**Partie 2**

Figure 1 : Capture d’écran de la méthode demandée

5

4

3

2

1

Text

Description automatically generated

13

14

12

11

100

9

8

7

6

Note :

Dans le bloc 4, à la ligne de secondPart, on a HamN + SpamN et non HamN \* SpamN. Cette erreur n’est pas présente dans le code.

Graphe 1 : Graphe CFG

Diagram

Description automatically generated

Tableau 1

|  |  |  |  |
| --- | --- | --- | --- |
| Nœud | DEF | C-USE | P-USE |
| user\_id | 1 | 1, 4 |  |
| groups | 1 | 4 | 2 |
| sum\_trust | 1, 3 | 3 |  |
| group | 2 | 3 | 2 |
| firstPart | 4 | 4 |  |
| secondPart | 4 | 4 |  |
| trust1 | 4 | 5 | 8 |
| trust2 | 4 | 5 | 6 |
| trust | 5, 7, 9,11, 13 | 14 | 10, 12 |

**All-Definition coverage** : au moins un def-clear path pour chaque nœud de définition

DC-PATH (user\_id, 1, 1) = {1}

DC-PATH (groups, 1, 2) = {1, 2}

DC-PATH (sum\_trust, 1, 3) = {1,2,3}

DC-PATH (group, 2, 3) = {2, 3}

DC-PATH (group, 2, 2) = {2}

DC-PATH (first\_part, 4,4) = {4}

DC-PATH (second\_part, 4,4) = {4}

DC-PATH (trust1, 4,5) = {4,5}

DC-PATH (trust1, 4,8) = {4, 5, 6, 8}

DC-PATH (trust2, 4,5) = {4,5}

DC-PATH (trust2, 4,8) = {4, 5, 6}

DC-PATH (trust, 5,14) = {5, 6, 8, 10, 12, 14}

DC-PATH (trust, 5,10) = {5, 6, 8, 10}

DC-PATH (trust, 5,12) = {5, 6, 8, 10, 12}

**Jeu de test:**

PathA = {1, 2, 3, 2, 4, 5, 6, 8, 10, 12, 14} couvre tous les DC-PATHS.

d1 = <user\_id, 1>, {return = 65}, on peut mock les fonctions pour retourner les valeurs que l’on retrouve dans l’image ci-dessous :

Text

Description automatically generated

Ce cas de test nous permet de passer par tous les noeuds du PathA.

**All C-USE coverage** : au moins un chemin def-clear à partir de chaque nœud de définition et vers chaque nœud de C-Utilisation.

DC-PATH (user\_id, 1,1) = {1}

DC-PATH (groups, 1,4) = {1, 2, 4}

DC-PATH (sum\_trust, 1,3) = {1,2,3}

DC-PATH (sum\_trust, 3, 3) = {3}

DC-PATH (group, 2,3) = {2,3}

DC-PATH (first\_part, 4, 4) = {4}

DC-PATH (second\_part, 4, 4) = {4}

DC-PATH (trust1, 4, 5) = {4, 5}

DC-PATH (trust2, 4, 5) = {4, 5}

DC-PATH (trust, 5, 14) = {5, 6,8,10,12,14}

DC-PATH (trust, 7, 14) = {7,8,10,12,14}

DC-PATH (trust, 9, 14) = {9,10,12,14}

DC-PATH (trust, 11, 14) = {11,12,14}

DC-PATH (trust, 13, 14) = {13,14}

**Jeu de test:**

PathA = {1, 2, 3, 2, 4, 5, 6, 7, 8, 9, 10, 12, 14} trust2 < 50, trust1 > 100

PathB = {1, 2, 3, 2, 4, 5, 6, 8, 10,11, 12, 14} trust > 100

PathC = {1, 2, 3, 2, 4, 5, 6, 8, 10, 12,13, 14} trust < 0

d1 = <user\_id = 1>, {return = 95} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

Text

Description automatically generated

d2 = <user\_id = 1>, {return = 100} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

A screenshot of a computer screen

Description automatically generated with medium confidence

d3 = <user\_id = 1>, {return = 0} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

Text

Description automatically generated

Ces trois tests nous permettent de parcourir les chemins A, B et C.

