tables avec `SageMath`, utiliser la commande `Permutations`, après consultation de l'aide (qui s'obtient avec `Permutations?`).

4 Recherche et exploration avec `SageMath`

Question 8: (ouverte) Explorer quelques groupes avec `SageMath` : ordre, sous-groupes, centre, ..., et en particulier :

1. **(Test du Théorème de Lagrange)** Un théorème de Lagrange affirme que l'ordre d'un sous-groupe divise l'ordre du groupe [2]. Tester sur quelques groupes.
   Indications : Utiliser une commande du type `map(order,G.subgroups())`, où `G` désigne un groupe (dans `SageMath`), puis tester la divisibilité. Pour parcourir les groupes de `SageMath`, commencer par `groups?`, consulter aussi le tutoriel thématique « Group Theory and Sage ».
2. **(graphe de Cayley)** Qu'appelle-t-on graphe de Cayley pour un groupe (et un système de générateurs) ? Quel rapport avec la table d'opération du groupe ? Comment `SageMath` représente-t-il les graphes de Cayley ?
   Indication : Commencer par lire la page correspondante sur `Wikipedia`.

Question 9: (défi) Trouver un carré latin de taille minimale qui n'est pas la table d'une loi de groupe.
Indication : Force brute : tester en taille 5, 6, ...

2. s'il est fini.