

LOG3430 - MÉTHODES DE TEST ET DE VALIDATION DU LOGICIEL

LABORATOIRE 5

TESTS MÉTAMORPHIQUES

Département de génie informatique et de génie logiciel
École Polytechnique de Montréal



Hiver 2022

1 Introduction

Dans ce travail pratique vous allez effectuer les tests métamorphiques pour tester le système RENEGE.

2 Objectifs

L'objectif principal de ce laboratoire est :

1. Pratiquer la conception des tests métamorphiques.

3 Mise en contexte théorique

Une des techniques utilisée pour tester des programmes sans oracle consiste à utiliser un «pseudo-oracle» (Davis et Weyuker, 1981)¹, dans lequel plusieurs implémentations d'un algorithme traitent la même entrée et les résultats sont comparés. Cependant, ce n'est pas toujours possible, car il n'y a pas toujours plusieurs implémentations.

Même sans implémentations multiples, les applications présentent souvent des propriétés telles que si l'entrée est modifiée d'une certaine manière, il est possible de prédire la nouvelle sortie, étant donné la sortie d'origine. Cette approche est connue sous le nom de tests métamorphiques. Les tests métamorphiques ont été proposés par Chen et al (1998)².

L'idée est simple : même si nous ne connaissons pas la sortie correcte d'une seule entrée, nous pouvons connaître les relations entre les sorties correspondant aux différentes entrées. Nous pouvons vérifier le logiciel pour ces relations, appelées propriétés métamorphiques. Si elles ne tiennent pas, c'est un signe que le logiciel présente des défauts.

Tests métamorphiques

Dans ce travail pratique on va tester les propriétés métamorphiques pour le système RENEGE : la performance du système ne change pas

1. après le nettoyage des emails du "train dataset" ;
2. après le nettoyage des emails du "test dataset" ;
3. après 10 permutations des mots dans le "train dataset" ;
4. après 10 permutations des mots dans le "test dataset" ;
5. après avoir triplé les e-mails dans le "train dataset" ;
6. après avoir triplé les e-mails dans le "test dataset" ;
7. après avoir dupliqué les mots dans le "train dataset" ;
8. après avoir dupliqué les mots dans le "test dataset" ;

1. <https://dl.acm.org/doi/10.1145/800175.809889>

2. <https://arxiv.org/abs/2002.12543>

Remarques

1. Pour ce TP, téléchargez le zip Code TP1 sur Moodle.
2. Pour caractériser la performance de notre système, on va utiliser la métrique f1. Vous pouvez modifier le main pour calculer la métrique f1 à partir de la précision et du recall. Il serait aussi pertinent pour vos tests que vous retourniez la valeur de la métrique f1 dans la méthode evaluate.
3. Pour le nettoyage, vous pouvez utiliser la fonction `clean_text` disponible dans le fichier `text_cleaner`
4. Pour permuter l'ordre des mots, il faut tout d'abord les "tokenizer" (transformer dans une liste). Vous pouvez utiliser la fonction `split()` de Python. Vous devez ensuite faire la manoeuvre suivante 10 fois pour chaque email : sélectionner deux mots au hasard dans le body du email et les inverser. Pour "detokenizer" vous pouvez utiliser la méthode `.join()` de Python.
5. "Tripler les emails" cela veut dire que si vous avez 300 courriels, il faut tripler ces 300 courriels pour obtenir 900 courriels. Donc dans ces 900 courriels vous allez avoir 3 copies de chaque courriel. Dans le cas de train dataset vous allez avoir un dataset de 2100 courriels et pour test dataset - 900 courriels.
6. Pour la duplication des mots, vous devez prendre chaque mot dans le body et le rajouter à la fin. Un email de 20 mots en aura donc 40 après la modification.

Les tâches

La tâche principale est de, avec l'outil Unittest, créer un jeu de tests métamorphiques pour vérifier les propriétés métamorphiques de votre système.

1. Créer les fichiers json pour chaque transformation métamorphique réalisée. Plus précisément :
 - Pour nettoyage des e-mails : fichiers "train_clean.json" et "test_clean.json".
 - Pour changement d'ordre des mots : fichiers "train_shuffle.json" et "test_shuffle.json".
 - Pour le triplage des e-mails : fichiers "train700x3.json" et "test300x3.json".
 - Pour la duplication des mots : fichiers "train_words.json" et "test_words.json".
2. Créer 8 tests métamorphiques dans le fichier `test_main.py`. Chaque test doit correspondre à une chaque propriété c.a.d. il faut nommer les tests comme "test_nettoyage_train", "test_permutations_test", etc.
3. Fournir un tableau avec "f1" initial et "f1" après chaque test.
4. Vous allez utiliser les fichiers json avec transformations métamorphiques pour effectuer les tests. Pour que le test passe il faut que la différence entre le score f1 initial et le score f1 post-métamorphique ne dépasse 3%.
5. Si certains de vos tests ne passent pas, essayez de trouver les défauts dans votre système. Si vous ne réussissez pas, précisez quelles peuvent être les sources potentielles des bogues.
6. Pour les tests qui passent, expliquez quelles différences vous remarquez dans la performance de votre système.

7. Finalement, selon vous, quelles autres transformations métamorphiques peuvent être utilisées pour tester le système (deux exemples) ?

4 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire [4 points]. Dans le rapport il faut inclure le tableau des résultats et aussi votre analyse des résultats.
- Le dossier contenant tous les modules du RENEGE, test_main.py et les fichiers json avec transformations. [16 points].

Le tout est à remettre dans une seule archive **zip** avec pour nom matricule1_matricule2_matricule3_lab5.zip à téléverser sur Moodle.

Le rapport doit contenir le titre et numéro du laboratoire, les noms et matricules des coéquipiers ainsi que le numéro du groupe.

Consultez le site Moodle du cours pour la date et l'heure limites de remise des fichiers. Un retard de]0,24h] sera pénalisé de 10%, de]24h, 48h] de 20% et de plus de 48h de 50%.