LOG3430

MÉTHODES DE TEST & DE VALIDATION DU LOGICIEL

Laboratoire 4

Tests basés sur les états

Sanghyuk Lee 1589242

François-Xavier Legault 1876882

Groupe 2

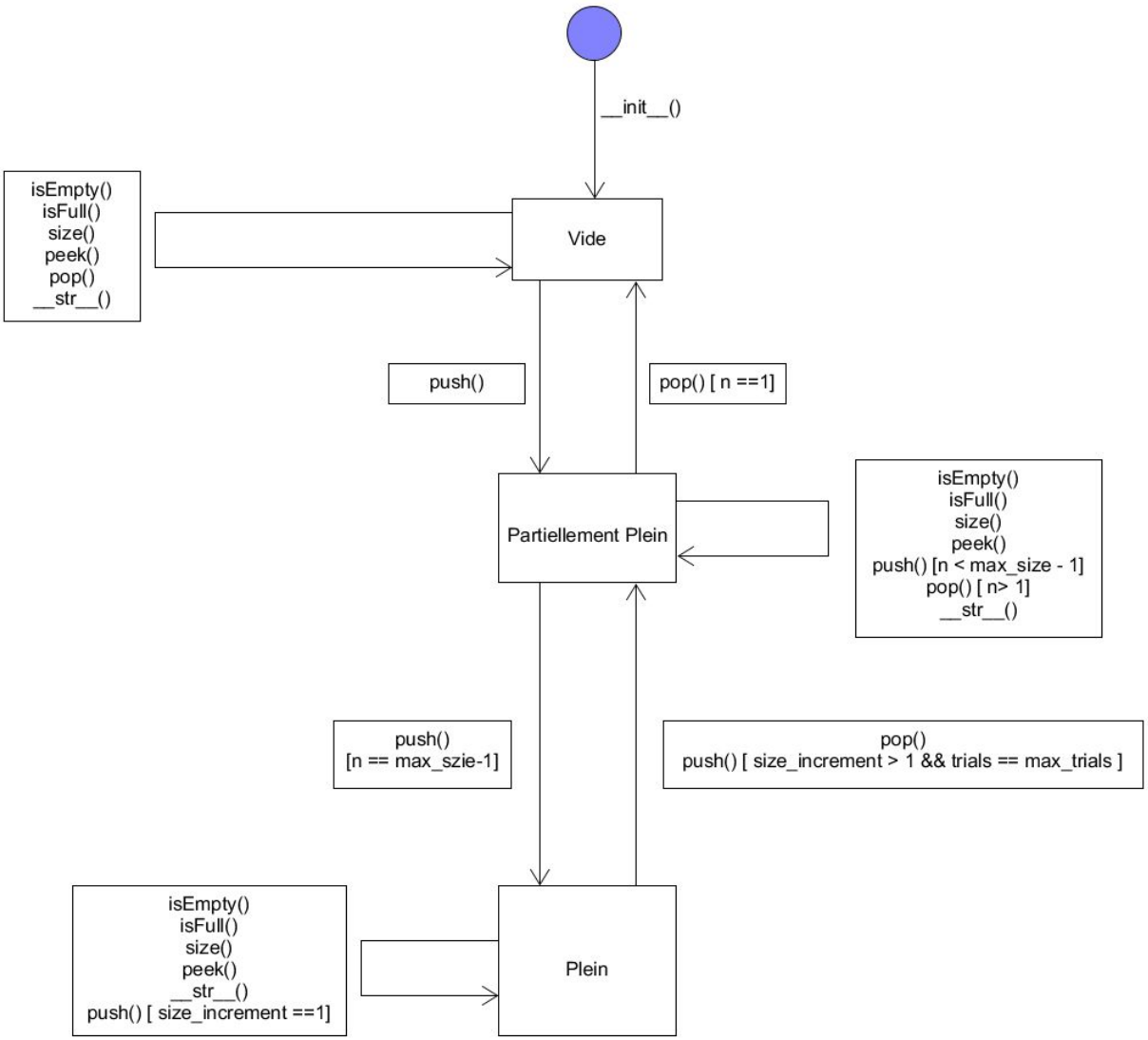
5 Novembre 2019

Introduction

L'objectif de ce travail pratique est de se familiariser avec les tests basés sur les états. Pour procéder, il faut générer un arbre de transitions en utilisant un diagramme d'état d'une composante sous test. Avec l'aide de cette méthode de test, on peut modéliser les transitions simples afin d'obtenir des cas de tests. Ici, la classe "Stack.py" sera notre programme qu'il faudra modéliser des tests.

