TP 3 - Définitions Inductives en Coq

1 Entiers de Gauss

Les entiers de Gauss sont les nombres complexes de la forme $a + ib$ avec a et b des entiers relatifs.

On dispose dans `Coq` d'une théorie sur les entiers relatifs $\mathbb{Z}$. Concrètement cela veut dire que l'on dispose d'un type de données `Z` de sorte `Set`, de constantes 0 et 1 et des opérations habituelles : l'addition `Zplus : Z -> Z -> Z`, l'opposé `Zopp : Z -> Z`, la soustraction `Zminus : Z -> Z -> Z` et la multiplication `Zmult : Z -> Z -> Z`. On dispose d'une tactique `ring` permettant de démontrer automatiquement les égalités sur les expressions à valeurs entières. Par exemple la formule suivante

$$\forall a\ b\ c : Z, a * (b + c - d) = a * b - d * a + a * c$$

peut se démontrer grâce à la tactique `ring`.

1- Proposer une structure de données inductive `gauss` pour décrire les entiers de Gauss. Comment construire les valeurs 0 (noté $g0$), $g1$ et gi ?

2- Proposer des opérations d'addition, soustraction et multiplication sur les entiers de Gauss.

3- Comment procéder pour démontrer formellement en `Coq` que $\forall a : \texttt{gauss}, 0 + a = a$?

4- Prouver en Coq l'associativité et la commutativité de l'addition pour les entiers de Gauss. Les théorèmes établissant ces propriétés pour les entiers relatifs sont déjà disponibles dans `Coq` :

```
Zplus_comm : forall a b : Z, a+b = b+a.
Zplus_assoc : forall a b c : Z (a+b)+c = a+(b+c).
Zmult_comm : forall a b : Z, a*b = b*a.
```

2 Pour aller plus loin...

On va implanter un type inductif permettant de représenter les jours de la semaine.
1- Proposer une structure de données pour ce type.
2- Programmer en `Coq` des fonctions `jour_suivant` et `jour_precedent`.
3- Prouver les propriétés suivantes :
— $\forall j : \texttt{jour}, \texttt{jour\_suivant}\ (\texttt{jour\_precedent}\ j) = j$,
— $\forall j : \texttt{jour}, \texttt{jour\_precedent}\ (\texttt{jour\_suivant}\ j) = j$.
4- De manière récursive, écrire une fonction d'itération `iter` qui applique, à un jour donné, une fonction `f : jour -> jour` composée n fois (n un entier naturel passé également en paramètre). On évitera la définition en récursivité terminale. Prouver ensuite la propriété suivante : $\forall j : \texttt{jour}, \texttt{iter}\ 7\ \texttt{jour\_suivant}\ j = j$.
5- Prouver un théorème similaire avec `jour_precedent`.
6- Prouver par induction que, pour tout couple d'entiers `n` et `m`, et pour toute fonction `f` de type `jour -> jour`, on a $\forall j : \texttt{jour},\ \texttt{iter}\ (n + m)\ \texttt{f}\ j = \texttt{iter}\ n\ \texttt{f}\ (\texttt{iter}\ m\ \texttt{f}\ j)$.
7- Prouver que, pour tout entier n et jour j, on a : $\texttt{iter}\ (7 * n)\ \texttt{jour\_suivant}\ j = j$.