# Spécification et preuve de programmes
# Devoir 2 : spécifications et preuves avec micro-C

Alain Giorgetti

Licence d'Informatique de l'Université de Franche-Comté 2023-24

**Ce devoir doit être rendu sous la forme d'un unique fichier texte *NomPrenom*.c**, où ***NomPrenom*** sont les 8 premières lettres de votre nom de famille, suivies des 8 premières lettres de votre prénom, sans espaces ni accents. Ce fichier doit commencer par un commentaire entre `/*` et `*/` indiquant vos nom et prénom. Ensuite, des commentaires entre `/*` et `*/` doivent indiquer le numéro de chaque question traitée, par exemple

```
/* Exercice 2 question 3 */
```

juste avant la réponse à la question 3 de l'exercice 2.

La preuve et l'exécution du contenu complet de ce fichier doivent être possibles avec l'interface de micro-C en ligne. Dans ce but, les parties incorrectes ou incomplètes doivent être entourées par `/*` et `*/`.

**Lire l'énoncé jusqu'à la fin avant de commencer à traiter les exercices.**

## 1 Propriétés des tableaux d'entiers micro-C (5 points)

L'objectif de cet exercice est d'écrire des formules micro-C qui formalise des propriétés sur les tableaux d'entiers C.

1. Reproduire et compléter la définition

   ```
   predicate cte(int a[], int c) =
   ```

   pour définir en micro-C la propriété que tous les éléments du tableau `a` ont la même valeur `c`.

2. De même, définir en micro-C, par un prédicat `no_dup(int a[])`, la propriété que toutes les valeurs du tableau `a` sont deux à deux distinctes.

3. Utiliser ces deux prédicats pour formaliser en micro-C le lemme suivant : "Tout tableau constant qui a au moins deux éléments n'a pas tous ses éléments deux à deux distincts.".

4. Même si ce lemme est vrai et bien écrit en micro-C, vous pouvez observer que l'interface de micro-C ne permet pas de le démontrer, car le prouveur ne sait pas bien choisir des indices particuliers dans le tableau comme témoins. Pour aider le prouveur, formalisez les lemmes suivants :

   (a) "Dans tout tableau $t$ constant qui a au moins deux éléments, les cases $t[0]$ et $t[1]$ sont égales", et

(b) "Tout tableau $t$ qui a au moins deux éléments et dont les cases $t[0]$ et $t[1]$ sont égales a des éléments non deux à deux distincts."

5. Où doivent être placés ces lemmes pour aider la démonstration du lemme de la question 3 ?

## 2 Calcul du carré d'un nombre entier (4 points)

1. Définir une fonction C calculant le carré de son paramètre entier `x`, selon l'algorithme suivant :

$$i := 0;\ c := 0;\ \mathsf{while}\ i < x\ \mathsf{do}\ c := c + 2i + 1;\ i := i + 1\ \mathsf{od}$$

2. Spécifier le contrat de cette fonction en micro-C.
3. Spécifier un invariant de boucle assez précis pour permettre de démontrer la postcondition de ce contrat.
4. Spécifier un variant de boucle permettant de démontrer la terminaison de cette boucle.
5. Prouver le contrat avec l'interface en ligne de micro-C.

## 3 Calcul du maximum (4 points)

1. Définir en micro-C une fonction

   ```
   int max (int t[], int n)
   ```

   qui calcule l'indice d'un élément maximal dans tout tableau d'entiers `t` de longueur `n` non nulle.
2. Spécifier le contrat de cette fonction en micro-C.
3. Ajouter des invariants et un variant de boucle pour prouver ce contrat et la terminaison de la fonction.
4. Vérifier que la preuve est automatique avec l'interface en ligne de micro-C.

## 4 Recherche dichotomique en micro-C (7 points)

Cet exercice est la suite de l'exercice 3 du devoir 1.

1. Implémenter en C le programme de recherche dichotomique de la figure 2.4 du cours.
2. Spécifier le contrat de cette fonction en micro-C, en utilisant les prédicats définis dans le devoir 1.
3. Ajouter des invariants et un variant de boucle pour prouver ce contrat et la terminaison de la fonction.
4. Vérifier que la preuve est automatique avec l'interface en ligne de micro-C.