BWare B Proof Obligation Benchmark v1

David Delahaye Pierre Halmagrand
(for the BWare project consortium*)

2015-03

1 Contents

This package contains the version 1 of the B Proof Obligation (PO) benchmark of the BWare project [9, 12]. This package is under the CeCILL-B license[1], and consists of a set of 12,876 POs, which has been provided by ClearSy and Mitsubishi Electric R&D Centre Europe, two industrial partners of the BWare project. These POs files are not in the PO format of Atelier B [7], but are outputs of the BWare platform, which is built upon the Why3 platform [5] for deductive program verification. These POs are therefore intended to be used by automated deduction tools, like first order Automated Theorem Provers (ATP) or SMT (Satisfiability Modulo Theories) solvers.

To produce these PO files, the BWare platform works as follows. The proof obligations are initially produced by Atelier B. They are then translated by a specific tool into Why3 files, which are compatible with a Why3 encoding [10] of the B set theory [1]. Finally, from these files, Why3 can produce (by means of appropriate drivers) the proof obligations for a large set of automated deduction tools. This translation together with the encoding of the B set theory aims to generate valid statements that are appropriate for the automated deduction tools, i.e. whose proofs can be found by these tools. For further information regarding the BWare platform, the interested reader is referred to [9].

There are mainly two categories of formats depending on the target automated deduction tools: formats from the TPTP library [11] for first order ATPs, and formats from SMT-LIB [2] for SMT solvers. Within these two categories, there are several formats according to whether the downstream automated deduction tool is able to handle extensions of first order logic with elaborated type systems. More precisely, the encoding of the B set theory in Why3 is expressed using polymorphic types. Some tools (but very few), such as Zenon Modulo [8] and Alt-Ergo [4], are able to directly deal with polymorphic

*The BWare project [9, 12] (ANR-12-INSE-0010) is funded by the INS programme of the French National Research Agency (ANR).

[1] For more information, see: http://www.cecill.info/licences.en.html.

types, and the generation of POs by Why3 is therefore almost straightforward for these tools. For the other tools, Why3 relies on a first order encoding of the polymorphic layer, which is described in [6].

The proposed formats are divided into several archives and are the following:

- TPTP FOF format: this is the regular format for mono-sorted first order logic that is handled by most of the first order ATPs;
- TPTP TFF1 format: this is the format for first order logic with polymorphic types (à la ML), which is described in [3];
- SMT-LIB v2 format: this is the regular format for many-sorted first order logic that is intended to be used by SMT solvers;
- Alt-Ergo native format: this is the input format of Alt-Ergo that allows the user to use polymorphic types.

It should be noted that in these several archives, each file corresponds exactly to one PO, and that each file is standalone (it does not include any other file and contains the translation of the encoding of the B set theory in particular). It should also be noted that the encoding of the B set theory in Why3 is still work in progress, and that some POs may be not yet provable due to lack of axioms. Finally, in the case of SMT-LIB files, the selected logic is "UFNIA", which is the upper bound logic that deals with all the POs of this benchmark.

2 Support

You can contact us for any question via the email: David.Delahaye@cnam.fr.

References

[1] J.-R. Abrial. *The B-Book, Assigning Programs to Meanings*. Cambridge University Press, Cambridge (UK), 1996. ISBN 0521496195.

[2] C. Barrett, A. Stump, and C. Tinelli. The SMT-LIB Standard: Version 2.0. In *Satisfiability Modulo Theories (SMT)*, Edinburgh (Scotland), July 2010.

[3] J. C. Blanchette and A. Paskevich. TFF1: The TPTP Typed First-Order Form with Rank-1 Polymorphism. In *Conference on Automated Deduction (CADE)*, volume 7898 of *LNCS*, pages 414–420, Lake Placid (NY, USA), June 2013. Springer.

[4] F. Bobot, S. Conchon, É. Contejean, and S. Lescuyer. Implementing Polymorphism in SMT Solvers. In *Workshop on Satisfiability Modulo Theories (SMT)*, pages 1–5, Princeton, (NJ, USA), July 2008. ACM.

[5] 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.

[6] F. Bobot and A. Paskevich. Expressing Polymorphic Types in a Many-Sorted Language. In *Frontiers of Combining Systems (FroCoS)*, volume 6989 of *LNCS*, pages 87–102, Saarbrücken (Germany), Oct. 2011. Springer.

[7] ClearSy. *Atelier B 4.2*, Dec. 2014. http://www.atelierb.eu/.

[8] 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.

[9] D. Delahaye, C. Dubois, C. Marché, and D. Mentré. The BWare Project: Building a Proof Platform for the Automated Verification of B Proof Obligations. In *Abstract State Machines, Alloy, B, VDM, and Z (ABZ)*, volume 8477 of *LNCS*, pages 126–127, Toulouse (France), June 2014. Springer.

[10] 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*. Springer, 2012.

[11] 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.

[12] The BWare Project, 2012. http://bware.lri.fr/.