TP : LE GROUPE LIBRE

CPES FEYDER II

À la fin de la séance de TP, vos programmes seront récupérés pour être notés.

Le *groupe libre* $\mathsf{GrpLib}(X)$ construit sur un ensemble X est l'ensemble de toutes les *listes finies* $[\epsilon_1 x_1, \cdots, \epsilon_n x_n]$, $x_i \in X$ et $\epsilon_i = \pm 1$ pour $i = 1, \cdots, n$ (la *liste vide* $[\quad]$ y compris). Par exemple si $X = \{a, b, c\}$, alors $[-a, b, a, -c, c]$, $[-b, a, b]$, $[-b, -c, a, -c, c]$ sont des éléments de $\mathsf{GrpLib}(X)$. Cet ensemble devient un groupe lorsqu'on le munit de l'opération de *concaténation* avec de surcroît la relation $[x] \cdot [-x] = [\quad] = [-x] \cdot [x]$ quel que soit $x \in X$. En particulier, $[\quad]$ est l'élément neutre du groupe ($L \cdot [\quad] = L = [\quad] \cdot L$ quel que soit $L \in \mathsf{GrpLib}(X)$). Remarquons qu'une liste $[\epsilon_1 x_1, \cdots, \epsilon_n x_n]$ n'est alors rien d'autres que la concaténation des listes réduites à un élément $[\epsilon_1 x_1] \cdots [\epsilon_n x_n]$: ainsi on a $[-a, c, c, a, -b] = [-a] \cdot [c] \cdot [c] \cdot [a] \cdot [-b]$. Voyons des exemples de calculs dans $\mathsf{GrpLib}(X)$: $[-a, b, a, -c] \cdot [-b, a, b] = [-a, b, a, -c, -b, a, b]$ ou encore

$$\begin{aligned}
[-b, -a, b] \cdot [-b, a, -c] &= [-b, -a] \cdot \underbrace{[b][-b]}_{=[\quad]} \cdot [a, -c] \\
&= [-b, -a] \cdot [\quad] \cdot [a, -c] \\
&= [-b, -a] \cdot [a, -c] \\
&= [-b] \cdot \underbrace{[-a] \cdot [a]}_{=[\quad]} \cdot [-c] \\
&= [-b][\quad][-c] \\
&= [-b] \cdot [-c] \\
&= [-b, c] \ .
\end{aligned}$$

Une liste $[\epsilon_1 x_1, \cdots, \epsilon_n x_n]$ est dite sous *forme réduite* s'il n'existe pas d'indice $i = 1, \cdots, n-1$, tel que $x_i = -x_{i+1}$ ou $-x_i = x_{i+1}$ (en d'autres termes, $[\epsilon_1 x_1, \cdots, \epsilon_n x_n]$ ne peut pas être réduite en utilisant la règle $[x][-x] = [\quad]$ ou $[-x][x] = [\quad]$ pour $x \in X$ car il n'existe pas deux éléments consécutifs x et $-x$ ou $-x$ et x). Remarquons au passage que la liste vide est sous forme réduite, de même que toute liste réduite à un seul élément. À titre d'exemple, $[-a, b, -b, c]$ n'est pas sous forme réduite car on peut la réécrire $[-a, c]$ (puisque $[b][-b] = [\quad]$), alors que $[a, -b, -a, -c, -c, a, a]$ l'est. On peut démontrer (mais on se contentera de l'admettre) que toute liste de $\mathsf{GrpLib}(X)$ peut s'écrire sous forme réduite (et que cette forme réduite est unique).

Questions :

(1) Écrire une procédure `reduction` qui prend en argument un élément du groupe libre et renvoie sa forme réduite.

(2) Le but de cette question est d'écrire une procédure `produit` qui prend deux listes en argument G et D que l'on suppose être des listes écrites sous forme réduite (on ne le vérifie pas) et qui renvoie le produit $G \cdot D$ écrit sous forme réduite. En d'autres termes, la procédure `produit` ne se contente pas de renvoyer la liste concaténée $[\mathtt{op}(G), \mathtt{op}(D)]$ mais effectue toutes les réductions possibles également. Ainsi $\mathtt{produit}([-a, b], [-b, c]) = [-a, c]$. **Pour cette question, vous ne devez pas utiliser la procédure `reduction`.**

(3) Écrire une autre procédure `produit2` qui effectue le même traitement que la procédure `produit` mais qui cette fois utilise la procédure `reduction` de la question 1.

(4) Écrire une procédure `inverse` qui prend un élément du groupe libre en argument et renvoie son inverse (sous forme réduite).

(5) Écrire une procédure `puissance` qui prend deux arguments : un entier naturel n et un élément du groupe libre L, et qui renvoie la *puissance nième de* L : $\underbrace{L \cdot L \cdots L \cdot L}_{n \text{ facteurs}}$ sous sa forme réduite. En particulier `puissance(0,L)` doit renvoyer $[\quad]$. Par exemple, si $L = [-a, b, a]$, on a `puissance(1,L)` renvoie L, `puissance(2,L)` renvoie $[-a, b, a] \cdot [-a, b, a]$ sous forme réduite, soit $[-a, b, b, a]$, puis `puissance(3,L)` renvoie $[-a, b, a] \cdot [-a, b, b, a] = [-a, b, b, b, a]$, *etc.*