Option Informatique en Spé MP et MP*
Calculs dans l'algèbre des p.u.p. de $\mathbb{N}$: un corrigé

Quelques outils

▶ Les deux outils suivants sont bien connus de vous, et viennent compléter utilement la bibliothèque Caml ; notez le changement du nom : `filtre` devient `tamis` (pour éviter toute confusion avec les filtres de Caml).

```
let rec intervalle a b =
 if a>b then [] else a::(intervalle (a+1) b) ;;

let rec tamis p = function
 | [] -> []
 | t::q when p t -> t::(tamis p q)
 | _::q -> tamis p q ;;
```

Des réponses

Question 1 • Soient A et B deux p.u.p. ; notons n_A et n_B leur rangs d'entrée dans la période, et ${}_A$ et p_B leurs périodes. Alors $A \cup B$ est une p.u.p. : son rang d'entrée est au plus $\max(n_A, n_B) + \mathrm{ppcm}(p_A, p_B)$, sa période est au plus $\mathrm{ppcm}(p_A, p_B)$.

• Le complémentaire d'une p.u.p. est clairement une p.u.p. qui a même rang d'entrée et même période que A.

• Enfin, la formule $A \cap B = \overline{\overline{A} \cup \overline{B}}$ montrez que l'intersection de deux p.u.p. est elle-même une p.u.p.

Question 2 La condition $n < 50$ et $n \equiv 3$ [7] nous donne l'ensemble $\{3, 10, 17, 24, 31, 38, 45\}$. La condition $n > 25$ et $n \equiv 5$ [8] nous donne les nombres 29, 37 et 45, plus tous les entiers de la forme $8k + 5$, avec $k \geqslant 6$. L'ensemble A est bien une p.u.p. : le rang d'entrée est 44, la période est 8 et la base est la liste `[3;10;17;24;29;31;37;38;45]`.

Question 3 Si $k < n_0 + p$, alors $k \in A$ ssi k est dans la base. Sinon, $k \in A$ ssi $k - n_0$ est multiple de p. Traduction en Caml :

```
let reduit k mypup =
 ((k - mypup.n_zero) mod mypup.p) + mypup.n_zero ;;

let appartient mypup = function
 | k when mem k mypup.base -> true
 | k when k < mypup.n_zero -> false
 | k -> mem (reduit k mypup) mypup.base ;;
```

Question 4 Notons n_A et n_B les rangs d'entrée de A et B, et p_A et p_B leurs périodes respectives. Notons $n_0 = \max(n_A, n_B)$ et $p = p_A p_B$. Pour montrer que B est contenue dans A, il suffit de vérifier que les éléments de B qui sont dans l'intervalle $[0, n_0 + p - 1]$. La fonction `get_intrv` va faire une partie de ce travail ;

```
let get_intrv mypup kmax =
 tamis (appartient mypup) (intervalle 0 kmax) ;;

let contient pupA pupB =
 let z0 = max pupA.n_zero pupB.n_zero
 and per = pupA.p * pupB.p in
 let tete = get_intrv pupB (z0 + per) in
 for_all (appartient pupA) tete ;;
```

Question 5 Voici une fonction calculant le complémentaire d'une p.u.p. :

```
let complémentaire mypup =
 let mon_filtre k = not(mem k mypup.base) in
 let zone = intervalle 0 (mypup.n_zero + mypup.p - 1) in
 let newbase = tamis mon_filtre zone in
```

```
 { n_zero = mypup.n_zero ;
   p = mypup.p ;
   base = newbase
 } ;;
```

Question 6 Ici, il y a une petite astuce : une p.u.p. est vide si sa période p est nulle et si la liste `base` est vide.

```
let est_vide mypup = mypup.p = 0 && mypup.base = [] ;;
```

Question 7 Voici une fonction réalisant la réduction :

```
let réduction mypup =
 let seuil = mypup.n_zero + mypup.p in
 { n_zero = mypup.n_zero ;
   p = mypup.p ;
   base = tamis (prefix >= seuil) mypup.base } ;;
```

Question 8 La fonction suivante ne vise pas l'efficacité maximale :

```
let intersection pupA pupB =
 let z0 = max pupA.n_zero pupB.n_zero
 and per = pupA.p * pupB.p in
 let liste = get_intrv pupB (z0 + per - 1) in
 { n_zero = z0 ;
   p = per ;
   base = tamis (appartient pupA) liste } ;;
```

On peut améliorer le programme suivant, en calculant le PGCD des périodes des p.u.p. A et B.

Question 9 Très facile, avec la formule $A \cup B = \overline{\overline{A} \cap \overline{B}}$:

```
let réunion pupA pupB =
 let pupAbarre = complémentaire pupA
 and pupBbarre = complémentaire pupB
 in complémentaire( intersection pupAbarre pupBbarre ) ;;
```

FIN