HEIG VD HAUTE ÉCOLE D'INGÉNIERIE ET DE GESTION DU CANTON DE VAUD

Prof. Yann Thoma

Exercices du cours VSE
Exercices de vérification SystemVerilog
Randomisation

semestre automne 2024 - 2025

Exercice

Reprenez le code proposé (sous git), et complétez la classe `RTest` de manière à disposer des champs suivants :

1. `a` qui est un vecteur sur 16 bits
2. `b` qui est un vecteur sur 16 bits
3. `c` qui est un vecteur sur 16 bits
4. `m` qui est un vecteur sur 2 bits

Ajoutez de quoi créer un objet de cette classe et générer 1000 randomisations différentes.
Ajoutez maintenant les contraintes suivantes :

1. `m` doit être compris dans l'intervalle $[0, 2]$
2. Si `m` vaut 0, alors a doit être plus petit que 10
3. Si `m` vaut 1, alors b doit être compris dans l'intervalle $[12, 16]$
4. En tout temps, `c` doit être plus grand que `a + b`

Afin de valider vos contraintes, ajoutez du code dans votre boucle permettant de vérifier que vos contraintes sont bien respectées.
Créez maintenant une autre classe `STest` avec les champs suivants :

1. `sa` qui est un vecteur sur 8 bits
2. `sb` qui est un vecteur sur 8 bits

Et la contrainte suivante :

1. Si `sa` est pair, alors `sb` doit aussi être pair

Ajoutez à la classe `RTest` une variable membre de type `STest` et faites qu'elle soit randomisée.
Modifiez votre boucle de façon à vérifier les contraintes du membre `STest`.
Ajoutez à la classe `RTest` une variable membre étant un tableau de `STest`, et faites qu'il soit randomisé. A vous de choisir une taille maximale.
Modifiez votre boucle de façon à vérifier les contraintes des éléments du tableau.
Ajoutez, dans une de vos classes, des champs et une (ou des) contrainte offrant un biais significatif, en exploitant une structure du type a implique b. Mettez en place de quoi quantifier ce biais, et comparez ce que vous observez avec ce que vous auriez prédit si le solver ne fonctionnait qu'en générant de l'aléatoire jusqu'à ce que les contraintes soient satisfaites.