

Boucles conditionnelles

Skander Zannad et Judicaël Courant

2013-09-21

1 Limites des boucles inconditionnelles

- On peut certes s'arrêter avant la fin (**break**).
- Mais il faut avoir une idée du nombre maximal d'itérations.

2 Calcul du pgcd

Algorithme d'Euclide pour calculer le plus grand diviseur de deux entiers relatifs a et b :

1. Remplacer a et b par leurs valeurs absolues.
2. Si $b = 0$, c'est terminé, le pgcd vaut a .
3. Sinon, noter r le résultat de la division de a par b .
4. Remplacer a par b et b par r et recommencer à la deuxième étape.

Problème : on n'a pas d'idée du nombre d'itérations nécessaires.

3 Boucle indéfinie/conditionnelle

```
def pgcd(a,b):  
    # précondition : (a, b) != (0, 0)  
    u = abs(a)  
    v = abs(b)  
    while v != 0:  
        # invariant : PGCD(a, b) == PGCD(u, v)  
        r = u % v  
        (u, v) = (v, r)  
    return u
```

4 Correction

4.1 Correction partielle

On justifie la correction (partielle) de la boucle par un invariant en début de boucle. Il convient de vérifier :

- Que l'invariant est vrai à l'entrée dans la boucle.
- Que s'il est vrai à la fin d'une itération, il est vrai au début de la suivante.

4.2 Terminaison

Pourquoi *partielle*? Parce qu'il se peut que la boucle ne s'arrête jamais. Reste à justifier la *terminaison* de la boucle.

On le fait avec un *variant* : une quantité entière positive ou nulle qui décroît à chaque tour de boucle.

```
while v != 0:  
    # invariant : PGCD(a, b) == PGCD(u, v)  
    # variant : v  
    r = u % v  
    (u, v) = (v, r)
```