Text

Description automatically generated with medium confidence

LOG3430 -Méthodes de test et de validation du logiciel

TP1 – Tests unitaires

Groupe 1

Équipe 7

Hugo Lachieze-Rey (1934177)

Dimitry Kamga (1898357)

Remis à :

Hanane Ikhelef

Hiver 2022

Table des matières

[1. Tests unitaires 3](#_Toc94984678)

[a. Couverture avant les tests 3](#_Toc94984679)

[b. Couvertures après les tests 3](#_Toc94984680)

[c. Ajout de tests custom 3](#_Toc94984681)

[2. Tests de flots de données 4](#_Toc94984682)

[a. Ajout de la fonction compute\_user\_trust 4](#_Toc94984683)

[b. Graphe CFG 5](#_Toc94984684)

[c. Tableaux des nœuds 6](#_Toc94984685)

[d. Cas de tests 8](#_Toc94984686)

# Tests unitaires

## Couverture avant les tests

Table

Description automatically generated with medium confidence

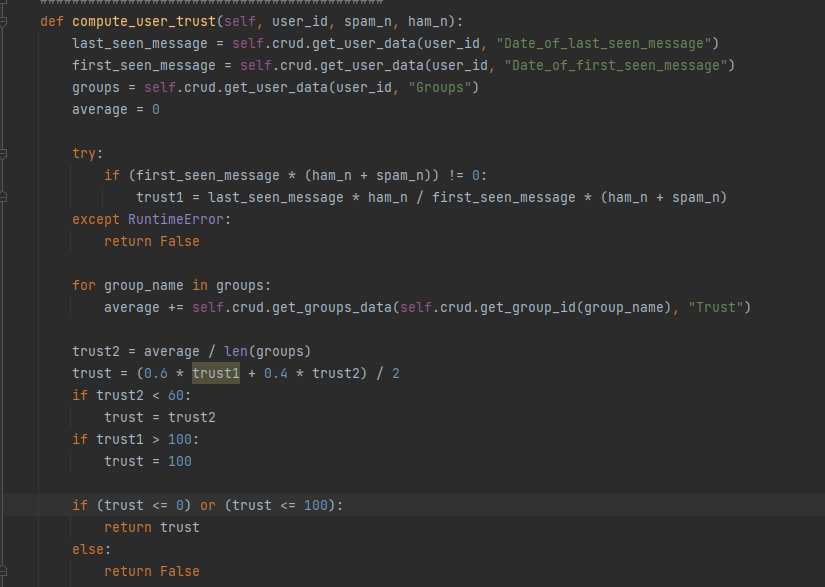
## Couvertures après les tests

## Ajout de tests custom

Nous avons ajouté des tests custom afin d’augmenter notre *coverage* pour chacun des fichiers demandés :

# Tests de flots de données

## Ajout de la fonction compute\_user\_trust



## Graphe CFG

Diagram, engineering drawing

Description automatically generated

## Tableaux des nœuds

|  |  |  |  |
| --- | --- | --- | --- |
| Nœud | DEF | C-USE | P-USE |
| user\_id | 1 | 1 |  |
| spam\_n | 1 | 2, 3 | 2 |
| ham\_n | 1 | 2, 3 | 2 |
| last\_seen\_message | 1 | 3 |  |
| first\_seen\_message | 1 | 2, 3 |  |
| trust1 | 3 | 6 | 9 |
| average | 1, 5 | 6 |  |
| groups | 1 | 6 | 4 |
| group\_name | 4 | 5 | 4 |
| trust2 | 6 | 6 | 7 |
| trust | 6, 8, 10 | 12 | 11 |

|  |  |  |  |
| --- | --- | --- | --- |
| Nœud | DEF | C-USE | P-USE |
| 1 | user\_id, spam\_n, ham\_n, groups, average, first\_seen\_message, last\_seen\_message | user\_id |  |
| 2 |  |  | first\_seen\_message, spam\_n, ham\_n |
| 3 | trust1 | spam\_n, ham\_n, first\_seen\_message, last\_seen\_message |  |
| 4 | group\_name | groups | group\_name |
| 5 | average | groupe\_name, average |  |
| 6 | trust2, trust | average, groups, trust1, trust2 |  |
| 7 |  |  | trust2 |
| 8 | trust | trust2 |  |
| 9 |  |  | trust1 |
| 10 | trust |  |  |
| 11 |  |  | trust |
| 12 |  | trust |  |

Trouvons tous les DU-PATHS (selon l’exemple du cours 02C).

DC-PATH(user\_id, n, m) à partir de tous les nœuds DEF(user\_id, n) avec zéro/une itération de boucle.

* Path1 = {1} avec C-USE(user\_id, 1)

DC-PATH(spam\_n, n, m) à partir de tous les nœuds DEF(spam\_n, n) avec une itération de boucle.

* Path2 = {1, 2} avec P-USE(spam\_n, 2)
* Path3 = {1, 2, 3} avec C-USE(spam\_n, 3)

DC-PATH(ham\_n, n, m) à partir de tous les nœuds DEF(ham\_n, n) avec une itération de boucle.

* Path4 = {1, 2, 3, 4} avec C-USE(groups, 4)
* Path5 = {1, 2, 3} avec C-USE(ham\_n, 3)

DC-PATH(groups, n, m) à partir de tous les nœuds DEF(groups, n) avec une itération de boucle.

