Tuning the Alt-Ergo SMT Solver for B Proof Obligations

Sylvain Conchon, Mohamed Iguernelala

Tuning the Alt-Ergo SMT Solver for B Proof Obligations

Sylvain Conchon[1,2] and Mohamed Iguernelala[3,1]

[1] LRI, Université Paris-Sud, Orsay F-91405
[2] INRIA Saclay – Ile-de-France, Toccata, Orsay, F-91893
[3] OCamlPro SAS, Gif-sur-Yvette F-91190

Abstract. In this paper, we present recent developments on the Alt-Ergo SMT-solver to efficiently discharge proof obligations (PO) generated by Atelier B. This includes a new plugin architecture to facilitate experiments with different SAT engines, new heuristics to handle quantified formulas, and important modifications in its internal data structures to boost performances of core decision procedures. Benchmarks realized on more than 10,000 PO generated from industrial B projects show significant improvements.

Keywords: SMT solvers, B Proof Obligations, B Method

1 The Alt-Ergo SMT Solver

Alt-Ergo is an open-source SMT solver capable of reasoning in a combination of several built-in theories such as uninterpreted equality, integer and rational arithmetic, arrays, records, enumerated data types and AC symbols. It is the unique SMT solver that natively handles polymorphic first-order quantified formulas, which makes it particularly suitable for program verification. For instance, Alt-Ergo is used as a back-end of SPARK and Frama-C to discharge proof obligations generated from Ada and C programs, respectively.

Recently, we have started using Alt-Ergo in the context of the ANR project BWare [10] which aims at integrating SMT solvers as back-ends of Atelier B. The proof obligations sent to Alt-Ergo are extracted from Atelier B as logical formulas that are combined with a (polymorphic) model of B's set theory [8]. This process relies on the Why3 plateform [7] which can target a wide range of SMT solvers. However, we show (Section 2) on a large benchmark of industrial B projects that it is not immediate to obtain a substantial gain of performances by using only SMT solvers. Without a specific tunning for B, Alt-Ergo together with other SMT solvers compete just equally with Atelier B's solver on those industrial benchmarks.

In this paper, we report on recent developments in Alt-Ergo that significantly improve its capacities to handle PO of Atelier B. Our improvements are: (1) better heuristics for instantiating polymorphic quantified formulas from B model; (2) new efficient internal data structures; (3) a plugin architecture to facilitate experiments with different SAT engines; and (4) the implementation of a new CDCL-based SAT solver.

2 Benchmarks

The test-suite of BWare contains 10572 formulas obtained from three industrial B projects provided by ClearSy and Mitsubishi Electric. The first one, called DAB, is an automated teller machine (ATM). The last two ones, called P4 and P9, are obfuscated and unsourced programs.

The formulas generated from these projects are composed of two parts. The first one is the *context*, a large set of axioms (universally and polymorphic quantified formulas). The second one is the *goal*, a ground, unique and large formula. A more thorough investigation of the shape of these formulas shows that they mainly contain equalities over uninterpreted function symbols and atoms involving enumerated data types. Only a small portion of atoms contains linear integer arithmetic and polymorphic records. In comparison with other benchmarks coming from deductive program verification plateforms, the average number of axioms, as well as the size of these PO are much larger in this test suite, as shown in the following table:

	number of POs	avg. number of axioms	avg. size (Ko)
ANR Decert	80	12	6
VSTTE-Comp	125	32	8
Why3 gallery	1920	41	9
Hi-Lite	3431	125	23
DAB	860	*257*	236
P4	9341	*259*	248
P9	371	*284*	402

From what we know, the solver of Atelier B 4.0 proves 84% of the BWare benchmark [**?**].

Timeout was set to 30 seconds and memory was limited to 2GB per formula. Benchmarks descriptions and the results of our evaluation are given below.

