CONFident at the 2021 Confluence Competition*

Miguel Vítores[2], Raúl Gutiérrez[1], and Salvador Lucas[2]

[1] Universidad Politécnica de Madrid, Madrid, Spain
r.gutierrez@upm.es
[2] VRAIN, Universitat Politècnica de València, Valencia, Spain
mvitvic@posgrado.upv.es
slucas@dsic.upv.es

1 Overview

CONFident 1.0 is a tool for checking the *confluence* or *non-confluence* of systems based on rewriting by means of its logical representation. The tool is available here: http://zenon.dsic.upv.es/confident/. It is written in Haskell following a DP-framework structure, i.e., by defining problems and processors:

- problems are tuples $\tau = (\mathbb{T}, \mathcal{G})$, where $\mathbb{T}$ is a $\mathbb{P}$-indexed theory, $\mathbb{P}$ is a set of predicates and $\mathcal{G}$ are logical goals representing conditions to be checked during the analysis: joinability, reachability, feasibility, etc. We can use predicate symbols as $\to$ and $\to^*$ to represent different kind of relations between terms defined by a logical theory. We say that τ is confluent if $\mathcal{G}$ is $\mathbb{T}$-confluent; otherwise, it is non-confluent.

- processors are defined as partial functions from problems into set of (hopefully simpler) problems. Our processors are based on transforming problems of confluence into logical problems that can be solved by external tools (infeasibility checkers, theorem provers, model generators...).

We implement these processors using the logical approach presented in [1, 3, 4] and mechanizing them by external tools like MU-TERM [3], infChecker [1], AGES [2], Prover9 and Mace4 [6] and Barcelogic[1].

References

[1] 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.

[2] 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.

[3] R. Gutiérrez and S. Lucas. MU-TERM: Verify Termination Properties Automatically (System Description). In N. Peltier and V. Sofronie-Stokkermans, editor, *Proc. of IJCAR'2020*, LNCS 12167:436–447. Springer, 2020.

[4] S. Lucas. Proving semantic properties as first-order satisfiability. *Artificial Intelligence* 277, paper 103174, 24 pages, 2019.

[5] S. Lucas and R. Gutiérrez. Use of Logical Models for Proving Infeasibility in Term Rewriting. *Information Processing Letters*, 136:90–95, 2018.

[6] W. McCune. Prover9 and Mace4. [online]. Available at https://www.cs.unm.edu/~mccune/mace4/.

*Partially supported by the EU (FEDER) and the Spanish MCIU under grant RTI2018-094403-B-C32 and by the Spanish Generalitat Valenciana under grant PROMETEO/2019/098.

[1] https://barcelogic.com/