infChecker at the 2020 Confluence Competition*

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
slucas@dsic.upv.es

1 Overview

infChecker 1.0 is a tool for checking *(in)feasibility* of goals $\mathcal{G} = \{F_i\}_{i=1}^m$ where $F_i = (s_{ij} \bowtie_{ij} t_{ij})_{i=1}^{n_i}$ and $\bowtie_{ij} \in \{\to, \to^*, \to^+, \hookrightarrow, \hookrightarrow^*, \hookrightarrow^+, \rhd, \unrhd, \downarrow, \downarrow_{\hookrightarrow}, \leftrightarrow, \hookleftarrow\!\!\!\!\hookrightarrow, \leftrightarrow^*, \hookleftarrow\!\!\!\!\hookrightarrow^*\}$ where predicates $\bowtie_{ij}$ represent binary relations on terms (most of them well-known or easy generalizations of well-known relations) defined by provability of goals $s \bowtie_{ij} t$ with respect to a *first-order theories* $\mathsf{Th}_{\bowtie_{ij}}$ [2, 4]. The tool is available here: http://zenon.dsic.upv.es/infChecker/. It is written in Haskell and provides a first implementation of the *Feasibility Framework* [2], where three *processors* have been implemented:

- $\mathsf{P}^{\mathsf{Sat}}$ integrates the satisfiability approach described in [3] to prove infeasibility. In infChecker, we use the model generators AGES [1] and Mace4 [6] to find a proof.
- $\mathsf{P}^{\mathsf{Prov}}$ integrates the logic-based approach to program analysis described in [3] to prove feasibility by theorem proving. In infChecker, we use the theorem prover Prover9 [6].
- P^{NC} adapt the processor that narrow conditions in the 2D DP framework for proving operational termination of CTRs [5] to be used with feasibility sequences.

Our proof strategy is: (1) first, we try to prove feasibility using $\mathsf{P}^{\mathsf{Prov}}$; (2) if $\mathsf{P}^{\mathsf{Prov}}$ fails, we apply $\mathsf{P}^{\mathsf{Sat}}$; (3) if $\mathsf{P}^{\mathsf{Sat}}$ fails, we apply P^{NC}; (4) if P^{NC} succeeds and modifies the feasibility sequence, we go to (2), otherwise we return MAYBE.

References

[1] 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.

[2] R. Gutiérrez and S. Lucas. Automatically Proving and Disproving Feasibility Conditions. In *Proc. of IJCAR'2020*, LNCS to appear. Springer, 2020.

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

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

[5] S. Lucas, J. Meseguer, and R. Gutiérrez. The 2D Dependency Pair Framework for conditional rewrite systems. Part I: Definition and basic processors. *Journal of Computer and System Sciences*, 96:74–106, 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.