Le projet BWare : une plate-forme pour la vérification automatique d'obligations de preuve B

David Delahaye, Claude Marché, David Mentré

Le projet **BWare** : une plate-forme pour la vérification automatique d'obligations de preuve **B**

David Delahaye[1], Claude Marché[2] et David Mentré[3]
(pour le consortium du projet BWare)

[1] Cedric/Cnam/Inria, Paris, France
[2] Inria Saclay - Île-de-France & LRI, CNRS, Univ. Paris-Sud, Orsay, France
[3] Mitsubishi Electric R&D Centre Europe, Rennes, France

Le projet de recherche industrielle BWare (ANR-12-INSE-0010) est financé pour 4 ans par le programme « Ingénierie Numérique & Sécurité » (INS) de l'Agence Nationale de la Recherche (ANR) et a débuté en septembre 2012 (voir le site web du projet : `http://bware.lri.fr`). Le consortium du projet BWare associe les partenaires académiques Cedric, LRI, et Inria, ainsi que les partenaires industriels Mitsubishi Electric R&D Centre Europe (MERCE), ClearSy, et OCamlPro.

1 Présentation

Le projet BWare vise à produire un environnement permettant la vérification automatique d'obligations de preuve (OP) provenant du développement d'applications industrielles à haute intégrité utilisant la méthode B. Son cœur est la plate-forme générique de vérification déductive de programmes Why3 [2] intégrant différents outils de démonstration automatique tels que des systèmes au premier ordre et des solveurs SMT (« Satisfiability Modulo Theories »). Les outils au premier ordre considérés sont Zenon [4] et iProver Modulo [5], tandis que nous avons choisi Alt-Ergo [1] comme solveur SMT. Au-delà de l'aspect multi-outils, l'originalité du projet BWare réside également dans la production d'objets preuves par les outils de vérification. Pour vérifier indépendemment ces objets preuves, nous considérons deux vérificateurs : l'outil d'aide à la preuve Coq et le vérificateur de preuve universel Dedukti [3]. Pour évaluer notre méthodologie et tester notre plate-forme, une large collection d'OP est fournie par les partenaires industriels du projet qui développent des outils ou applications autour de la méthode B.

2 Résultats préliminaires

La plate-forme BWare est déjà opérationnelle. Les OP initialement produites par l'Atelier B sont traduites par un outil spécifique en buts Why3, qui sont fondés sur un encodage de la théorie des ensembles de B en Why3 [8]. La plate-forme Why3 permet alors d'envoyer ces OP aux outils de démonstration automatique, utilisant le format TPTP pour Zenon et iProver Modulo, et un format natif pour Alt-Ergo. Enfin, une fois que les preuves ont été trouvées par ces outils, certains peuvent générer des objets preuves pouvant être vérifiés : Zenon peut produire des objets preuves pour Coq et Dedukti [4, 7], et iProver Modulo des objets preuves pour Dedukti [6].

Pour évaluer la plate-forme BWare, deux partenaires industriels du projet ont fourni un banc d'essais initial de plus de 10 500 OP provenant de différentes applications industrielles : pour MERCE, il s'agit d'un cas d'utilisation complet d'un passage à niveau, et pour ClearSy, de trois projets industriels déployés. Les résultats obtenus au début du projet sont les suivants (obtenus sur une machine Intel Xeon X5650 2.67GHz, avec un temps limite de 30s) : le « main prover » (mp) de l'Atelier B (4.0) est capable de prouver 84% de ces OP, tandis qu'Alt-Ergo (0.95.1) obtient un taux de 58%, iProver Modulo (basé sur iProver 0.7) 19%, et Zenon (0.7.2) moins de 1%. Les

outils au premier ordre (iProver Modulo et surtout Zenon) rencontrent des difficultés parce que ces systèmes ne connaissent pas la théorie des ensembles de B. Concernant le solveur SMT Alt-Ergo, un ensemble intermédiaire de résultats obtenus avec des versions améliorées est publié sur le blog d'OCamlPro[1]. Ces résultats sont très prometteurs puisque la version de développement est maintenant capable de décharger automatiquement plus de 98% des OP.

