

NaTT in CoCo 2022

Akihisa Yamada

National Institute of Advanced Industrial Science and Technology, Japan
akihisa.yamada@aist.go.jp

NaTT [3] is a termination prover for plain term rewriting, written in OCaml and the source code is available at:

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

Though NaTT is not a confluence prover, since the last year it is participating in the infeasibility category of the Confluence Competition, as its quick reachability checker [1] can be used to solve infeasibility problems.

In this year, NaTT solves infeasibility problems by generalizing its term ordering techniques, the core of termination proving. The details of the technique will be presented at IJCAR 2022 [2].

Since the last year, the input format of NaTT is a (relatively simple) XML, now described at the above page. In the competition, it reads the COPS format by translating it into the XML format, using the text-to-and-from-XML translator TXtruct, which was presented at WPTE 2022. The format description in the above page is automatically generated by TXtruct.

References

- [1] Christian Sternagel and Akihisa Yamada. Reachability analysis for termination and confluence of rewriting. In Tomás Vojnar and Lijun Zhang, editors, *TACAS (1) 2019*, volume 11427 of *LNCS*, pages 262–278. Springer, 2019. doi:10.1007/978-3-030-17462-0_15.
- [2] Akihisa Yamada. Term orderings for non-reachability of (conditional) rewriting. In *IJCAR 2022*, volume 13385 of *LNCS*, 2022. To appear.
- [3] 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. doi:10.1007/978-3-319-08918-8_32.