Amélioration des raisonneurs du langage B avec des techniques SMT

Vincent Trélat

Université de Lorraine, CNRS, Inria, LORIA, Nancy, France

Résumé

Les récents progrès en déduction automatisée ont permis l'extension des techniques de déduction au premier ordre à la logique d'ordre supérieur (HOL). L'objectif principal de ma thèse est de rendre ces avancées disponibles pour la méthode B, augmentant ainsi de manière significative le degré d'automatisation des preuves.

1 Introduction

La méthode B [2] est une méthode formelle pour le développement logiciel fondée sur un langage ensembliste expressif dans lequel on décrit des machines abstraites [1] et sur la notion de raffinement et la génération d'obligations de preuve. L'outil Atelier B [10] intègre cette méthode pour générer du code vérifié par raffinements successifs. Les propriétés à vérifier sont exprimées sous forme d'invariants et de propriétés de sûreté, prouvées par des solveurs automatiques. Toutefois, certaines obligations de preuve nécessitent une intervention humaine. Actuellement, les obligations de preuve générées sont traduites en logique du premier ordre (FOL) multi-sortes (*many-sorted first-order logic*) pour être prouvées par des solveurs internes – comme le *Predicate Prover* d'Atelier B [14] – ou externes comme des solveurs SMT [12].

Les récentes avancées en raisonnement automatisé ont permis d'étendre ces solveurs à la logique d'ordre supérieur [6, 19], ouvrant la voie à une automatisation plus large. Concrètement, la logique d'ordre supérieur permet un encodage plus direct des obligations de preuve de B qui utilise la théorie des ensembles. Ma recherche de thèse est motivée par l'idée que l'utilisation de cet encodage permettrait une meilleure automatisation des preuves.

2 Encodage des obligations de preuve de B

2.1 Le traducteur *ppTrans*

En B, tout est représenté par des ensembles, y compris les fonctions, qui sont des relations binaires fonctionnelles (i.e. injectives sur la deuxième composante). Ces ensembles sont spécifiés au travers de prédicats d'appartenance non interprétés décrivant les relations entre les éléments atomiques (par exemple, les entiers) et les ensembles. L'encodage actuel des obligations de preuve générées par Atelier B est basé sur une réduction au premier ordre. Le traducteur, appelé *ppTrans*, fournit un système de réécriture basé sur environ 70 règles. On donne ci-dessous en guise d'exemple deux règles, en notant $\hookrightarrow$ la réécriture en expressions de logique du premier ordre :

$$s \in \mathcal{P}(t) \quad \hookrightarrow \quad \forall X.X \in s \to X \in t \qquad \text{(PowD)}$$
$$f \in s \nrightarrow t \quad \hookrightarrow \quad f \in s \leftrightarrow t \wedge \texttt{is\_func}(f) \qquad \text{(pfunI)}$$

où $\mathcal{P}(t)$ est l'ensemble des parties de t, `is_func` est un prédicat qui évalue si une relation binaire est fonctionnelle, $s \nrightarrow t$ est l'espace des fonctions partielles de s dans t et $s \leftrightarrow t$ est l'espace des relations binaires entre s et t.

La traduction des modèles B est exprimée dans un sous-ensemble du langage mathématique de B et dont le seul symbole de théorie des ensembles est le symbole non interprété $\in$ qui dénote l'appartenance. Ce traducteur effectue les transformations suivantes :

décomposition des relations (binaires). Considérons par exemple la formule ψ définie par $\psi := (a = b) \wedge (a \in S)$ bien typée dans $\Gamma := \{a : S, b : S, S : \mathcal{P}(\mathbb{Z} \times \mathbb{Z})\}$. On réécrit ψ comme suit :

$$\begin{aligned} \psi \quad \hookrightarrow \quad & \forall x_0 x_1 x_2 x_3.(a = x_0 \mapsto x_1 \wedge b = x_2 \mapsto x_3) \to \\ & (x_0 \mapsto x_1 = x_2 \mapsto x_3) \wedge (x_0 \mapsto x_1 \in S) \end{aligned}$$

où $\mapsto$ est le symbole *maplet* de B qui dénote les paires.

