

LOG3430 - MÉTHODES DE TEST ET DE VALIDATION DU LOGICIEL

LABORATOIRE 4

TESTS BASÉS SUR LES ÉTATS

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



Automne 2019

1 Mise en contexte théorique

L'approche des tests basés sur les états consiste à générer un arbre des transitions à partir du diagramme d'état d'un programme ou une composante sous test. Les chemins qu'on peut trouver dans cet arbre représentent les transitions et les noeuds représentent les états. Chaque transition dans l'arbre est un chemin qui part d'un état S et revient à ce même état. Le retour à l'état S marque la fin de la branche de l'arbre. Donc, il ne peut pas y avoir des répétitions de chemins dans l'arbre. Cette méthode de test nous permet de modéliser les transitions simples, qui ne contiennent pas de boucles, afin de générer les cas de tests correspondant. Pour plus de détails et d'exemples, voir les notes de cours sur les [Tests d'états](#).

2 Objectifs

- Représenter la classe *Stack.py* avec un diagramme d'états
- Générer l'arbre des transitions à partir d'un diagramme d'états.
- Identifier les cas de test et les implémenter à l'aide de unittest.

3 Mise en contexte pratique

Pour réaliser ce travail, il faut commencer par construire le diagramme d'états de la classe *Stack.py*.

Ensuite, il faut générer l'arbre des transitions correspondant en spécifiant les états et les conditions de garde pour chaque transition, s'il existe une.

Finalement, il faut identifier les cas de test à partir de l'arbre trouvé et implémenter les séquences trouvées avec unittest.

4 Travail à effectuer

1. Construire le diagramme d'états de la classe *Stack.py*.
2. Construire l'arbre des transitions de la classe *Stack.py*
3. Identifiez tous les cas de tests (avec les conditions de garde, s'il y en a) à partir de l'arbre trouvé.
4. À l'aide de unittest, écrire une classe de test unitaire pour tester les cas de test identifiés dans la question précédente.
5. À l'aide de l'outil Coverage.py, évaluez la couverture de la classe *Stack.py* selon la couverture des branches et identifiez les branches non couvertes, s'il y en a.

5 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire.
- Le dossier COMPLET contenant le projet.

Le tout à remettre dans une seule archive **zip** avec pour nom matricule1_matricule2_lab1.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%.