

*Delphine Longuet, Thibaut Balabonski, Robin Pelle, Hadi Zaatiti*  
*longuet@lri.fr, blsk@lri.fr, pelle@lri.fr, hadizaatiti@gmail.com*

## TD 5 - Test structurel

Semaine du 14 novembre 2016

### Exercice 1

Soit la procédure Ada suivante qui renvoie deux entiers  $c$  et  $d$  comptant d'une part les éléments du tableau  $A$  qui sont inférieurs ou égaux à une constante  $m1$ , et d'autre part ceux qui sont supérieurs ou égaux à une constante  $m2$ . On suppose  $n \geq 0$ .

```
type tab = array(1..n) of natural;

procedure f(A: in tab; m1, m2, n: in natural; c, d: out natural) is
  i: natural;
begin
  c := 0;
  d := 0;
  i := 1;
  while (i <= n) loop
    if m1 >= A[i] then c := c + 1;
    else if m2 <= A[i] then d := d + 1; end if;
    end if;
    i := i + 1;
  end loop;
end f;
```

1. Donner le graphe de flot de données de cette procédure.
2. Donner une expression régulière caractérisant l'ensemble des chemins du graphe.
3. Donner les expressions régulières représentant les flots de données pour les variables  $c$ ,  $d$  et  $i$ . Ces flots de données comportent-ils des anomalies ?
4. Pour quelles valeurs de  $m1$  et  $m2$  existe-t-il des chemins infaisables ?
5. On considère le jeu de tests suivant :

Données d'entrée				Résultat attendu	
$m1$	$m2$	$n$	$A[1]$	$c$	$d$
12	25	1	5	1	0
12	25	1	37	0	1

Pour chacun des critères ci-dessous, indiquer s'il est satisfait par ce jeu de tests en justifiant votre réponse. Si ce n'est pas le cas, étendre le jeu de tests de manière à satisfaire ce critère.

- (a) Toutes les instructions ;

- (b) Toutes les définitions de  $c$ ,  $d$  et  $i$ ;
  - (c) Toutes les utilisations de  $c$ ,  $d$  et  $i$ .
6. Le jeu de tests obtenu à la question précédente vous semble-t-il adéquat ? Quel critère semble satisfaisant ? Quel autre critère de test structurel pourrait-on éventuellement chercher à satisfaire ?

### Exercice 2

On considère la fonction suivante qui calcule le plus grand commun diviseur (pgcd) de deux entiers positifs ou nuls  $m$  et  $n$ .

```
int pgcd(int m, int n) {
    int temp;
    while(n > 0) {
        if(n > m) {
            temp = n;
            n = m;
            m = temp;
        }
        temp = n;
        n = m-n;
        m = temp;
    }
    return m;
}
```

### Partie 1.

1. Donner le graphe de flot de données de ce programme.
2. Donner un chemin permettant de couvrir toutes les instructions. Déterminer par exécution symbolique la condition associée à ce chemin. Est-il faisable ? Si oui, donner un test concret pour ce chemin, sinon, trouver un autre chemin couvrant toutes les instructions qui soit faisable et donner un test concret pour ce chemin.
3. Dédurre de la question précédente une condition suffisante pour qu'un chemin soit infaisable.
4. Le test précédent permet-il de couvrir toutes les décisions ? Justifier.
5. Parmi les critères suivants sur le flot de données, déterminer ceux qui sont satisfaits par le test précédent. Pour les critères qui ne sont pas satisfaits, compléter par les chemins faisables nécessaires et donner des tests concrets pour ces nouveaux chemins. On utilisera l'exécution symbolique pour déterminer les conditions associées à ces chemins.
  - (a) Toutes les définitions de  $m$ , de  $n$  ;
  - (b) Toutes les utilisations de  $m$ , de  $n$ .
6. Le jeu de tests obtenu est-il satisfaisant ?

**Partie 2.** On considère maintenant l'implantation suivante de la même fonction.

```
int pgcd(int m, int n) {
    int mult = 1;
    while(n > 0) {
        if(n > m) {
            int temp = m;
            m = n;
            n = temp;
        }
        if(m%2 == 0) {
            if(n%2 == 0) {
                mult = mult*2;
                n = n/2;
            }
            m = m/2;
        } else {
            if(n%2 == 0) {
                n = n/2;
            } else {
                m = (m-n)/2;
            }
        }
    }
    return mult*m;
}
```

1. Donner le graphe de flot de contrôle de ce programme.
2. Quels sont les chemins couverts par le jeu de tests final de la partie 1 ?
3. Les critères sur le flot de contrôle « toutes les instructions » et « toutes les décisions » sont-ils satisfaits par ce jeu de tests ? Justifier. Ce jeu de tests est-il satisfaisant ?
4. Compléter le jeu de tests précédent par les chemins nécessaires pour satisfaire ces critères si besoin, et donner des tests concrets pour ces chemins en utilisant l'exécution symbolique.
5. Annoter le graphe de flot de contrôle par les informations du flot de données.
6. Les critères sur le flot de données « toutes les définitions de *mult* » et « toutes les utilisations de *mult* » sont-ils satisfaits ? Justifier. Compléter par les chemins nécessaires de manière à satisfaire ces critères et donner des tests concrets pour les nouveaux chemins.
7. L'ensemble de tests finalement obtenu est-il satisfaisant ?
8. Combien faudrait-il de chemins pour satisfaire le critère « tous les DU-chemins de *mult* » ?