

INFO0947: Rapport Projet 1

Groupe 23: Andrew WILLEMS, Pierre LORENZEN

Table des matières

1	Description du problème	3
2	Découpe en sous-problème	3
2.1	Relation entre les sous-problème	3
3	Sous-problème 2	3
3.1	Spécification du prédicats	3
3.2	Objets Utilisés	3
3.3	Signature	4
3.4	Spécification	4
3.5	Code	4
4	Sous-problème 1	4
4.1	Spécification des prédicats	4
4.1.1	Spécification du prédicat B	4
4.2	Spécification du prédicat A	5
4.3	Spécification du prédicat C	5
4.4	Spécification du prédicat D	5
5	Invariant	5
5.1	Invariant graphique	5
5.2	Invariant formel	6

1 Description du problème

Il est demandé de filtrer un tableau d'entiers par rapport à une certaine propriété p .
On peut représenter le problème comme suit :

T :	0	i	N-1

Avec la zone **bleu** qui concerne la zone filtrée du tableau(A), la taille de la zone filtrée(B) et tout les éléments qui s'y trouve se trouvait dans le même ordre dans le tableau initiale(C). La zone **brune** concerne la zone non filtrée ce qui veut dire que la zone est remplie de 0(D).

2 Découpe en sous-problème

Pour avoir plus de facilité à résoudre le problème nous pouvons le séparer en 2 sous-problèmes. Le premier sous-problème étant le problème principal : Filtrer le tableau par rapport à une certaine propriété p . Les éléments respectant cette propriété sont à l'avant du tableau et dans le même ordre que dans le tableau initiale. Les éléments ne respectant pas la propriété sont placé en bout de tableau et leur valeur est mise à 0.

Le deuxième sous-problème est la permutation de 2 éléments d'un même tableau sans changer leur valeur.

2.1 Relation entre les sous-problème

$$SP_2 \subset SP_1$$

3 Sous-problème 2

Nous allons commencer par définir un prédicat avec leur objets utilisés et une notation pour pouvoir décrire ce sous-problème plus facilement.

3.1 Spécification du prédicats

1. Objets Utilisés

- ★ T : Un tableau d'entier initialisé de taille N.
- ★ $N > 0$ ($\in \mathbb{N}$)
- ★ T_0 : Le tableau T avant modification.

2. Signature

$$Permutation(T, N, T_0)$$

3. Spécification

$$Permutation(T, N, T_0) \equiv (\forall i, 1 \leq i < N, (\exists j, 0 \leq j < N, T_0[j] = T[i]))$$

3.2 Objets Utilisés

Ensuite, nous allons décrire les objets utilisés par le sous-problème 2.

- ★ T : Un tableau d'entier initialisé de taille N.
- ★ $N > 0$ ($\in \mathbb{N}$)

- ★ i : est la destination de la valeur à permuter.
- ★ j : est le depart de la valeur à permuter.

3.3 Signature

La signature du sous-problème 2 est la suivante :

```
1 void perm(int *T, int const N, int const i, int const j);
```

3.4 Spécification

Ici, nous décrivons la précondition et la postcondition du sous-problème.

```
1 /**
2  * @Précond :  $N > 0 \wedge T_{init}$ 
3  * @Postcond :  $N = N_0 \wedge Permutation(T, N, T_0)$ 
4  */
5 void perm(int *T, int const N, int const i, int const j);
```