purification. Cette étape sépare les termes de différentes catégories syntaxiques (arithmétique, théorie des ensembles, etc.). Par exemple, la formule $\psi := a \mapsto (1 \mapsto 2) \in S$ est réécrite en $x_0 = 1 \wedge x_1 = 2 \wedge a \mapsto (x_0 \mapsto x_1) \in S$ où x_0 et x_1 sont des variables fraiches. Si ψ apparait dans la portée d'un quantificateur Q, les variables x_0 et x_1 seraient également quantifiées par Q dans la nouvelle formule.

simplification des expressions booléennes. Par exemple :

$$(\psi \wedge \top) \vee (\top \to \varphi) \vee (\bot \wedge \top) \quad \hookrightarrow \quad \psi \vee \varphi$$

Cette approche comporte quelques inconvénients. La réécriture fournie par le traducteur *ppTrans* fait exploser la taille des formules, en ajoutant de nouvelles

variables auxiliaires quantifiées. De plus, il est nécessaire en pratique d'introduire un prédicat d'appartenance par type, ce qui alourdit notamment la traduction au niveau des équations fonctionnelles, au sens où $y = f(x)$ est traduit en $x \mapsto y \in_\kappa f$ où $\in_\kappa$ est un prédicat d'appartenance spécifique au type de la fonction f. Enfin, la quantification est possible uniquement sur des variables (par exemple, les entiers).

Jusqu'à présent, une approche par prédicats caractéristiques avait été écartée car le langage SMT-LIB 2.6 a l'expressivité de la logique du premier ordre multi-sortes [7], et ne supporte donc ni les λ-expressions ni le polymorphisme de types (d'où la duplication des prédicats d'appartenance). Bien que l'approche actuelle fournisse un lien direct entre la logique du premier ordre et la théorie de l'égalité avec fonctions non interprétées (EUF), elle entrave le raisonnement sur les ensembles d'ensembles, qui nécessitent un encodage à l'ordre supérieur.

Les avancées récentes à l'ordre supérieur motivent l'exploration d'un encodage par prédicats caractéristiques, d'autant plus avec la proposition préliminaire de la version 3.0 de SMT-LIB [8] qui apporte l'ordre supérieur et les types dépendants, ce qui permet de définir des prédicats caractéristiques pour des ensembles quelconques, supprimant le besoin de prédicat d'appartenance. Notons que le prédicat caractéristique d'un ensemble S existe toujours : soit τ le type des éléments de S, considérer alors le prédicat $\lambda x : \tau.\ x \in S$. On note $\hat{S}$ le prédicat caractéristique de S. Ainsi, $x \in S$ se réécrit simplement en $\hat{S}(x)$.

Par exemple, considérons l'ensemble $S := \{1, 2, 3\}$. Son prédicat caractéristique s'écrit $\hat{S} := \lambda x.x = 1 \vee x = 2 \vee x = 3$. La formule $2 \in S$ se réécrit alors comme suit :

$$\begin{aligned} 2 \in S \quad &\hookrightarrow \quad \hat{S}(2) \\ &\triangleq \quad (\lambda x.x = 1 \vee x = 2 \vee x = 3)(2) \\ &\hookrightarrow \quad 2 = 1 \vee 2 = 2 \vee 2 = 3 \\ &\hookrightarrow \quad \text{True} \end{aligned}$$

En suivant la même notation, si $\mathcal{E} := \{E_1, \ldots, E_n\}$ est un ensemble d'ensembles, on peut représenter $\mathcal{E}$ par le prédicat caractéristique (d'ordre supérieur) suivant :

$$\hat{\mathcal{E}} := \lambda e.\ \bigvee_{i=1}^{n} e = \hat{E}_i$$

3 Instanciation à l'ordre supérieur

Les quantificateurs, déjà au premier ordre, sont une source de difficulté pour les solveurs SMT. Il existe plusieurs approches pour résoudre le problème de l'instanciation de quantificateurs, notamment par *triggers* [13], par conflits [5], par modèles [11], et par énumération [16]. Les avancées récentes en déduction automatisée ont permis d'étendre certaines techniques d'instanciation à l'ordre supérieur [6] mais peu d'entre elles sont efficaces [17, 18]. Dans le cadre de ma thèse, je m'intéresse à l'une d'entre elles nommée *Higher-Order Congruence Closure with Free*

