

# ACP: System Description for CoCo 2016

Takahito Aoto<sup>1</sup> and Yoshihito Toyama<sup>2</sup>

<sup>1</sup> Faculty of Engineering, Niigata University

`aoto@ie.niigata-u.ac.jp`

<sup>2</sup> RIEC, Tohoku University

`toyama@nue.riec.tohoku.ac.jp`

ACP is an automated confluence prover for term rewriting systems (TRSs) that has been developed in Toyama–Aoto group in RIEC, Tohoku University. ACP integrates multiple direct criteria for guaranteeing confluence of TRSs. It incorporates divide-and-conquer criteria by which confluence or non-confluence of TRSs can be inferred from those of their components. Several methods for disproving confluence are also employed. A list of implemented criteria and methods can be found on the website of ACP [1]. For a TRS to which direct confluence criteria do not apply, the prover decomposes it into components using divide-and-conquer criteria, and tries to apply direct confluence criteria to each component. Then the prover combines these results to infer the (non-)confluence of the whole system.

ACP is written in Standard ML of New Jersey (SML/NJ) and is provided as a heap image that can be loaded into SML/NJ runtime systems. It uses a SAT prover such as MiniSAT and an SMT prover YICES as external provers. It internally contains an automated (relative) termination prover for TRSs but external (relative) termination provers can be substituted optionally. The input TRS is specified in the (old) TPDB format. Users can specify criteria to be used so that each criterion or any combination of them can be tested. Several levels of verbosity are available for the output so that users can investigate details of the employed approximations for each criterion or can get only the final result of prover’s attempt. For some criteria, it supports generation of proofs in CPF format that can be certified by certifiers. The source code and a list of implemented criteria are found on the webpage [1].

The internal structure of the prover is kept simple and is mostly inherited from the version 0.11a, which has been described in [2]. No new (non-)confluence criterion has been incorporated from the one submitted for CoCo 2015.

## References

- [1] ACP (Automated Confluence Prover). <http://www.nue.riec.tohoku.ac.jp/tools/acp/>.
- [2] T. Aoto, J. Yoshida, and Y. Toyama. Proving confluence of term rewriting system automatically. In *Proc. of 20th RTA*, volume 5595 of *LNCS*, pages 93–102. Springer-Verlag, 2009.