

TermComp 2016 Participant: NaTT*

Akihisa Yamada

University of Innsbruck, Austria

1 Overview

NaTT [4], standing for Nagoya Termination Tool, or New alps Termination Tool,¹ is a termination prover for TRSs, which is available at

<http://www.trs.cm.is.nagoya-u.ac.jp/NaTT/>

Supported categories are as follows:

TRS/SRS Standard: NaTT implements only basic components of the *dependency pair (DP) framework* [2], and its power is mostly due to the *weighted path order* [5], which provides and strengthens many previously known reduction pair techniques as its instances. This year's version has a finer analysis of the dependency graph and usable rules.

TRS/SRS Relative: Since the last year NaTT is capable of proving relative termination via the DP framework [3].

TRS Equational: This year it implements a new formalized AC-DP framework [6].

All the reduction pair constraints are encoded into incremental SMT problem scripts, which can be piped to any solver that complies the SMT-LIB 2.0 standard. The competition version of NaTT uses Z3 [1] as the back-end SMT solver.

NaTT is particularly fast, due to several efforts in SMT encoding. Since state-of-the-art SMT solvers including Z3 are still not so efficient on non-linear problems, NaTT transforms non-linear expressions into linear ones using a straightforward but effective *if-then-else* blasting [4]. It moreover utilizes the incremental feature of SMT solvers, and this year it has been further optimized: generating variables and constraints corresponding to a rewrite rule (or a DP) is delayed until the rewrite rule is involved in the considered DP problem.

References

- 1 Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3: an efficient SMT solver. In *TACAS 2008*, volume 4963 of *LNCS*, pages 337–340, 2008.
- 2 J. Giesl, R. Thiemann, and P. Schneider-Kamp. The dependency pair framework: Combining techniques for automated termination proofs. In *LPAR 2004*, volume 3452 of *LNAI*, pages 75–90, 2004.
- 3 J. Iborra, N. Nishida, G. Vidal, and A. Yamada. Relative termination via dependency pairs. *J. Autom. Reasoning*, 2016. Available online.
- 4 A. Yamada, K. Kusakari, and T. Sakabe. Nagoya Termination Tool. In *RTA-TLCA 2014*, volume 8560 of *LNCS*, pages 466–475, 2014.
- 5 A. Yamada, K. Kusakari, and T. Sakabe. A unified order for termination proving. *Sci. Comput. Program.*, 111:110–134, 2015.
- 6 Akihisa Yamada, Christian Sternagel, René Thiemann, and Keiichirou Kusakari. AC dependency pairs revisited. In *CSL 2016*, LIPIcs. To appear.

* This research was partly supported by the Austrian Science Fund (FWF) projects Y 757.

¹ Thanks to Masahiko Sakai for the original name, and Nao Hirokawa for the alternative.

