Construire des abstractions par preuve pour générer des tests

P.-C. Bué, J. Julliand, P.-A. Masson
LIFC - Université de Franche-Comté - 16, route de Gray F-25030 Besançon Cedex, France
Email: {bue, julliand, masson}@lifc.univ-fcomte.fr

I. Motivations

Dans une approche de test à partir de modèles (*Model-Based Testing*) [1], [2], [3], un modèle est rédigé et validé séparément d'une implantation, à partir d'un document initial de spécification. Des tests sont calculés à partir du modèle, en utilisant des critères de couverture de celui-ci. En confrontant l'implantation aux tests issus du modèle, on peut détecter des non conformités de l'implantation par rapport au modèle.

Mais la mise en œuvre de ces techniques de test reste difficile pour des systèmes de taille industrielle, en raison de la très grande taille de l'espace d'états de leurs modèles. On peut alors calculer et utiliser des abstractions de ces modèles pour faire face à ce problème. Notre intention est de réduire la taille du modèle à partir duquel sont générés les tests. L'objectif est de maîtriser l'explosion combinatoire du nombre d'états, de transitions ou de chemins qui sont les critères de couverture de test habituels utilisés sur ce type de modèle.

Mais abstraire un modèle initial pose deux problèmes, son coût de calcul et sa relation, en termes de correction et de précision, avec le modèle initial et le critère d'évaluation. Deux sortes d'abstractions sont envisageables, soit une sur-approximation, soit une sous-approximation au sens où l'ensemble des éxécutions de l'abstraction est respectivement, soit un sur-ensemble, soit un sous-ensemble des exécutions du modèle initial. Une sur-approximation est adaptée à la vérification des propriétés de sûreté qui sont préservées : si "quelque-chose de mauvais n'arrive pas sur un plus grand nombre d'exécutions, il n'arrivera pas sur le modèle initial". Par contre, la génération de tests à partir d'une sur-approximation nécessite de vérifier que chaque test issu de l'abstraction est bien une exécution du système initial. Au contraire, les tests issus d'une sous-approximation sont tous exécutables sur le modèle initial. Mais se pose alors un problème de précision : le nombre de tests issus d'une sous-approximation couvre-t-il de manière suffisante le système ? Finalement, pour la génération de tests, les deux approches sont possibles, mais le calcul de sur-approximation nécessite une phase de filtrage des tests réellement exécutables alors que le calcul de sous-approximation peut engendrer un nombre insuffisant de tests dû à une trop faible précision.

Ce résumé aborde cette problématique en décrivant une méthode de calcul de sur et de sous-approximations. Le problème à résoudre est de trouver le meilleur compromis entre la qualité de la couverture de test et les performances de calcul des tests. Les techniques de sur-approximation ont un coût en temps de calcul important dû à la phase de filtrage par instanciation des tests qui consiste à réaliser une recherche combinatoire. Le taux de couverture dépend de la réussite de cette phase d'instanciation. Par contre, les méthodes par sous-approximation par concrétisation sont plus efficaces puisqu'elles ne nécessitent pas de phase de filtrage et d'instanciation. Mais le taux de couverture dépend de la capacité à engendrer des graphes d'états concrets connexes.

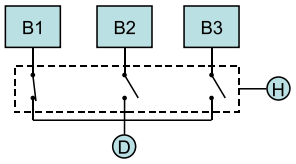

Fig. 1. Schéma du Système Electrique

Nous présentons un algorithme de calcul d'abstraction qui prend en entrée, outre le modèle à abstraire, un ensemble de n prédicats d'abstraction [4]. Il calcule une relation de transitions symboliques sur-approximée Δ entre les 2^n états symboliques en évaluant des conditions établissant l'existence de transitions entre ces états. Il calcule une sous-approximation Δ^c en instanciant ces transitions à la volée en s'inspirant des travaux de [5]. L'ensemble des exécutions concrètes partant des états initiaux constituent la sous-approximation. Cet algorithme repose sur l'utilisation de trois fonctions primitives de preuve : un calcul de plus faible précondition, de plus forte postcondition et une évaluation de satisfiabilité (par un solveur SAT). Nous avons implanté notre méthode en utilisant des solvers SAT SMT, en l'occurrence Z3 [6].

II. Exemple

Nous présentons un exemple pour illustrer nos propos. C'est un système réactif de contrôle de l'alimentation électrique d'un dispositif D alimenté par l'une des trois batteries B_1, B_2, B_3 comme le montre la figure 1. Un interrupteur connecte (ou non) une batterie B_i à l'appareil D. Une horloge H envoie périodiquement un signal de commutation, c'est à dire un ordre de changement de la batterie alimentant D. Le fonctionnement du système doit satisfaire les trois exigences suivantes:

- Req_1 : il n'y a pas de court-circuit,
- Req_2 : l'appareil est constamment alimenté,
- Req_3 : lorsque l'horloge envoie une commande de commutation, l'interrupteur fermé est modifié.

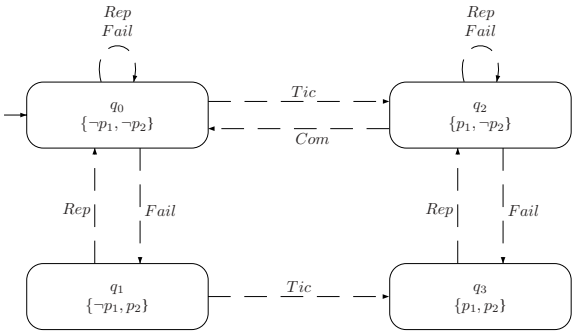

Fig. 2. Relation Δ du système électrique

Mais les batteries peuvent tomber en panne. Quand la batterie qui alimente D tombe en panne, le système effectue une commutation exceptionnelle pour satisfaire l'exigence Req_2. Les batteries en panne sont remplacées par un service de maintenance. Nous supposons que ce service travaille suffisamment rapidement pour qu'il n'y ait jamais trois batteries en panne simultanément. Quand deux batteries sont en panne, l'exigence Req_3 est relâchée et les ordres de commutation de l'horloge ne sont plus pris en compte.

