

INFO2009-Introduction à l'informatique

Aide à la réussite - Session n°4 - Les boucles et les invariants

Bertrand Alexis

Université de Liège

2023

Les différents types de boucle

Il y a 3 types de boucles en C : les `while`, les `do while`, et les `for`.

`while`

```
while(condition){...}
```

`do while`

```
do {...}  
while(condition);
```

`for`

```
for(init var; condition; opération fin de boucle){...}
```

Complexité

Nous utilisons la notation \mathcal{O} pour désigner la complexité en temps d'une fonction. Cette complexité correspond au nombre d'itération que devra faire la fonction dans le pire des cas, donc le nombre d'itération maximum.

Exemple

La complexité en temps d'une fonction qui doit faire n itération est donc $\mathcal{O}(n)$.

Triplet d'Hoare

Un triplet d'Hoare est une formule qui permet de raisonner formellement sur le fonctionnement d'un fragment de code. Un triplet d'Hoare correspond donc à la formule

$$\{\text{Précondition}\}\text{Fragment de code}\{\text{Postcondition}\}$$

La précondition est une formule logique qui doit être vraie avant d'exécuter le fragment de code, et la postcondition est une formule logique qui doit être vraie après avoir exécuté le fragment de code

Invariant de boucle

Un invariant de boucle est une formule logique qui doit être vrai avant d'entrer dans la boucle, à chaque itération de la boucle, et à la fin de la boucle. Lorsqu'on l'utilise avec le triplet d'Hoare, nous pouvons dégager les propriétés suivantes :

- ▶ L'invariant est une conséquence de la précondition du triplet. L'invariant est donc vrai chaque fois que la précondition l'est.
 $\text{Précondition} \implies \text{Invariant}$
- ▶ Si l'invariant est vrai avant une itération de la boucle, alors l'invariant l'est également après cette itération.
 $\{\text{Invariant} \wedge \text{Condition}\} \text{Itération} \{\text{Invariant}\}$
- ▶ En sortie de boucle, l'invariant implique la postcondition du triplet. L'invariant est donc vrai en sortie de boucle, comme la postcondition. $(\text{Invariant} \wedge \neg \text{Condition}) \implies \text{Postcondition}$

Prouver la terminaison d'un programme

Afin de prouver la terminaison d'un fragment de code, en plus de démontrer que son triplet d'Hoare est valide, il faut aussi démontrer la validité d'un variant de boucle, aussi appelé *fonction de terminaison*. Un variant de boucle doit satisfaire la propriété disant qu'à chaque itération complète de la boucle, on diminue sa valeur. Pour ce faire, on peut s'aider de l'invariant et montrer que le variant va diminuer tout en restant borné par l'invariant.

Exercices

Pour les exercices, nous allons retourner sur les exercices de la session précédente et calculer leur complexité et les prouver par la méthode des invariants.