

M1 Info – ARC - LAB4

Olivier HUREAU - Groupe 3

25/03/2020

Table des matières

| | | |
|----------|---|----------|
| 1 | Évaluation de la qualité des tests. | 2 |
| 1.1 | Première analyse | 2 |
| 1.1.1 | Résultats du test de coverage | 2 |
| 1.1.1.1 | Résultats du test de coverage sur le système entier | 2 |
| 1.1.1.2 | Résultats du test de coverage sur le robot | 3 |
| 1.1.1.3 | Interprétation des résultats | 3 |
| 1.2 | Correction du code après analyse | 3 |
| 1.3 | Deuxième analyse | 4 |
| 2 | Vérification d'assertions temporelles | 4 |

1 Évaluation de la qualité des tests.

Les différents types de couverture nous intéressant sont

- State / Transition coverage : En principe les testbench vérifie que la machine passe dans tout les états. Si nous n'avons pas 100% de couverture ici c'est qu'il y a un problème.
- Branch coverage : Si le teste de couverture remonte des erreurs, il sera alors plus facile de traquer les erreurs de transissions..
- Expression coverage : Les expressions influant sur les branchements, ce test de couverture nous permet aussi de vérifier le comportement du robot.

Les tests "line coverage" et "statements coverage" donnent une indication sur la présentation ou l'optimisation du code, ne nous souciant uniquement du bon fonctionnement de notre implémentation, il ne nous sont pas utiles.

Le test Toggle coverage donnant une indication sur la robustesse du l'implémentation. Il est bien trop compliqué et trop long d'analyser les résultats de ce test dans le cadre de nos labs.

1.1 Première analyse

1.1.1 Résultats du test de coverage

1.1.1.1 Résultats du test de coverage sur le système entier Ci dessous : S est l'architecture System, C1 et C2 les compteurs et R le robots dans le testbench System.

| Instance | Branch % | State % | State graph | Transition % | Transition graph |
|------------|----------|---------|------------------------|--------------|------------------------|
| testsystem | 74.7% | 92.3% | <div><div></div></div> | 64.1% | <div><div></div></div> |
| S | 74.7% | 92.3% | <div><div></div></div> | 64.1% | <div><div></div></div> |
| C1 | 95% | 100% | <div><div></div></div> | 100% | <div><div></div></div> |
| C2 | 90% | 100% | <div><div></div></div> | 100% | <div><div></div></div> |
| R | 61.8% | 88.9% | <div><div></div></div> | 54.8% | <div><div></div></div> |

Figure 1: Résultats de couverture Branch, State et Transition sur le système entier

Le graph de couverture du robot dans le testbench system

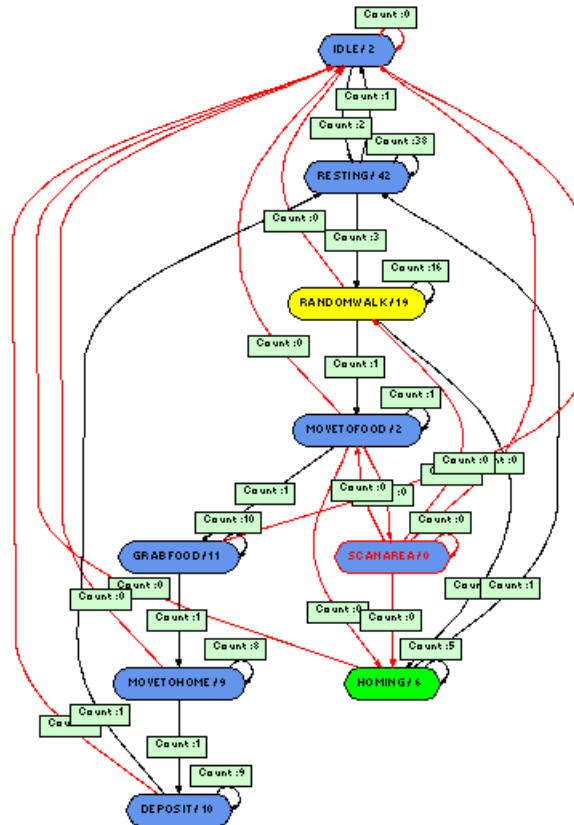


Figure 2: Graph de couverture d'état sur le système entier

1.1.1.2 Résultats du test de coverage sur le robot Ci dessous : S est l'architecture System, C1 et C2 les compteurs et R le robots dans le testbench System.