Par exemple, ce processus d'abstraction donne la relation de transition Δ representée dans Fig. 2.

III. Application du calcul d'abstractions à la génération de tests

Notre intention est d'utiliser les abstractions produites selon l' algorithme présenté précédemment pour la génération de tests. Nous présentons deux processus de génération de tests illustrés dans la figure 3, le premier $\mathcal{P}_{sur}$ utilise la sur-approximation Δ et le second $\mathcal{P}_{sous}$ la sous-approximation Δ^c.

La figure 3 présente les grandes lignes de ces processus. Les deux premières étapes sont communes aux deux processus. L'abstraction AM est synchronisée avec un objectif de test OT pour cibler les exécutions décrites par l'objectif. Les tests sont extraits du produit synchronisé PS à partir d'un critère de sélection structurel comme la couverture des transitions et/ou des états. Si l'abstraction était la sous-approximation, le processus $\mathcal{P}_{sous}$ est terminé car les tests TI sont définis au niveau du modèle M qui a été abstrait. Si l'abstraction était la sur-approximation, le processus $\mathcal{P}_{sur}$ se poursuit par l'instanciation des tests TAS qui sont transformés du niveau de l'abstraction AM au niveau du modèle M.

Sur l'exemple, l'objectif de test est d'observer les comportements lorsque la batterie qui alimente le système tombe en panne et lorsqu'il n'y a plus qu'une seule batterie en fonctionnement. On extrait les deux prédicats suivants de cet objectif de test : p_1=*déclenchement d'une commutation* et p_2=*une seule batterie est en fonctionnement*. Dans [7], nous avons décrit le processus $\mathcal{P}_{sur}$ et défini une méthode pour extraire l'ensemble de prédicats à partir de l'objectif de test OT et du modèle comportemental M.

IV. Conclusion et perspectives

Nous avons présenté dans cet article un algorithme de génération d'abstraction à partir d'un ensemble de prédicats et

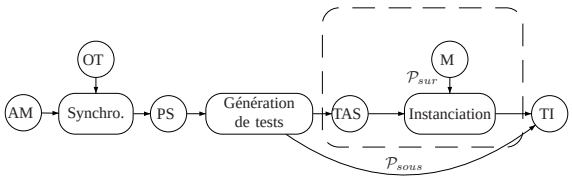

Fig. 3. Génération de tests à partir d'une abstraction et d'un objectif de test

son application à la génération de tests. Cet algorithme calcule d'une part une sur-approximation constituée par l'ensemble des transitions symboliques potentiellement déclenchables par l'ensemble des opérations présentes dans le système. Il calcule d'autre part, à la volée, une concrétisation de cette relation de transition qui est une sous-approximation. Nous présentons l'utilisation de ces approximations pour générer automatiquement des tests de deux manières. La première extrait des exécutions potentielles de la sur-approximation et tente de les instancier *a posteriori*. La seconde extrait directement des tests de la sous-approximation. Nous comparons les résultats obtenus sur des exemples selon plusieurs critères, notamment le temps de calcul des tests et les taux de couverture des abstractions.

***Keywords*-Abstraction, Résolution de contraintes, Preuve, Génération de tests à partir de modèles**

References

[1] B. Beizer, *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. New York, USA: John Wiley & Sons, 1995.

[2] M. Broy, B. Jonsson, J.-P. Katoen, M. Leucker, and A. Pretschner, Eds., *Model-Based Testing of Reactive Systems*, ser. LNCS. Springer, 2005, vol. 3472.

[3] M. Utting and B. Legeard, *Practical Model-Based Testing*. Morgan Kaufmann, 2006.

[4] S. Graf and H. Saïdi, "Construction of abstract state graphs with pvs," in *CAV'97*, ser. LNCS, vol. 1254, 1997, pp. 72–83.

[5] C. S. Pasareanu, R. Pelánek, and W. Visser, "Predicate abstraction with under-approximation refinement," *CoRR*, vol. abs/cs/0701140, 2007.

[6] L. de Moura and N. Bjørner, "An efficient smt solver," in *TACAS'08*, ser. LNCS, vol. 4963, 2008, pp. 337–340.

[7] F. Bouquet, P.-C. Bué, J. Julliand, and P.-A. Masson, "Test generation based on abstraction and test purposes to complement structural tests," in *A-MOST'10, 6th int. Workshop on Advances in Model Based Testing*, ser. IEEE proceedings of ICST'2010, Paris, France, Apr. 2010.

V. Biographies

Jacques Julliand et Pierre-Alain Masson sont respectivement professeur et maître de conférences au laboratoire d'informatique de l'Université de Franche-Comté. Leurs thèmes de recherche sont la vérification et la validation de systèmes critiques. Les mots-clés de leur activité sont : Vérification de propriétés de logique temporelle, génération de tests à partir de modèles, expression et test de propriétés de sécurité, test guidé par les propriétés et abstraction de modèles pour le test.

Pierre-Christophe Bué est doctorant au laboratoire d'informatique de l'Université de Franche-Comté. Il travaille sur une approche de génération de tests utilisant des objectifs de test dynamiques et des abstractions qui sur-approximent ou sous-approximent des modèles comportementaux.