* Path6 = {1, 2, 3, 4} avec P-USE(groups, 4)
* Path7 = {1, 2, 3, 4, 5, 6} avec C-USE(groups, 6)

DC-PATH(average, n, m) à partir de tous les nœuds DEF(average, n) avec une itération de boucle.

* Path8 = {1, 2, 3, 4, 5} avec C-USE(average, 5)
* Path9 = {5, 4, 6} avec C-USE(average, 6)

DC-PATH(first\_seen\_message, n, m) à partir de tous les nœuds DEF(first\_seen\_message, n) avec une itération de boucle.

* Path10 = {1, 2} avec P-USE(first\_seen\_message, 2)
* Path11 = {1, 2, 3} avec C-USE(first\_seen\_message, 3)

DC-PATH(last\_seen\_message, n, m) à partir de tous les nœuds DEF(last\_seen\_message, n) avec une itération de boucle.

* Path12 = {1, 2, 3} avec C-USE(last\_seen\_message, 3)

DC-PATH(trust1, n, m) à partir de tous les nœuds DEF(trust1, n) avec une itération de boucle.

* Path13 = {3, 4, 5, 4, 6} avec C-USE(trust1, 6)
* Path14 = {3, 4, 5, 4, 6, 7, 9} avec P-USE(trust1, 9) + {3, 4, 5, 4, 6, 7, 8, 9}

DC-PATH(group\_name, n, m) à partir de tous les nœuds DEF(group\_name, n) avec une itération de boucle.

* Path16 = {4, 5} avec C-USE(group\_name, 5)

DC-PATH(trust2, n, m) à partir de tous les nœuds DEF(trust2, n) avec une itération de boucle.

* Path17 = {6, 7} avec P-USE(trust2, 7)
* Path18 = {6, 7, 8} avec C-USE(trust2, 8)

DC-PATH(trust, n, m) à partir de tous les nœuds DEF(trust, n) avec une itération de boucle.

* Path19 = {6, 7, 9, 11} avec P-USE(trust, 11)
* Path20 = {6, 7, 9, 10} avec C-USE(trust, 10)
* Path21 = {8, 9, 11} avec P-USE(trust, 11)
* Path22 = {8, 9, 10} avec C-USE(trust, 10)
* Path23 = {10, 11} avec P-USE(trust, 11)
* Path24 = {10, 11, 12} avec C-USE(trust, 12)

## Cas de tests

**ALL-DEFinition coverage :** au moins un chemin *definition-clear* pour chaque nœud de définition.

PathA = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 13}

d1 = <{ len(groups) > 0, trust2 > 60, trust1 < 100, trust > 0, {false}}>

**ALL C-USE coverage :** au moins un chemin *definition-clear* partant de chaque nœud de définition à chaque utilisation-calcul.

PathA = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 13} pour le cas où trust2 > 60, trust1 < 100 et trust > 0.

PathB = {1, 2, 3, 4, 5, 4, 6, 7, 8, 9, 11, 13} pour le cas où trust2 > 60.

PathC = {1, 2, 3, 4, 5, 4, 6, 7, 9, 10, 11, 13} pour le cas où trust1 > 100.

PathD = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 12} pour le cas où 0 <= trust <= 100.

d1 = <{groups.length > 0, , trust2 > 60, trust1 < 100, 0 <= trust <= 100, {trust}}>

d2 = <{groups.length > 0, trust2 < 60, trust1 < 100, {false}}>

d3 = <{groups.length > 0, trust2 > 60, trust1 > 100, trust < 0, {false}}>

d4 = <{groups.length > 0, trust2 > 60, trust1 < 100, trust > 100, {false}}>

**ALL P-USE coverage :** au moins un chemin *definition-clear* partant de chaque nœud de définition à chaque utilisation-predicat.

PathA = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 13}

PathB = {1, 2, 3, 4, 5, 4, 6, 7, 8, 9, 11, 13}

PathC = {1, 2, 3, 4, 5, 4, 6, 7, 9, 10, 11, 13}

d1 = <{groups.length > 0, , trust2 > 60, trust1 < 100, 0 <= trust <= 100, {trust}}>

d2 = <{groups.length > 0, trust2 < 60, trust1 < 100, {false}}>

d3 = <{groups.length > 0, trust2 > 60, trust1 < 100, {false}}>

**ALL USE coverage :** tous les chemins *definition-clear*.

PathA = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 13}

PathB = {1, 2, 3, 4, 5, 4, 6, 7, 8, 9, 11, 13}

PathC = {1, 2, 3, 4, 5, 4, 6, 7, 9, 10, 11, 13}

PathD = {1, 2, 3, 4, 5, 4, 6, 7, 9, 11, 12}

d1 = <{groups.length > 0, , trust2 > 60, trust1 < 100, 0 <= trust <= 100, {trust}}>

d2 = <{groups.length > 0, trust2 < 60, trust1 < 100, {false}}>

d3 = <{groups.length > 0, trust2 > 60, trust1 > 100, trust < 0, {false}}>

d4 = <{groups.length > 0, trust2 > 60, trust1 < 100, trust > 100, {false}}>