

infChecker at the 2026 Confluence Competition*

Raúl Gutiérrez and Salvador Lucas

VRAIN, Universitat Politècnica de València, Valencia, Spain
raul.gutierrez@upv.es
slucas@dsic.upv.es

1 Overview

infChecker is a tool for checking *(in)feasibility* of sequences of rewrite relations with respect to *first-order theories*, called goals [7]. Input systems can be TRSs, CS-TRSs, CTRSs, CS-CTRSs y GTRSs [6]. infChecker participates in the INF category at the Confluence Competition but it is also used as an external tool in CONFident [1, 2], which participates in several categories in the Competition.

The tool is available here:

<http://zenon.dsic.upv.es/infChecker/>.

Some processors are mechanized using external tools like AGES [4], Prover9 and Mace4 [8]. Latest description of the tool can be found in [3, 5]. The set of relations accepted by infChecker are $\{\rightarrow, \rightarrow^*, \rightarrow^+, \leftrightarrow, \leftrightarrow^*, \leftrightarrow^+, \triangleright, \triangleright^+, \downarrow, \downarrow^+, \leftrightarrow, \leftrightarrow^*, \leftrightarrow^+, \leftrightarrow^*\}$ [5].

References

- [1] R. Gutiérrez. and S. Lucas On Proving Confluence of Generalized Term Rewriting Systems Using CONFident. *Proc. of the 15th International Workshop on Confluence, IWC'26*, to appear, 2026.
- [2] R. Gutiérrez. and S. Lucas Proving Confluence in the Confluence Framework with CONFident. *Fundamenta Informaticae*, 192, Issue 2: LOPSTR 2022, 2024.
- [3] R. Gutiérrez and S. Lucas. Automatically Proving and Disproving Feasibility Conditions. In N. Peltier and V. Sofronie-Stokkermans, editor, *Proc. of IJCAR'2020*, LNCS 12167:416–435. Springer, 2020.
- [4] R. Gutiérrez and S. Lucas. Automatic Generation of Logical Models with AGES. In *CADE 2019: Automated Deduction - CADE 27*, LNCS 11716:287:299. Springer, 2019.
- [5] R. Gutiérrez and S. Lucas. Proving and disproving feasibility with infChecker. In *Proc. of the 14th International Workshop on Confluence, IWC'25*, 38, 2025.
- [6] S. Lucas. Local confluence of conditional and generalized term rewriting systems. *Journal of Logical and Algebraic Methods in Programming*, 136, paper 100926, pages 1-23, 2024.
- [7] S. Lucas and R. Gutiérrez. Use of Logical Models for Proving Infeasibility in Term Rewriting. *Information Processing Letters*, 136:90–95, 2018.
- [8] W. McCune. Prover9 and Mace4. [online]. Available at <https://www.cs.unm.edu/~mccune/mace4/>.

*Partially supported by MCIN/AEI project PID2024-162030OB-100 funded by MCIN/AEI/10.13039/501100011033 and by “ERDF A way of making Europe” and by the grant CIPROM/2022/6 funded by Generalitat Valenciana.