

# ACP: System Description for CoCo 2026

Takahito Aoto

Faculty of Engineering, Niigata University  
aoto@ie.niigata-u.ac.jp

ACP (Automated Confluence Prover) is a tool for proving confluence and some related properties of (conditional) term rewriting systems. Below we provide a brief overview of ACP.

A primary functionality of ACP is proving confluence (CR) of term rewriting systems (TRSs). ACP integrates multiple direct criteria for guaranteeing confluence of TRSs. It also 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. For some criteria, it supports generation of proofs in CPF format that can be certified by certifiers. The internal structure of the prover is kept simple and is mostly inherited from the version 0.11a, which has been described in [3]. It also deal with confluence of oriented conditional term rewriting systems. Besides confluence, ACP supports proving the UNC property (unique normal form property w.r.t. conversion), the UNR property (unique normal form property w.r.t. reduction), and the commutation property of term rewriting systems [2, 4, 5, 6]. Some commutativity (dis)proving proofs have capability of producing certifiable proofs in CPF format.

ACP is written in Standard ML of New Jersey (SML/NJ) and the source code is also available from [1]. 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. 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 the prover's attempt.

## References

- [1] ACP (Automated Confluence Prover). <http://www.nue.ie.niigata-u.ac.jp/tools/acp/>.
- [2] T. Aoto and Y. Toyama. Automated proofs of unique normal forms w.r.t. conversion for term rewriting systems. In *Proc. of 12th FroCoS*, volume 11715 of *LNAI*, pages 330–347. Springer-Verlag, 2019.
- [3] 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.
- [4] T. Aoto. Proving uniqueness of normal forms w.r.t. reduction of term rewriting systems. In *Proc. of 34th LOPSTR*, volume 14919 of *LNCS*, pages 185–201. Springer-Verlag, 2024.
- [5] M. Yamaguchi and T. Aoto. A fast decision procedure for uniqueness of normal forms w.r.t. conversion of shallow term rewriting systems. In *Proc. of 5th FSCD*, volume 167 of *LIPICs*, pages 9:1–9:23. Schloss Dagstuhl, 2020.
- [6] J. Yoshida, T. Aoto, and Y. Toyama. Automating confluence check of term rewriting systems. *Computer Software*, 26(2):76–92, 2009.