Stage de master : analyse de robustesse par interprétation abstraite et programmation par contraintes

30 septembre 2016

Ce sujet est destiné à un étudiant souhaitant prolonger son master par une une thèse. Durant son stage de master, il pourra explorer la partie la plus en accord avec ses goûts et ses compétence ; par exemple :

- un étudiant plus intéressé par les aspects formels et théoriques pourra se focaliser sur la formalisation et la représentation de la robustesse dans la PPC (programmation par contraintes).
- un étudiant plus intéressé par les approches concrètes pourra se focaliser sur la génération automatique de cas de tests instables.
- un étudiant désirant se focaliser sur l'implémentation pourra participer au développement d'une plateforme commune entre interprétation abstraite et programmation par contraintes.

1 Sujet

L'analyse de robustesse de systèmes plus généraux que les programmes, par exemple les systèmes hybrides, qui modélisent les programmes de contrôle commande (embarqués critiques), et leur environnement physique (système d'équations différentielles, généralement) est un axe de recherche important. Une première approche, uniquement par interprétation abstraite et implémentée dans FLUCTUAT [3], a fait l'objet d'une expérience industrielle déjà positive [2]. Cette approche permet d'obtenir des résultats précis, car, pour bon nombre de systèmes de contrôle, la connaissance de la physique contrôlée est généralement indispensable à l'analyse du code. Cet axe est à l'heure actuelle en plein développement, particulièrement aux Etats-Unis, avec une idée en rupture à laquelle cette thèse devrait contribuer : les « autocodeurs » générant du code de contrôle prouvé (NSF CRAVES, Eric Féron, GeorgiaTech). Le terme « robustesse » ou « continuité » a également été récemment utilisé aussi bien en automatique qu'en informatique, ou dans la problématique plus générale des « cyberphysical systems » (des avions, automobiles, aux flottes d'automobiles, des smartgrids etc.), dans [7, 6, 11, 4], pour décrire la stabilité de programmes et de systèmes de contrôle soumis à des perturbations variées (incertitude due aux capteurs, à l'implémentation en précision finie, etc.).

Il s'agit donc de développer une analyse de robustesse de programmes et systèmes hybrides, par de nouvelles méthodes mêlant interprétation abstraite et programmation par contraintes. Les résultats visés sont soit l'amélioration de la précision de certaines analyses existantes de la robustesse, soit l'aide à la génération automatique de cas de tests instables. Interprétation abstraite et programmation par contraintes sont deux techniques d'analyse des valeurs atteignables qui ont montré leur capacités respectives en vérification de programmes. Elles font l'objet de nombreuses recherches pour en synthétiser les différentes qualités au sein d'un même système. On peut par exemple citer [10] où la programmation par contraintes permet de réduire les domaines calculés par l'interprétation abstraite, sur les variables du programme. Mais aussi [5], qui utilise déjà, dans un domaine d'interprétation abstraite, des contraintes pour exprimer les conditions de test instable et estimer la robustesse des conditionnelles. Ici, les contraintes sont utilisées pour exprimer des propriétés fines, sur des quantités internes aux domaines abstraits. Mais elles ne sont que très peu exploitées dans la résolution, et uniquement sous forme de contraintes linéarisées. Il est naturellement intéressant d'exprimer et exploiter également des contraintes non linéaires. Dans le cas du calcul de conditions de test instable, et des erreurs de discontinuité liées à ces tests instables, l'intérêt pour toute forme de coopération entre interprétation

abstraite et programmation par contraintes, est particulièrement criant, et c'est un point qu'il s'agira d'explorer dans cette thèse. En effet, l'explosion du nombre de couples de chemins possibles entre exécution flottante et réelle, et la possible divergence des calculs effectués dans chaque branche, rendent un calcul précis et efficace de bornes de ces discontinuités particulièrement difficile. Les conditions de test instable s'écrivent tout naturellement sous forme de contraintes, tandis qu'un calcul par interprétation abstraite est tout indiqué pour borner la divergence entre les différentes branches du programme. Il est donc naturel d'envisager une combinaison des deux pour en obtenir une analyse précise et qui passe à l'échelle.

Naturellement, cette problématique se retrouve lorsque le programme est considéré avec son interaction avec l'environnement, c'est-à-dire qu'on s'intéresse au système hybride.

L'analyse de robustesse peut naturellement être envisagée sous de nombreux angles. Dans cette thèse, il s'agira d'exploiter les qualités respectives de l'interprétation abstraite et de la programmation par contraintes afin d'améliorer les approches existantes. En particulier, les directions suivantes ont été identifiées :

— L'interprétation des tests dans les domaines d'interprétation abstraite peut s'inspirer de qui a été fait dans [5], pour traduire en contraintes les conditions d'instabilité des tests. En s'aidant d'autres éléments du projet COVERIF, il devrait être possible de déterminer une méthode mixte interprétation abstraite et programmation par contraintes, permettant de donner une surapproximation correcte des conditions de stabilité des tests, et des états après les conditionnelles, même en présence de comportements présentant une divergence de flot de contrôle, en présence d'incertitudes. Le calcul d'instabilité dans les tests en interprétation abstraite fait apparaître de nombreuses contraintes dont pour l'instant ne sont exploités que des linéarisations [5]. Une extension aux contraintes non-linéaires serait en mesure de pallier aux limites liées au linéaire.