Variables (HOCCFV) [19]. Elle est pour l'instant définie sur la logique d'ordre supérieur sans λ-abstraction (notée λ-fHOL) et utilise la clôture de congruence avec variables libres dans une approche par conflits avec unification équationnelle (E-*matching*) [3]. Cette technique semble insuffisante pour traiter les obligations de preuve de B, mais elle permet déjà de résoudre certains problèmes d'instanciation.

De manière informelle, soit E une théorie (i.e. un ensemble d'équations exprimées en logique λfHOL), appelé ensemble *ground* dans la technique HOCCFV et φ une clause de litéraux de λfHOL, i.e. un ensemble conjonctif de (non-)égalités sur des symboles de constantes, variables ou fonctions appliquées (potentiellement partiellement). On cherche alors une substitution σ telle que $E \models \varphi\sigma$. Si une telle substitution existe, on obtient un modèle C de φ dans E. On peut alors itérer ce processus et chercher une substitution σ' telle que $E, \neg C \models \varphi\sigma'$ jusqu'à l'obtention d'une formule insatisfiable, si elle existe. Si tel est le cas, on peut obtenir un contre-modèle de φ dans E, ce qui prouve que φ est insatisfiable dans E. C'est généralement ce que l'on cherche à obtenir lorsqu'on traite les obligations de preuve de B dans un solveur SMT : on essaye de réfuter leur négation.

En pratique, l'intérêt est qu'on obtient une unification à l'ordre supérieur. Par exemple, pour une relation de congruence notée $\simeq$, considérons l'ensemble *ground* E et la clause L définis comme suit :

$$E := \{f(g(a)) \simeq h(b)\} \quad \text{et} \quad L := \{y(a) \simeq x\}$$

La procédure décrite dans HOCCFV permet par exemple d'obtenir la substitution $\sigma := \{y \mapsto f(g), x \mapsto h(b)\}$ qui instancie le symbole de fonction y par le terme d'ordre supérieur $f(g)$ – vu comme un symbole de fonction partiellement appliqué, comprendre $f \circ g$.

Il reste encore à implémenter cette procédure dans un solveur SMT. Des tests manuels extensifs de la méthode seront réalisés avec des solveurs SMT existants (CVC5 [4], VeriT [9]) qui supportent au moins partiellement l'ordre supérieur. Des efforts sont également en cours pour formaliser les résultats théoriques de la technique HOCCFV dans un assistant de preuve (par exemple, Isabelle/HOL [15]).

4 Conclusion

Étant données les limitations de l'encodage actuel des obligations de preuve B en logique du premier ordre – notamment l'explosion de la taille des formules et la complexité des traductions – je propose une alternative motivant l'utilisation de prédicats caractéristiques rendue possible par les récentes extensions des solveurs SMT à l'ordre supérieur. Cette approche promet de simplifier les traductions et d'améliorer l'efficacité des solveurs SMT en reconnaissant plus facilement les schémas fréquents dans les spécifications B. Un de ces schémas fréquents est l'utilisation de fonctions du premier ordre. Plus récemment, je réfléchis à un encodage plus spécifique pour ces fonctions qui permettrait de les traiter plus efficacement.

Références

[1] Jean-Raymond Abrial. *The B-Book : Assigning Programs to Meanings*. Cambridge University Press, USA, 1996.

[2] Jean-Raymond Abrial, Matthew Lee, D. Neilson, P. Scharbach, and I. Sørensen. *The B-method*, volume 2, pages 398–405. 01 2006.