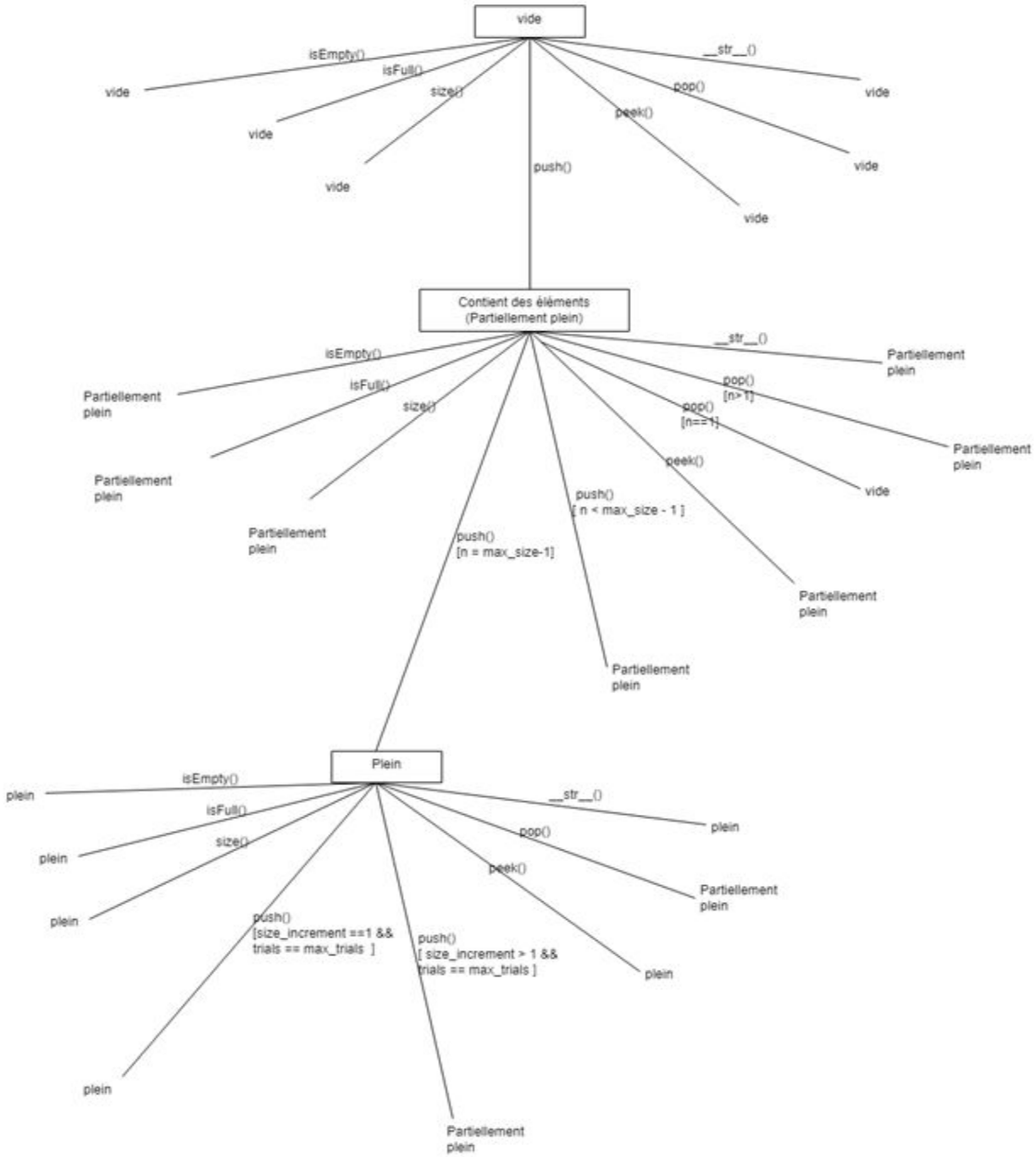
Diagramme d'états

Voici le diagramme d'états de la classe "Stack.py" que nous avons fait:



Arbre de transition

Voici l'arbre des transitions de la classe "Stack.py" que nous avons générer:



Nous avons ensuite identifié tous les cas de tests avec conditions de garde à partir de cet arbre trouvé. À l'aide d'unittest, nous avons écrit une classe de test unitaire pour tester ces cas de test que nous avons trouvé. Finalement, nous nous sommes assuré à l'aide de l'outil Coverage.py que la couverture de la classe "Stack.py" soit de 100%. Voici nos cas de tests:

- Les fonctions "pop" et "peek" créent une erreur lorsque la "stack" est vide.
- La fonction "push" sur une "stack" vide ajoute l'élément, puis "pop" l'enlève.
- La fonction "push" sur une "stack" jusqu'à ce qu'elle soit partiellement pleine, puis jusqu'à ce qu'elle soit pleine, et "pop" pour la vider et revenir sur nos pas.
- La fonction "push" sur une "stack" déjà pleine, puis avec l'incrémentation de taille.

À l'aide de l'outil "coverage", nous pouvons alors s'assurer que la couverture de nos tests ont atteint 100%:

```
[chris@localhost TP4]$ coverage run TestStack.py
....
----------------------------------------------------------------------
Ran 4 tests in 0.001s

OK
[chris@localhost TP4]$ coverage report
Name           Stmts   Miss  Cover
--------------------------------
Stack.py          51      0   100%
TestStack.py     147      0   100%
--------------------------------
TOTAL            198      0   100%
```

Conclusion

Nous nous sommes donc familiarisé lors de ce laboratoire avec les tests basés sur les états. Nous avons généré un diagramme montrant les états de la classe "Stack.py", puis nous avons modélisé les transitions simples afin d'obtenir un arbre de transitions, pour finalement obtenir des cas de tests que nous avons implémentés à l'aide de l'outil "unittest".