

Méthodes de test – Master 2 LC

TP 4 : Test unitaire avec JUnit et Coverlipse

Nous continuons le test unitaire des programmes Java en combinant JUnit avec un outil qui permet de calculer certaines couvertures de tests, Coverlipse. Nous appliquons ces outils au test d'un programme qui implémente en Java une API pour les automates finis (FSA). Ce programme, écrit par une équipe de Massachusetts University, a une taille assez importante (17K lignes de code).

1 Coverlipse

Coverlipse est un logiciel libre dont le site d'accueil est le <http://coverlipse.sourceforge.net/>. Il est installé comme plug-in Eclipse dans la salle 110.

Coverlipse calcule la couverture des noeuds (instructions) et *all-uses* pour les programmes Java. En combinaison avec JUnit, il permet d'évaluer la couverture des tests écrits. Les résultats de l'analyse peuvent être présentés soit directement dans l'éditeur Java d'Eclipse, soit exportés vers un fichier XML.

2 L'API LASER pour les FSA

LASER est un projet implémentant en Java une bibliothèque pour les FSA. Nous nous intéressons à tester que deux classes de ce projet, la classe `MutableFSA` et sa super classe `AbstractFSA`. Afin de tester ces classes, le code est structuré en deux projets Java :

1. Le projet `FSA` contient toutes le code et les archives Java (jar) nécessaires à ses paquetages.
2. Le projet `FSA Unit Test` contient trois fichiers et la documentation du paquetage `FSA`. Il est construit pour être compilé avec `FSA`. Les trois fichiers de `FSA Unit Test` sont :
 - `MutableFSA.java` et `AbstractFSA.java` dans le paquetage `laser.fsa` contiennent le code à tester.
 - `MutableFSATest.java` dans le paquetage `laser.fsa.test` à été généré automatiquement par le *wizard* JUnit d'Eclipse pour tester les méthodes des deux fichiers précédents. Il contient 18 méthodes

de test dont 17 vides et une complète (`testRemoveState`), ainsi que des champs privés et la méthode `setUp()` initialisant ces champs.

Les sources sont fournis dans une archive `fsa.zip` qu'il faudrait importer dans vos projets Eclipse.

3 Travail à réaliser

Vous devez compléter les méthodes de la classe `MutableFSATest.java`.

Pour cela, vous devez partie de la spécification Javadoc de l'implantation. La documentation utilise est celle des classes `MutableFSA` et `FSAInterface` (il n'y a pas de documentation pour la classe `AbstractFSA`). En cas de conflit avec le code, ne pas changer le code mais laisser JUnit déclarer une erreur.

Pour chaque méthode, écrire de tests afin de couvrir à 100% les instructions et au plus possible les utilisations. Pour cela, il est utile d'exécuter les méthodes de test en utilisant Coverlipse.