[3] Leo Bachmair and Harald Ganzinger. Rewrite-based Equational Theorem Proving with Selection and Simplification. *Journal of Logic and Computation*, 4(3) :217–247, 06 1994.

[4] Haniel Barbosa, Clark Barrett, Martin Brain, Gereon Kremer, Hanna Lachnitt, Makai Mann, Abdalrhman Mohamed, Mudathir Mohamed, Aina Niemetz, Andres Nötzli, Alex Ozdemir, Mathias Preiner, Andrew Reynolds, Ying Sheng, Cesare Tinelli, and Yoni Zohar. cvc5 : A Versatile and Industrial-Strength SMT Solver. In Dana Fisman and Grigore Rosu, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 415–442, Cham, 2022. Springer International Publishing.

[5] Haniel Barbosa, Pascal Fontaine, and Andrew Reynolds. Congruence closure with free variables. In Axel Legay and Tiziana Margaria, editors, *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, volume 10206 of *Lecture Notes in Computer Science*, pages 214–230, 2017.

[6] Haniel Barbosa, Andrew Reynolds, Daniel El Ouraoui, Cesare Tinelli, and Clark Barrett. Extending SMT solvers to higher-order logic. In Pascal Fontaine, editor, *Proceedings of the 27th International Conference on Automated Deduction (CADE '19)*, volume 11716 of *Lecture Notes in Artificial Intelligence*, pages 35–54. Springer, August 2019. Natal, Brazil.

[7] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. The SMT-LIB Standard : Version 2.6. Technical report, Department of Computer Science, The University of Iowa, 2017. Available at `www.SMT-LIB.org`.

[8] Clark Barrett, Pascal Fontaine, and Cesare Tinelli. SMT-LIB 3.0 - Preliminary Proposal. `http://smtlib.cs.uiowa.edu/version3.shtml`, 2021.

[9] Thomas Bouton, Diego Caminha B. de Oliveira, David Déharbe, and Pascal Fontaine. veriT : An Open, Trustable and Efficient SMT-Solver. In Renate A. Schmidt, editor, *Automated Deduction – CADE-22*, pages 151–156, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[10] Clearsy. Atelier B. `https://www.atelierb.eu`.

[11] Leonardo de Moura and Nikolaj Bjørner. Efficient E-Matching for SMT Solvers. In Frank Pfenning, editor, *Automated Deduction – CADE-21*, pages 183–198, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[12] David Déharbe, Pascal Fontaine, Yoann Guyot, and Laurent Voisin. Integrating SMT solvers in Rodin. *Science of Computer Programming*, 94, 11 2014.

[13] Yeting Ge and Leonardo de Moura. Complete instantiation for quantified formulas in satisfiability modulo theories (extended version). 2009.

[14] Matthias Konrad. Translation from Set-Theory to Predicate Calculus. Technical report, ETH Zurich, 2012.

[15] Tobias Nipkow, Lawrence C Paulson, and Markus Wenzel. *Isabelle/HOL : a proof assistant for higher-order logic*, volume 2283. Springer Science & Business Media, 2002.

[16] Andrew Reynolds, Haniel Barbosa, and Pascal Fontaine. Revisiting enumerative instantiation. In Dirk Beyer and Marieke Huisman, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part II*, volume 10806 of *Lecture Notes in Computer Science*, pages 112–131. Springer, 2018.

[17] Geoff Sutcliffe and Martin Desharnais. CASC-23. `https://tptp.org/CASC/29/WWWFiles/DivisionSummary1.html`, 2023.

[18] Geoff Sutcliffe and Martin Desharnais. The 11th IJCAR Automated Theorem Proving System Competition - CASC-J11. *AI Communications*, 36(2) :73–91, 2023.

[19] Sophie Tourret, Pascal Fontaine, Daniel El Ouraoui, and Haniel Barbosa. Lifting congruence closure with free variables to λ-free higher-order logic via SAT encoding. In *SMT 2020 - 18th International Workshop on Satisfiability Modulo Theories*, Online COVID-19, France, July 2020.