Projet de thèse

SMT modulo

Sujet

En matière de déduction automatique pour la logique classique du premier ordre, il y a aujourd'hui globalement deux approches. Il y a les outils au premier ordre, basés sur différentes méthodes de recherche de preuve, comme la méthode des tableaux ou la résolution, et il y a les solveurs SMT (« Satisfiability Modulo Theories »), basés sur des noyaux SAT (capables de résoudre des problèmes propositionnels) et qui peuvent combiner plusieurs théories décidables en utilisant différentes méthodes, comme celles de Nelson-Oppen [8] ou de Shostak [9].

L'avantage des outils au premier ordre est, outre le fait qu'ils soient capables de raisonner dans n'importe quelle théorie (décidable ou non), qu'ils sont complets au premier ordre et reposent sur des techniques efficaces d'instanciation. En revanche, ils restent bien moins efficaces que les solveurs SAT sur le fragment propositionnel, dans lequel ces derniers sont spécialisés. Quant aux solveurs SMT, s'ils sont capables de raisonner efficacement sur un certain nombre de théories décidables bien connues (égalité sur les fonctions non interprétées, arithmétique linéaire, etc.), leur force réside surtout dans les appels systématiques aux noyaux SAT sur lesquels ils reposent. Cependant, les solveurs éprouvent des difficultés au premier ordre, même s'il existe des techniques comme les « triggers » pour trouver certaines instanciations, mais ces techniques ne sont généralement pas complètes et sont plus des heuristiques et des stratégies.

La déduction modulo [6] est 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. La déduction modulo a bien été étudiée dans le cadre d'outils au premier ordre avec la réalisation d'extensions d'outils existants utilisant différentes méthodes de recherche de preuve, comme les tableaux [1] avec des outils comme Zenon Modulo [5] (extension de Zenon [2] à la déduction modulo), ou la résolution [6] avec des outils comme iProver Modulo [4] (extension d'iProver [7] à la déduction modulo). Dans toutes ces extensions, la

déduction modulo a permis une amélioration significative du nombre de problèmes prouvés par ces outils (tests réalisés sur la bibliothèque de problèmes TPTP [10]), ainsi que la preuve de problèmes difficiles (c'est-à-dire prouvés par peu voire aucun autre outil existant). Cependant, la déduction modulo n'a jamais été étudiée dans le cadre des solveurs SMT, et c'est ce que nous nous proposons de faire dans cette thèse. Cela répond à un double besoin : celui de mesurer l'impact de la déduction modulo sur les méthodes de recherche de preuve actuelles, et aussi celui de la communauté SMT, qui depuis plusieurs années, commence à essayer de comprendre ce que pourrait apporter l'intégration de la réécriture aux techniques de SMT.

Concrètement, pour entreprendre cette étude, nous proposons de le faire dans le cadre d'un outil expérimental, que nous aurons au préalable réalisé. Dans cet outil, nous souhaitons prendre le meilleur des deux mondes entre les outils au premier ordre et les solveurs SMT, et de réaliser un outil au premier ordre avec un noyau SAT. Ce nouvel outil permettrait de raisonner efficacement sur la partie propositionnelle et d'utiliser les techniques efficaces d'instanciation pour la partie premier ordre. La méthode au premier ordre choisie est la méthode des tableaux, qui est une méthode de recherche de preuve dans le calcul des séquents sans coupure, et qui a l'avantage de pouvoir produire des preuves très directement. Informellement, l'interaction entre instanciation et noyau SAT sera gérée de la manière suivante : appliquer toutes les règles de la méthode des tableaux ; appeler ensuite le noyau SAT ; s'il répond que l'ensemble de clauses est insatisfiable alors on s'arrête (la formule initiale est valide), sinon il nous rend un modèle, que l'on va chercher à invalider par unification de deux clauses contradictoires ; après avoir appliqué la substitution provenant de l'unification, on rappelle le noyau SAT, et ainsi de suite (cette méthode peut potentiellement boucler).