| Instance | Branch % | State % | State graph | Transition % | Transition graph |
|-----------|----------|---------|---|--------------|--|
| testrobot | 76.4% | 100% | <div style="width: 100%; height: 10px; background-color: green;"></div> | 90.3% | <div style="width: 90.3%; height: 10px; background-color: green;"></div> |
| A | 76.4% | 100% | <div style="width: 100%; height: 10px; background-color: green;"></div> | 90.3% | <div style="width: 90.3%; height: 10px; background-color: green;"></div> |

Figure 3: Résultats de couverture Branch, State et Transition sur le robot

Le graph de couverture du robot dans le testbench system

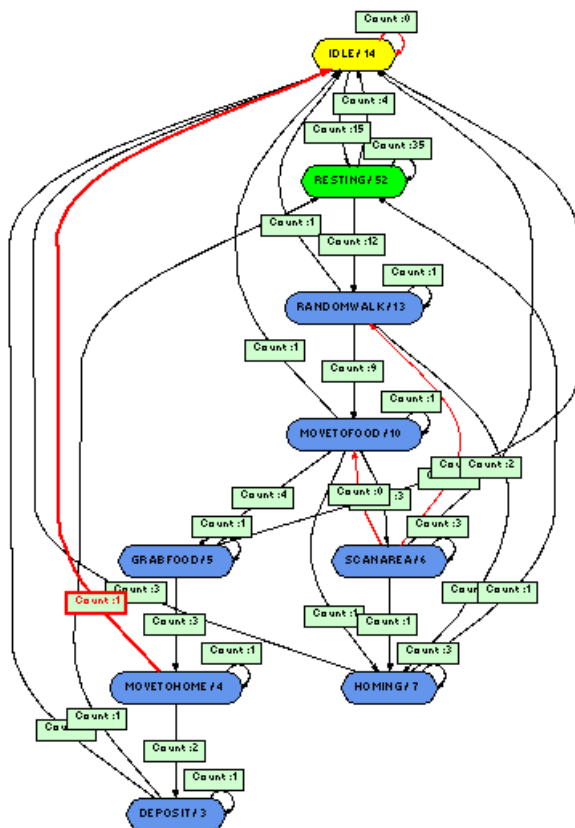


Figure 4: Graph de couverture d'état sur le robot

1.1.1.3 Interprétation des résultats

Les tests de couverture sont très probant (87% pour le système entier). Cependant, on voit que dans le système entier, l'état SCANAREA n'est jamais atteint bien que les tests de système (se reporter au tableau du précédent lab) aient prévu un chemin pour arriver jusqu'à cet état ainsi que d'utiliser ses transitions.

L'analyse du robot avec un testbench différent nous donne alors des indications suivantes : certaines transitions ne sont jamais atteintes, l'état SCANAREA est bien accessible pour l'automate.

Deux hypothèses se posent alors : Soit le système est mal conçu, soit c'est les testbenches. Avant de se replonger dans une éventuelle réécriture des architectures et/ou entité (la couverture des branch pourra nous aider pour cela). Il faut être certains que c'est les testbenches qui ne sont pas faux.

1.2 Correction du code après analyse

Après interprétation des résultats, on crée des séquences de signaux qui nous permettront d'être sûr que le testbenches du robot n'est pas faux.

Grace au graph de couverture (figure 4) il faut tester les transitions suivantes :

- Reset depuis MOVETOHOME
- Aller de SCANAREA à MOVETOFOOD (rappel de contraintes de transition : Abovesearch = '0' & findfood = '0' & scantimeup = '1')
- Aller de SCANAREA à RANDOMWALK (rappel de contraintes de transition : Abovesearch = '0' & findfood = '1')

1.3 Deuxième analyse

2 Vérification d'assertions temporelles