

Test boîte blanche

Date : 5 avril 2013

Exercice 1

La fonction `compare_occ` prend en paramètres un tableau de caractères quelconque s , éventuellement vide, et deux caractères a et b . Elle renvoie *true* si et seulement si s contient strictement plus d'occurrences du caractère a que du caractère b .

1. De façon informelle, donner des objectifs de test et des cas de test correspondant (avec les sorties attendues) pour cette fonction.

On considère maintenant l'implantation suivante de cette fonction :

```
bool compare_occ(char[] s, char a, char b) {
    int i, e, n;
    bool res;
    i := 0;
    e := 0;
    n := s.length;
    while(i < n) {
        if (s[i] == a) then e := e + 1;
        if (s[i] == b) then e := e - 1;
        i := i+1;
    }
    res := (e > 0);
    return res;
}
```

2. Donner le graphe de contrôle de cette fonction.
3. Choisir un entier k tel que l'ensemble des chemins satisfaisant le critère « tous les chemins de longueur au plus k » contienne tous les chemins passant au plus deux fois par la boucle.
4. Choisir un chemin ne passant par la boucle, un chemin passant une fois par la boucle et un chemin passant deux fois par la boucle et calculer par exécution symbolique les conditions de chemin associées à ces chemins. Existe-t-il des chemins infaisables ?

5. De façon générale, parmi les chemins passant au plus deux fois par la boucle, existe-t-il des chemins faisables qui ne sont pas couverts par les tests construits en question 1? Donner les cas de test qui permettraient de les couvrir (et les sorties attendues dans chacun des cas).