

Méthodes de test basées sur les automates à états finis: Etat de l'art et perspectives.

Z. Lotfi,, F. Ouardi

Résumé— La génération de cas de test est une partie essentielle de l'évaluation des logiciels. La littérature contient différentes méthodes pour développer une suite de tests modélisés par des automates à états finis (AEF). Dans cet article, nous présentons un aperçu concis et compact des méthodes les plus utiles lorsque la spécification est donnée par un déterministe automate à états finis. Nous réalisons également une synthèse où on montre les points forts et limites de ces méthodes. Nous proposons aussi des voies possibles qui sont dans la continuité des travaux précédents

Mots-clés— Test basé sur les modèles, Test de conformité, Séquence de Distinction.

1 INTRODUCTION

LE principe du test de conformité est de déterminer si l'implémentation et la spécification sont équivalentes. Nous présentons ici les méthodes qui permettent de répondre à cette question: Tour de transitions [1], séquence de distinction DS-Method [2], séquence unique d'entrée/sorties UIO-method [3] et enfin séquence de caractérisation W-method [4].

2 METHODES DE TEST BASEES SUR LES AUTOMATES A ETATS FINIS

Tour de transitions (Méthode T) Elle s'applique sur une spécification minimale et fortement connexe. La méthode T consiste à générer une séquence de transition en utilisant des entrées aléatoires jusqu'à ce que la machine ait traversé chaque transition au moins une fois. Le point faible de cette méthode c'est que les fautes de transfert ne peuvent pas être détectées. Le meilleur moyen de générer une tour de transition d'AEF consiste à utiliser l'algorithme du facteur chinois. La complexité de cette méthode est polynomiale.

Séquence de Distinction (Méthode DS) La séquence de distinction (DS) est une séquence d'entrée qui distingue deux états quelconques dans AEF en fonction de la sortie observée. Elle peut être représentée sous forme adaptative (ADS) ou prédéfinie (PDS). ADS est un arbre où chaque chemin de racine à feuille représente une séquence d'entrée spécifique à l'état représenté par la feuille. Ainsi, pour chaque état, l'entrée suivante dépend de la sortie de l'entrée actuelle. Par contre PDS, la séquence est fixée d'avance et sa séquence de sortie est différente pour chaque état initial, PDS est une séquence d'entrée unique pouvant être utilisée pour distinguer chaque état des autres. Il y a deux inconvénients majeurs; Le premier est que, dans la pratique, très peu des AEF possèdent en réalité une DS. Deuxièmement, même si un AEF a une DS, la limite supérieure de la longueur de la DS sera trop grande pour être utile en général.

Méthode UIO (Unique input output) Une séquence de vérifica-

tion d'état d'un état $s \in S$ est une séquence d'entrée $x \in I^*$, telle que la séquence de sortie produite par la machine en réponse à x à partir de tout autre état est différente de celle de s . Pour vérifier un état par une séquence UIO, on peut appliquer la séquence d'entrée d'UIO et comparer la séquence de sortie à celle attendue. S'ils sont identiques, l'AEF est dans l'état s . Sinon, il n'est pas dans l'état s . La complexité du calcul des UIO ainsi que leur taille sont exponentielles [5].

Séquences caractéristiques (Méthode W) Une séquence caractéristique est constituée de séquences d'entrée permettant de distinguer les comportements de chaque paire d'états. Les techniques de test basées sur un ensemble de caractérisations nécessitent l'exécution d'un certain nombre de séquences d'entrée pour chaque état, et peuvent donc également prendre un temps de test long. La complexité de la construction de cette séquences est en $O(n^2)$ [4].

3 DISCUSSION GENERALE ET FUTURES ORIENTATIONS

Les méthodes basées sur les automates ont certains avantages et certains inconvénients. La spécification doit être dans beaucoup de cas fortement connexe et complète, et de plus l'IUT doit être minimal. Les algorithmes connus de ces méthodes sont souvent polynomiaux en temps et en mémoire, et ceci réduit d'autant plus la taille des spécifications en entrée. Il existe donc plusieurs pistes de travail surtout pour le cas des AEF non déterministes. Pour ADS, il reste à déterminer si son problème d'existence est décidable et, si le DS existe [6], il est important de trouver des conditions réalistes dans lesquelles les problèmes du SDA peuvent être résolus en temps polynomial. Autre direction pour ADS, il serait intéressant de trouver la limite supérieure étroite en ce qui concerne la longueur des tests élémentaires distincts pour les AEF observables et non observables [7]. En plus, il est peut être possible de développer des méthodes heuristiques permettant de dériver des PDS contrôlables à partir de AEF et de réduire la complexité temporelle de la vérification de l'existence de PDS. Il est également possible de développer de nouveaux algorithmes de génération de tests automatisés lorsque les tests sont distribués [8]. Finalement,

nous remarquons également qu'il a un défi de rechercher de près des algorithmes parallèles pour générer certains types de séquences telles que les séquences UIO, les ensembles de caractérisation et DS pour les AEF partiels non observables et d'évaluer l'efficacité de chaque algorithme [9].

4 CONCLUSION

Dans cet article, nous avons présenté une étude compacte et concise des méthodes de test basées sur les automates à états finis. Nous avons réalisé une synthèse sur les points forts et faibles de ces méthodes et donné des futures orientations de recherche.

REFERENCES

- [1] S. Naito and M. Tsunoyama. Fault detection for sequential machines by transitions tours. IEEE Fault Tolerant Comput. Symp, IEEE Computer Soc. Press, pages 238-243, 1981.
- [2] G.Gonenc. A method for the design of fault detection experiment. IEEE transactions on Computers, C-19 :551-558,1970
- [3] D. Lee and M. Yannakakis. Principles and methods of testing finite state machines- a survey. Proceedings of the IEEE, pages 84(8) :1090-1123, August 1996.
- [4] T. Chow. Testing software design modelled by finite state machines. IEEE Transactions on Software Engineering,, SE-4(3), May 1978.
- [5] D. Lee and M. Yannakakis. Testing finite state machines : State identification and verification. IEEE Trans Computer, pages 43(3) :306-320, 1994.
- [6] R. Hierons and U. Turker, Distinguishing Sequences for Distributed Testing: Adaptive Distinguishing Sequences, The Computer Journal, vol. 59, no. 8, pp. 1186-1206, 2016.
- [7] Yenign, H., Yevtushenko, N., and Kushik, N., The complexity of checking the existence and derivation of adaptive synchronizing experiments for deterministic FSMs, Inf. Process. Lett., 2017, vol. 127, pp. 4953.
- [8] Hierons, R.M. and Turker, U.C. (2017) Parallel algorithms for generating distinguishing sequences for observable nondeterministic FSMs. ACM Trans. Softw. Eng. Methodol., 26, 5: 15:34.
- [9] Kushik, N., El-Fakih, K., Yevtushenko, N., Cavalli, A.R.: On adaptive experiments for nondeterministic finite state machines. Softw. Tools Technol. Transf. (2014). doi: 10.1007/s10009-014-0357-7