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

TD 4 - Test fonctionnel et structurel

Semaine du 31 octobre 2016

Exercice 1

L'opération `middle` prend en entrée trois entiers différents deux à deux et renvoie l'entier parmi les trois qui n'est ni le plus grand ni le plus petit.

1. Donner la spécification de cette opération en termes de pré et post-conditions.
2. Construire la forme normale disjonctive de cette spécification et en déduire un ensemble de cas de test pour l'opération `middle`.
3. Donner des tests concrets pour chacun des cas trouvés à la question précédente.

Exercice 2

On considère le programme suivant, qui calcule X^N pour $N \geq 0$.

```
int puissance(int X, int N) {
    int S = 1;
    int P = N;
    while(P >= 1) {
        if(P mod 2 != 0) {
            P = P - 1;
            S = S * X;
        }
        S = S * S;
        P = P/2;
    }
    return S;
}
```

On veut générer des tests pour ce programme en utilisant les différents critères de couverture sur le graphe de flot de contrôle.

1. Écrire la spécification de ce programme sous forme de pré et post-conditions.
2. Construire le graphe de flot de contrôle de ce programme.
3. Sélectionner un ensemble de chemins pour satisfaire le critère « toutes les instructions ». Calculer par exécution symbolique les conditions de chemin associées à ces chemins et donner des tests concrets pour chacun des cas de test obtenus.
4. Compléter l'ensemble de chemins précédent pour satisfaire le critère « toutes les décisions ». Calculer les conditions de chemins associées. Parmi ces chemins, existe-t-il des chemins infaisables ? Si oui, lesquels et pourquoi ?
5. Compléter l'ensemble de chemins précédent pour satisfaire le critère « tous les chemins passant au plus deux fois dans la boucle ». Ces chemins sont-ils tous faisables ? Expliquer.

6. Pour les chemins faisables de la question précédente, calculer par exécution symbolique les conditions de chemin associées et donner des tests concrets pour chacun des cas de test obtenus.

Exercice 3

La fonction `days` calcule le nombre de jours écoulés entre deux dates d'une même année. Elle prend en entrée le jour et le mois de la première date, le jour et le mois de la deuxième date, et l'année considérée. On suppose que les jours et mois donnés en entrée forment des dates valides et que la deuxième date est postérieure à la première.

```
public static int days(int j1, int m1, int j2, int m2, int annee)
```

1. Donner un ensemble de tests fonctionnels pour cette fonction, en essayant au maximum de couvrir tous les cas de la spécification. (Rappel : on ne considère que les tests dont les données d'entrée satisfont les préconditions de la fonction.) Pour chaque test, donner son objectif, des données d'entrée concrètes ainsi que le résultat attendu du test.
2. On considère l'implantation suivante de cette fonction. Donner son graphe de flot de contrôle.

```
public static int days(int j1, int m1, int j2, int m2, int annee) {
    int daysin[] = new int[13];
    int res;

    daysin[1] = daysin[3] = daysin[5] = daysin[7] = 31;
    daysin[8] = daysin[10] = daysin[12] = 31;
    daysin[4] = daysin[6] = daysin[9] = daysin[11] = 30;
    if(m1 == m2) {
        res = j2 - j1;
    } else {
        if((annee%4 == 0) || (annee%100 == 0 && annee%400 != 0)) {
            daysin[2] = 29;
        } else {
            daysin[2] = 28;
        }
        res = j2 + (daysin[m1] - j1);
        for(int i = m1+1; i < m2; i++) {
            res = res + daysin[i];
        }
    }
    return res;
}
```

3. Pour chacun des critères de couverture suivants, dire s'il est satisfait par les tests donnés en question 1 et pourquoi. S'il n'est pas satisfait, donner un ensemble de chemins permettant de le satisfaire et des tests concrets pour ces chemins (on ne demande pas de faire l'exécution symbolique).
 - (a) Toutes les instructions;
 - (b) Toutes les décisions;
 - (c) Toutes les conditions multiples.
4. Reste-t-il des chemins non couverts qu'il serait pertinent de tester? Si oui, donner des tests concrets permettant de les couvrir.