— Une direction possible de ces recherches repose sur des solveurs de contraintes en nombres réels. L'I3S a développé depuis plusieurs années des solveurs sur les nombres flottants [1, 8, 9]. Il s'agirait de combiner la résolution de contraintes sur les nombres réels, qui permettent de résoudre les conditions d'instabilité des tests fournies par la sémantique abstraite, en considérant que la différence entre la sémantique flottante, ou réelle avec incertitudes, d'avec la sémantique réelle, est un nombre réel. Mais quand il s'agit de divergence de flots due aux arrondis en précision finie (virgule flottante), la structure fine de l'arithmétique sous-jacente est discrète, et pour nombre de problèmes subtils, l'utilisation d'un solveur en nombres flottants peut permettre d'atteindre des précisions bien meilleures. Cette approche devrait permettre un raffinement des méthodes d'analyse en en prouvant toujours la correction.

— Un des avantages de l'explicitation des contraintes donnant les cas d'instabilité dans les tests, est qu'il est possible non seulement de réduire les valeurs obtenues dans chaque branche des conditionnelles, mais aussi d'exhiber des valeurs des variables pour lesquelles on atteint effectivement une divergence de flot dans les conditionnelles, voire une divergence de flot amenant à une erreur, après la conditionnelle, maximale. L'objectif de cette sous-tâche est donc d'utiliser la programmation par contrainte pour exhiber des cas de tests pour lesquels une instabilité se produit ou pour démontrer l'absence d'instabilité.

2 Contexte

L'ensemble de ces travaux s'inscrivent dans le cadre de l'ANR COVERIF (`http://www.anr-coverif.fr`). Ils seront encadrés par l'I3S (`http://www.i3s.unice.fr`) en collaboration avec le LIX (`https://www.lix.polytechnique.fr`).

Le stage de master, comme la thèse, seront rénumérés grâce au soutien de l'ANR.

3 Contact

Pour de plus amples informations, n'hésitez pas à contacter l'une des personnes suivantes :

— Michel Rueher (I3S), michel.rueher@gmail.com

— Claude Michel (I3S), Claude.Michel@i3s.unice.fr

— Éric Goubault (LIX), goubault@lix.polytechnique.fr

— Sylvie Putot (LIX), putot@lix.polytechnique.fr

Références

[1] M. S. Belaid, C. Michel, and M. Rueher. Boosting local consistency algorithms over floating-point numbers. In *Proc. of the 17th Int. Conf on Principles and Practice of Constraint Programming (CP'12)*, pages 127–140, 2012.

[2] O. Bouissou, É. Conquet, P. Cousot, R. Cousot, J. Feret, K. Ghorbal, É. Goubault, D. Lesens, L. Mauborgne, A. Miné, S. Putot, X. Rival, and M. Turin. Space software validation using abstract interpretation. In *Proc. of the Int. Space System Engineering Conf. on Data Systems in Aerospace (DASIA 2009)*, volume SP-669, page 7. ESA, May 2009.

[3] O. Bouissou, E. Goubault, S. Putot, K. Tekkal, and F. Védrine. HybridFluctuat : A static analyzer of numerical programs within a continuous environment. In *Proc. of the 21st Int. Conf. on Computer Aided Verification (CAV'09)*, pages 620–626, 2009.

[4] S. Chaudhuri, S. Gulwani, and R. Lublinerman. Continuity and robustness of programs. *Commun. ACM*, 55(8) :107–115, 2012.

[5] E. Goubault and S. Putot. Robustness analysis of finite precision implementations. In *Proc. of the 11th Asian Symposium on Programming Languages and Systems (APLAS'13)*, volume 8301 of *LNCS*, pages 50–57. Springer, 2013.

[6] C.-Y. Kao, A. Megretzki, U. Jonsson, and A. Rantzer. A matlab toolbox for robustness analysis. In *IEEE International Symposium on Computer-Aided Control Systems Design*. IEEE, 2004.

[7] R. Majumdar and I. Saha. Symbolic robustness analysis. In *Proc. of the 30th IEEE Real-Time Systems Symposium (RTSS '09)*, pages 355–363. IEEE Computer Society, 2009.

[8] B. Marre and C. Michel. Improving the floating point addition and subtraction constraints. In *Proc. of the 16th Int. Conf. on Principles and Practice of Constraint Programming (CP'10)*, pages 260–267, 2010.

[9] C. Michel. Exact projection functions for floating point number constraints. In *Proc. Int.l Symp. on Artificial Intelligence and Mathematics (AI&M 2002)*, 2002.

[10] O. Ponsini, C. Michel, and M. Rueher. Verifying floating-point programs with constraint programming and abstract interpretation techniques. *Automated Software Engineering*, May 2014.

[11] S. Chaudhuri R. Samanta, J. Deshmukh. Robustness analysis of string transducers. In *ATVA*, pages 427– 441, 2013.