**All P-USE coverage** : au moins un chemin def-clear à partir de chaque nœud de définition et vers chaque nœud de P-Utilisation.

DC-PATH (groups, 1,2) = {1,2}

DC-PATH (group, 2,2) = {2}

DC-PATH (trust1, 4,8) = {4,5,6,8}

DC-PATH (trust2, 4,6) = {4,5,6}

DC-PATH (trust, 5,10) = {5,6,8,10}

DC-PATH (trust, 5,12) = {5,6,8,10, 12}

DC-PATH (trust, 7,10) = {7,8,10}

DC-PATH (trust, 7,12) = {7,8,10,12}

DC-PATH (trust, 9,10) = {9 ,10}

DC-PATH (trust, 9,12) = {9 ,10, 12}

DC-PATH (trust, 11,12) = {11 ,12}

Jeu de test:

PathA = {1,2,3,2,4,5,6,7,8,9,10,12, 14} trust2 < 50, trust1 > 100, trust = 0

PathB = {1,2,3,2,4,5,6,7,8,10,11,12, 14} trust2 = 0, trust1 < 100

d1 = <user\_id = 1>, {return = 95} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

Text

Description automatically generated

d2 = <user\_id = 1>, {return = 100} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

A screenshot of a computer screen

Description automatically generated with medium confidence

Ces deux tests permettent de parcourir les chemins A et B.

**All USE coverage** : au moins un chemin def-clear à partir de chaque nœud de définition et vers chaque nœud d’utilisation.

DC-PATH (user\_id, 1,1) = {1}

DC-PATH (user\_id, 1,4) = {1,2,3,4}

DC-PATH (groups, 1,2) = {1,2}

DC-PATH (groups, 1,4) = {1,2,3,4}

DC-PATH (sum\_trust, 1,3) = {1,2,3}

DC-PATH (sum\_trust, 3,3) = {3}

DC-PATH (group, 2,2) = {2}

DC-PATH (group, 2,3) = {2,3}

DC-PATH (first\_part, 4,4) = {4}

DC-PATH (second\_part, 4,4) = {4}

DC-PATH (trust1, 4,5) = {4,5}

DC-PATH (trust1, 4,8) = {4,5,6,8}

DC-PATH (trust2, 4,5) = {4,5}

DC-PATH (trust2, 4,6) = {4,5,6}

DC-PATH (trust, 5, 14) = {5, 6,8,10,12,14}

DC-PATH (trust, 7, 14) = {7,8,10,12,14}

DC-PATH (trust, 9, 14) = {9,10,12,14}

DC-PATH (trust, 11, 14) = {11,12,14}

DC-PATH (trust, 13, 14) = {13,14}

DC-PATH (trust, 5,10) = {5,6,8,10}

DC-PATH (trust, 5,12) = {5,6,8,10, 12}

DC-PATH (trust, 7,10) = {7,8,10}

DC-PATH (trust, 7,12) = {7,8,10,12}

DC-PATH (trust, 9,10) = {9 ,10}

DC-PATH (trust, 9,12) = {9 ,10, 12}

DC-PATH (trust, 11,12) = {11 ,12}

**Jeu de tests:**

PathA = {1, 2, 3, 2, 4, 5, 6, 7, 8, 9, 10, 12, 14} trust2 < 50, trust1 > 100

PathB = {1, 2, 3, 2, 4, 5, 6, 8, 10,11, 12, 14} trust > 100

PathC = {1, 2, 3, 2, 4, 5, 6, 8, 10, 12,13, 14} trust < 0

d1 = <user\_id = 1>, {return = 95} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

Text

Description automatically generated

d2 = <user\_id = 1>, {return = 100} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

A screenshot of a computer screen

Description automatically generated with medium confidence

d3 = <user\_id = 1>, {return = 0} lorsqu’on mock les méthodes pour qu’elles retournent les valeurs selon le code ci-dessous.

Text

Description automatically generated

Les critères ALL USE et ALL C-USE semblent être les plus strictes puisqu’ils parcourent tous les deux 3 chemins. Toutefois, puisque ALL USE comprend les P-USE ET LES C-USE, nous dirons que c’est le critère le plus stricte.