Des travaux similaires existent autour de cette question. Il y a en particulier l'outil Satallax [3], qui est basé sur une méthode des tableaux avec un noyau SAT mais pour la logique d'ordre supérieure. Dans notre cas, nous sommes au premier ordre, ce qui simplifie grandement l'approche et surtout l'unification devient décidable. De même, il y a également l'outil iProver [7], qui repose sur la résolution avec un noyau SAT, c'est-à-dire une méthode complètement différente de celle que nous proposons. À ce jour, aucun outil au premier ordre basé sur la méthode des tableaux avec un noyau SAT n'a été conçu, ni implanté, ce qui assure à ce travail une certaine originalité.

Ce outil réalisé servira de base à nos investigations sur l'intégration de la déduction modulo aux techniques utilisées par les SMT. En effet, puisque cet outil possédera un noyau SAT, nous pourrons utiliser les méthodes de combinaison de théories des solveurs SMT, transformant ainsi notre outil en solveur SMT, tandis que nous aurons une partie premier ordre plus efficace (par rapport à l'instanciation) que les solveurs SMT traditionnels. Quant à l'intégration de la déduction modulo, contrairement aux solveurs SMT qui raisonnent sur des théories décidables données, la déduction modulo permet d'intégrer de manière générique des théories au premier ordre (qui n'ont pas de contraintes particulières et peuvent être de n'importe quelle forme) à des méthodes de recherche de preuve.

Dans un second temps, notre objectif durant cette thèse sera donc de voir comment combiner ces deux techniques au sein d'un même outil de déduction automatique.

Travail à réaliser

— Concevoir et implanter une méthode de recherche de preuve au premier ordre basée sur les tableaux et utilisant un noyau SAT ;
— Étendre la méthode précédemment conçue aux techniques de SMT, en considérant un certain nombre de théories (« congruence closure », arithmétique, etc.) ;
— Comprendre comment intégrer la déduction modulo dans ce cadre, puis étendre l'outil réalisé en conséquence et le tester sur les bibliothèques TPTP et SMT-LIB.

Références

[1] R. Bonichon. TaMeD : A Tableau Method for Deduction Modulo. In *International Joint Conference on Automated Reasoning (IJCAR)*, volume 3097 of *LNCS*, pages 445–459, Cork (Ireland), July 2004. Springer.

[2] R. Bonichon, D. Delahaye, and D. Doligez. Zenon : An Extensible Automated Theorem Prover Producing Checkable Proofs. In *Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, volume 4790 of *LNCS/LNAI*, pages 151–165, Yerevan (Armenia), Oct. 2007. Springer.

[3] C. E. Brown. Satallax : An Automatic Higher-Order Prover. In *International Joint Conference on Automated Reasoning (IJCAR)*, volume 7364 of *LNCS*, pages 111–117, Manchester (UK), June 2012. Springer.

[4] G. Burel. Experimenting with Deduction Modulo. In *Conference on Automated Deduction (CADE)*, volume 6803 of *LNCS/LNAI*, pages 162–176, Wrocław (Poland), July 2011. Springer.

[5] D. Delahaye, D. Doligez, F. Gilbert, P. Halmagrand, and O. Hermant. Zenon Modulo : When Achilles Outruns the Tortoise using Deduction Modulo. In *Logic for Programming Artificial Intelligence and Reasoning (LPAR)*, volume 8312 of *LNCS/ARCoSS*, pages 274–290, Stellenbosch (South Africa), Dec. 2013. Springer.

[6] G. Dowek, T. Hardin, and C. Kirchner. Theorem Proving Modulo. *Journal of Automated Reasoning (JAR)*, 31(1) :33–72, Sept. 2003.

[7] K. Korovin. iProver – An Instantiation-Based Theorem Prover for First-Order Logic (System Description). In *International Joint Conference on Automated Reasoning (IJCAR)*, volume 5195 of *LNCS*, pages 292–298, Sydney (Australia), Aug. 2008. Springer.

[8] G. Nelson and D. C. Oppen. Simplification by Cooperating Decision Procedures. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 1(2) :245–257, Oct. 1979.

[9] R. E. Shostak. Deciding Combinations of Theories. *Journal of the Association for Computing Machinery (JACM)*, 31(1) :1–12, Jan. 1984.

[10] G. Sutcliffe. The TPTP Problem Library and Associated Infrastructure : The FOF and CNF Parts, v3.5.0. *Journal of Automated Reasoning (JAR)*, 43(4) :337–362, Dec. 2009.