

# NaTT 2.2 in CoCo 2021

Akihisa Yamada

National Institute of Advanced Science and Technology

NaTT [4] is a termination prover for plain term rewriting. It is written in OCaml and the source code is available at:

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

Though it has nothing to do with proving confluence, NaTT implements a quick reachability check [3] for computing estimated dependency graphs [1]. To demonstrate the strength (or more precisely, weakness) of this reachability check, this year NaTT will participate in the “infeasibility” category of the Confluence Competition. Infeasibility means negated reachability, which can be tested by the aforementioned method. To meet the specification of the category, NaTT had to be modified to be able to

- expose the reachability checking function, and
- parse the COPS format for infeasibility.

An interesting task was the latter. To this end, the author incorporated a generic text-to-and-from-XML translator that he developed for another project, in order to translate the COPS format into a newly defined simple XML format for TRSs, which NaTT can understand. As a positive side effect, NaTT can now directly read the (complex) XML problem format of the Termination Competition [2]. A negative side effect is that the binary `bin/NaTT.exe` of version 2.2 does not read the old WST format anymore, but the script `bin/NaTT.sh` does.

At this point, it turned out that most of the infeasibility problems in COPS database are conditional TRSs. Therefore, the author had further to parse conditional rules. This was easy thanks to the above XML translator. However, as NaTT is for plain term rewriting, conditions are simply ignored. Thus it will only answer YES (unreachable) if unreachability could be proved without conditions, and will never answer NO (reachable).

## References

- [1] T. Arts and J. Giesl. Termination of term rewriting using dependency pairs. *Theor. Comput. Sci.*, 236(1–2):133–178, 2000.
- [2] Jürgen Giesl, Albert Rubio, Christian Sternagel, Johannes Waldmann, and Akihisa Yamada. The termination and complexity competition. In Dirk Beyer, Marieke Huisman, Fabrice Kordon, and Bernhard Steffen, editors, *TACAS 2019 (3)*, volume 11429 of *LNCS*, pages 156–166. Springer, 2019.
- [3] Christian Sternagel and Akihisa Yamada. Reachability analysis for termination and confluence of rewriting. In Tomás Vojnar and Lijun Zhang, editors, *TACAS 2019, Part I*, volume 11427 of *LNCS*, pages 262–278. Springer, 2019.
- [4] Akihisa Yamada, Keiichirou Kusakari, and Toshiki Sakabe. Nagoya Termination Tool. In Gilles Dowek, editor, *RTA-TLCA 2014*, volume 8560 of *LNCS*, pages 466–475. Springer, 2014.