3.5 Code

Dans cette section, nous écrivons le bout de code correspondant au sous-problème 2 avec ses assertions intermédiaires pour nous assurer la validité du code.

```
1 /**
2  * @Précond :  $N > 0 \wedge T_{init}$ 
3  * @Postcond :  $N = N_0 \wedge Permutation(T, N, T_0)$ 
4  */
5 void perm(int *T, int const N, int const i, int const j){
6     assert(T != NULL && N > 0);
7     // @Précond ( $\Rightarrow N > 0 \wedge T_{init} \wedge 0 \leq i, j < N$ )
8     int k = T[i];
9     // ( $\Rightarrow N > 0 \wedge T_{init} \wedge 0 \leq i, j < N \wedge k = T_0[i]$ )
10    T[i] = T[j];
11    // ( $\Rightarrow N > 0 \wedge T_{init} \wedge 0 \leq i, j < N \wedge k = T_0[i] \wedge T[i] = T_0[j] \wedge T[j] = k$ )
12    T[j] = k;
13    // ( $\Rightarrow N > 0 \wedge T_{init} \wedge 0 \leq i, j < N \wedge k = T[i] \wedge T[j] = T_0[j] \wedge T[j] = T_0[i]$ )
14    // ( $\Rightarrow N > 0 \wedge T_{init} (\forall i, 1 \leq i < N, (\exists j, 0 \leq j < N, T_0[j] = T[i]))$ )
15    // ( $\Rightarrow N = N_0 \wedge Permutation(T, N, T_0)$ )
16    // Postcond
17 }
```

4 Sous-problème 1

Nous allons commencer par définir quelques prédicats avec leur objets utilisés et quelques notations pour pouvoir décrire ce sous-problème plus facilement.

4.1 Spécification des prédicats

4.1.1 Spécification du prédicat B

1. Objets Utilisés

- ★ T : Un tableau d'entier initialisé de taille N .
- ★ $N > 0$ ($\in \mathbb{N}$)
- ★ p : Une certaine propriété.

2. Signature
 $TailleZoneFiltree(T, N, p)$
3. Spécification
 $TailleZoneFiltree(T, N, p) \equiv \#i, 0 \leq i < N, p(T[i])$

4.2 Spécification du prédicat A

1. Objets Utilisés
 - ★ T : Un tableau d'entier initialisé de taille N .
 - ★ $taille$: taille de la zone filtrée.
 - ★ p : Une certaine propriété.
2. Signature
 $ZoneFiltree(T, p, taille)$
3. Spécification
 $ZoneFiltree(T, p, taille) \equiv \forall i, 0 \leq i < taille, p(T[i])$

4.3 Spécification du prédicat C

1. Objets Utilisés
 - ★ T_0 : Le tableau T avant modification.
 - ★ T : Un tableau d'entier initialisé de taille N .
 - ★ $N > 0$ ($\in \mathbb{N}$)
 - ★ $taille$: taille de la zone filtrée.
2. Signature
 $LienTableau(T_0, T, N, taille)$
3. Spécification
 $LienTableau(T_0, T, N, taille) \equiv (\forall i, 1 \leq i < taille, (\exists j, 0 \leq j < N, T_0[j] = T[i]) \wedge (\exists k, 0 \leq k < j, T_0[k] = T[i - 1]))$

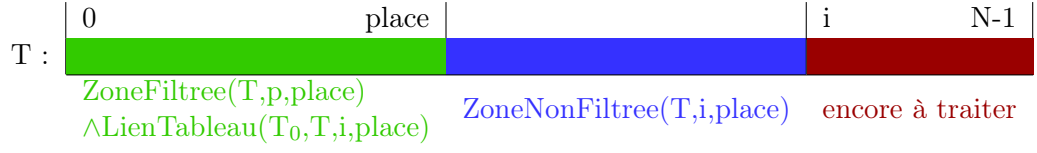
4.4 Spécification du prédicat D

1. Objets Utilisés
 - ★ T : Un tableau d'entier initialisé de taille N .
 - ★ $N > 0$ ($\in \mathbb{N}$)
 - ★ $taille$: taille de la zone filtrée.
2. Signature
 $ZoneNonFiltree(T, N, taille)$
3. Spécification
 $ZoneNonFiltree(T, N, taille) \equiv \forall i, taille < i < N, T[i] = 0$

5 Invariant

5.1 Invariant graphique

Avec la spécification des prédicats ci dessus on peut trouver un invariant graphique.
Avec $place = TailleZoneFiltree(T, i, p)$.



5.2 Invariant formel

De l'invariant graphique on peut en dériver l'invariant formel :

$$\text{INV} \equiv N = N_0 \wedge 0 \leq i \leq N-1 \wedge place = TailleZoneFiltree(T, i, p) \wedge ZoneFiltree(T, p, place) \wedge \\ LienTableau(T_0, T, i, place) \wedge ZoneNonFiltree(T, i, place)$$