

Campagne de tests sur le logiciel logisim

Note:

Ce sujet sera étendu pour couvrir la partie JUnits.

Objectif

L'objectif du travail en mini-projet c'est de simuler une campagne de tests sur un logiciel de taille moyenne. Vous devez obtenir des jeux de test en appliquant plusieurs techniques de test. La présence de chacun des tests dans votre suite doit être le résultat de l'application d'une des techniques de test.

En plus d'identifier les jeux de test, vous devez créer une documentation de test, qui répertorie et justifie brièvement le contenu du matériel de test que vous allez développer.

Sujet

La campagne de test porte sur le logiciel logisim, développé (modifié) dans le cadre du projet programmation.

Chaque étudiant doit travailler sur le projet logisim développé par quelqu'un d'autre.

Etape I. Mise au propre de la spécification.

Créer une documentation succincte qui énumère les exigences et les contraintes du logiciel. N'oubliez pas les propriétés, données dans le cours 2, pour les exigences (atomicité, non-ambiguïté, concision, testabilité, etc). Est-ce que par rapport à ce logiciel on peut définir des exigences non-fonctionnelles?

Etape II. Test unitaire

Effectuez des tests unitaires pour les fonctions correspondant aux portes MOD₃ et ADD, sur la fonction qui assure le lien avec OCAML, ainsi que sur les fonctions concernés par la fonctionnalité F₂.

Attention on s'intéresse uniquement à la fonction à partir de laquelle on appelle l'Ocaml, l'appel lui-même, ainsi que la partie Ocaml, ne font pas objet de notre campagne de test.

En utilisant Emma assurez vous que la couverture obtenue est complète.

Justifiez le choix de chacun de ces tests unitaires.

Indice: Les tests unitaires peuvent demander la création de stubs ou de lanceurs.

Etape III. Test d'intégration

Effectuez des tests d'intégration pour les fonctions concernées par la fonctionnalité F2.

Etape IV. Test de validation et test fonctionnel.

En utilisant la méthode des partitions donner des jeux de test pour les fonctionnalités du logiciel visé par l'étape II (on s'intéresse uniquement aux fonctionnalités qui ont fait l'objet de développement ou modification dans le cadre du projet de programmation).

On vise une couverture faible des partitions.

Effectuez des tests de robustesse, en utilisant l'hypothèse qu'il peut y avoir une seule faute (single fault assumption).

Fournitures

Le travail pour le mini-projet est individuel (pas de groupe autorisés).

A la fin du mini-projet (la date limite sera) vous devez livrer sous moodle une archive (zip) contenant

- La documentation de test (en format .pdf)
- Un répertoire contenant les jeux de test que vous avez obtenus (un fichier / jeu de test), accompagné d'un fichier read.me contenant une brève description du format attendu pour les données de ce fichier
- Un répertoire contenant le(s) script(s) de test qui exécutent les jeux de test que vous avez obtenus. Les jeux de test doivent être lus à partir des fichiers (non pas codés en dur). Chaque script doit calculer des statistiques de réussite / échec sur les tests exécutés.
- Tout autre matériel de test que vous avez développé.