3 Axes de travail actuels

Faute de place, nous ne pouvons pas décrire toutes les tâches du projet mais seulement deux axes de travail majeurs actuels. Le premier axe consiste à compléter en amont l'axiomatisation de la théorie des ensembles de B en Why3 pour pouvoir considérer toutes les OP fournies. Ce travail suit l'approche [8] en ajoutant des constructions B à l'axiomatisation et en modifiant le traducteur d'OP de l'Atelier B vers Why3 en conséquence. Ce travail nous permettra de considérer un large spectre d'OP et de tester également le passage à l'échelle de notre plate-forme.

Le second axe de travail se focalise sur les outils au premier ordre pour qu'ils puissent raisonner modulo la théorie des ensembles de B. Pour ce faire, nous utilisons la déduction modulo, une extension du calcul des prédicats permettant de réécrire des termes ainsi que des propositions, et qui est bien adaptée pour la recherche de preuve dans les théories axiomatiques, puisqu'elle transforme les axiomes en règles de réécriture. Nous avons ainsi étendu les deux outils au premier ordre pour obtenir Zenon Modulo [7] et iProver Modulo [5], deux extensions basées sur la déduction modulo et également capables de produire des preuves Dedukti [7, 6] (reposant aussi sur la déduction modulo). Actuellement, nos efforts sur cet axe consistent à construire une théorie des ensembles de B modulo adaptée à la démonstration automatique.

À plus long terme, nous prévoyons de faire une étude comparative plus complète des outils de vérification de manière à déterminer quel taux de couverture nous pouvons obtenir automatiquement avec notre plate-forme (notamment avec les outils de vérification étendus). Enfin, nous avons l'intention d'exploiter les résultats de notre projet en intégrant notre plate-forme à l'Atelier B, le transformant ainsi en un système multi-outils de vérification.

Références

[1] F. Bobot, S. Conchon, É. Contejean, M. Iguernelala, S. Lescuyer, and A. Mebsout. *Alt-Ergo, version 0.95.2*. CNRS, Inria, and Université Paris-Sud, 2013. `http://alt-ergo.lri.fr`.

[2] F. Bobot, J.-C. Filliâtre, C. Marché, and A. Paskevich. Why3 : Shepherd Your Herd of Provers. In *International Workshop on Intermediate Verification Languages (Boogie)*, 2011.

[3] M. Boespflug, Q. Carbonneaux, and O. Hermant. The λΠ-Calculus Modulo as a Universal Proof Language. In *Proof Exchange for Theorem Proving (PxTP)*, 2012.

[4] R. Bonichon, D. Delahaye, and D. Doligez. Zenon : An Extensible Automated Theorem Prover Producing Checkable Proofs. In *LPAR – Springer LNCS/LNAI 4790*, 2007.

[5] G. Burel. Experimenting with Deduction Modulo. In *CADE – Springer LNCS/LNAI 6803*, 2011.

[6] G. Burel. A Shallow Embedding of Resolution and Superposition Proofs into the λΠ-Calculus Modulo. In *Proof Exchange for Theorem Proving (PxTP) – EasyChair EPiC 14*, 2013.

[7] D. Delahaye, D. Doligez, F. Gilbert, P. Halmagrand, and O. Hermant. Zenon Modulo : When Achilles Outruns the Tortoise using Deduction Modulo. In *LPAR – Springer LNCS/ARCoSS 8312*, 2013.

[8] D. Mentré, C. Marché, J.-C. Filliâtre, and M. Asuka. Discharging Proof Obligations from Atelier B Using Multiple Automated Provers. In *ABZ – Springer LNCS 7316*, 2012.

1. Voir : `http://www.ocamlpro.com/blog/2013/10/22/alt-ergo-evaluation-october-2013.html`.