DAB: 860, P4: 9341, P9: 371

Provers Versions	Alt-Ergo *0.95.1*	Alt-Ergo *0.95.2*	z3 *4.3.1*	cvc3 *2.4.1*
DAB	707 82.2 %	**822** **95.6 %**	716 83.3%	684 79.5 %
P4	4709 50.4 %	**8402** **89.9 %**	7974 85.4 %	7981 85.4 %
P9	181 48.8 %	**213** **57.4 %**	162 43.7 %	108 29.1 %
Total	5597 52.9 %	**9437** **89.3 %**	8852 83.7 %	8852 83.0 %

3 Improvements

Versions	Alt-Ergo *0.95.2*	Alt-Ergo *master branch*	Ctrl-Alt-Ergo *master branch*
DAB	822 95.6 %	858 99.8 %	**860** **100 %**
P4	8402 89.9 %	8980 96.1 %	9236 **98.9 %**
P9	213 57.4 %	234 63.1 %	277 **74.7 %**
Total	9437 89.3 %	10072 95.3 %	**10373** **98.1 %**

4 Conclusion and Future Works

Originally developed at LRI, it is now maintained and distributed by the OCaml-Pro company.

As can be expected, the BWare project consists of several tasks, which cannot be exhaustively described in this paper due to space restrictions. We therefore choose to focus on two major current lines of work of the project.

The first current line of work is upstream and consists in completing the axiomatization of the B set theory in Why3 in order to be able to consider all the provided proof obligations. This is mainly carried out according to what is described in [9], i.e. by adding set constructs to the axiomatization and modifying the translator of proof obligations from Atelier B to Why3 accordingly. This line of work is quite important in the project as it will allow us to consider a broad scope of proof obligations related to different application domains and test the scalability of our platform as well.

A second current line of work focuses on the first order provers to make them able to reason modulo the B set theory. To do so, we rely on deduction modulo. The theory of deduction modulo is an extension of predicate calculus, which allows us to rewrite terms as well as propositions, and which is well suited for proof search in axiomatic theories, as it turns axioms into rewrite rules. Both first order provers considered in the project have been extended to deduction modulo to obtain Zenon Modulo [5,6] and iProver Modulo [2]. Both tools have also been extended to produce Dedukti proofs (see [5,6] and [3]), which is natural as Dedukti relies on deduction modulo as well. Currently, most of the efforts in this line of work consist in building a B set theory modulo, which is appropriate for automated deduction and keeps some properties such as cut-free completeness. [4]

References

1. *Abstract State Machines, Alloy, B, VDM, and Z - Third International Conference, ABZ 2012, Pisa, Italy, June 18-21, 2012. Proceedings*, volume 7316 of *Lecture Notes in Computer Science*. Springer, 2012.

2. 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.
3. G. Burel. A Shallow Embedding of Resolution and Superposition Proofs into the $\lambda\Pi$-Calculus Modulo. In *Proof Exchange for Theorem Proving (PxTP)*, volume 14 of *EPiC*, pages 43–57, Lake Placid (USA), June 2013. EasyChair.
4. D. Déharbe, P. Fontaine, Y. Guyot, and L. Voisin. SMT Solvers for Rodin. In *ABZ* [1], pages 194–207.
5. D. Delahaye, D. Doligez, F. Gilbert, P. Halmagrand, and O. Hermant. Proof Certification in Zenon Modulo: When Achilles Uses Deduction Modulo to Outrun the Tortoise with Shorter Steps. In *International Workshop on the Implementation of Logics (IWIL)*, Stellenbosch (South Africa), Dec. 2013. EasyChair. To appear.
6. 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.
7. J.-C. Filliâtre and A. Paskevich. Why3 - where programs meet provers. In *ESOP*, volume 7792 of *Lecture Notes in Computer Science*, pages 125–128. Springer, 2013.
8. D. Mentré, C. Marché, J.-C. Filliâtre, and M. Asuka. Discharging proof obligations from atelier b using multiple automated provers. In *ABZ* [1], pages 238–251.
9. D. Mentré, C. Marché, J.-C. Filliâtre, and M. Asuka. Discharging Proof Obligations from Atelier B Using Multiple Automated Provers. In *Abstract State Machines, Alloy, B, VDM, and Z (ABZ)*, volume 7316 of *LNCS*, pages 238–251, Pisa (Italy), June 2012. Springer.
10. The BWare Project, 2012. `http://bware.lri.fr/`.