EPiC Series in Computing

Volume 108, 2025, Pages 190–194

Proceedings of 12th Int. Workshop on Applied Verification for Continuous and Hybrid Systems

ARCH-COMP'25 Repeatability Evaluation Report

Tobias Ladner

Technical University of Munich, Germany
tobias.ladner@tum.de

Abstract

The repeatability evaluation for the 9th International Competition on Verifying Continuous and Hybrid Systems (ARCH-COMP'25) is summarized in this report. The competition was held as part of the Applied Verification for Continuous and Hybrid Systems (ARCH) workshop in 2025. In its 9th edition, participants submitted their tools via an automated evaluation system developed over recent years. Each submission includes a Dockerfile and the necessary scripts for running the tool, enabling consistent execution in a containerized environment with all dependencies preinstalled. This setup improves comparability by running all tools on the same hardware. Submissions and results are automatically synchronized with a Git repository for repeatability evaluation and long-term archiving. We plan to further extend the evaluation system by refining the submission pipeline, aiming to enable automated evaluation across all competition categories.

1 Introduction

This report summarizes the status of this year's ARCH[1] friendly competition and provides an overview of the participating tools in each category. ARCH-COMP aims to improve the repeatability of code produced by researchers and industry partners by archiving the self-contained submissions, which are tested on an independent platform. This is in strong contrast to the usual practice, where code is often either not available at all or can barely reproduce the results: While there are many platforms to submit code along with a paper, there is, unfortunately, only little incentive to do so after a paper is accepted. Thus, the code might stay within the research group and if not developed further, certain aspects break over time such as dependencies of used libraries, making it impossible to reproduce the results – even for the original authors of the paper. Our automated evaluation system[2] prevents this issue as the code is run on independent hardware, where all dependencies have to be specified for the code to be runable. As all tools are run on the same hardware, the comparability of the results, such as computation time, is improved as well. The code and the obtained results are archived in a Git repository[3], categorized by year, category, and participated tool. Each tool folder (`/<year>/<category>/<tool>`) contains a script that runs the entire evaluation with a single click. The results obtained via the evaluation system are archived within `./results_ref` as reference.

[1] Applied Verification for Continuous and Hybrid Systems (ARCH): https://cps-vo.org/group/ARCH
[2] Evaluation System of ARCH-COMP: https://arch.repeatability.cps.cit.tum.de
[3] Archive: https://gitlab.com/goranf/ARCH-COMP

The remainder of this report is structured as follows: In Sec. 2, we provide a high-level overview of the repeatability evaluation, broken down by category and listing the participating tools. In Sec. 3, we present details about this year's evaluation, including improvements over the previous year. Finally, in Sec. 4, we conclude the report and outline our goals for next year's ARCH-COMP.

2 Repeatability Evaluation Overview

The repeatability evaluation of the competition featured seven categories and eleven software tools, where several tools participated in multiple categories, but have been counted distinctly for their participation in each category. While the introduction of the automatic evaluation system has led to an overall improvement in the repeatability evaluation, not all categories have participated in the automatic evaluation yet. The categories that tools participated in the repeatability evaluation are:

- AFF: affine and piecewise affine dynamics (2 tools),
- AINNCS: artificial intelligence and neural network control systems (5 tools),
- FALS: falsification (no tools participating in the repeatability evaluation),
- HSTP: hybrid systems theorem proving (1 tool),
- NLN: nonlinear dynamics (3 tools),
- PCDB: piecewise constant dynamics and bounded model checking (no tools participating in the repeatability evaluation), and
- SM: stochastic models (no tools participating in the repeatability evaluation).

For the categories that have tools that participated in the repeatability evaluation, the tools evaluated, broken into their competition categories and alphabetically sorted, are:

- AFF:
  - CORA [1], and
  - JuliaReach [2];
- AINNCS:
  - CORA [1],
  - CROWN-Reach [9],
  - immrax [4],
  - JuliaReach [2], and
  - NNV [7];
- NLN:
  - CORA [1],
  - Dynibex [3], and
  - JuliaReach [2];

- HSTP:
  - HHLPy [8].

All of the tools listed above were deemed repeatable based on the evaluation, as detailed in the next section. The repeatability package for each tool in its respective category is available in the ARCH competition archive[4].

3 Repeatability Evaluation Details

The procedure for the repeatability evaluation builds on previous iterations of ARCH-COMP first held in 2017 [5], and the automatic evaluation system, which was introduced in 2023 [6]. Users can submit their tools as a zip file via the website of the evaluation system[5]. This zip file is then moved to a worker server (see Appendix A for specifications), where the submission gets executed in a containerized environment using Docker. To make the submission reproducible, the dependencies and setup of the tool are specified in the respective Dockerfile by the tool authors. Users are also required to save the verification results and plots in a results folder, which then gets extracted and is archived with the submission itself[6]. The verification results should be stored in a standardized csv file for displaying them on the website and easy usage for the report. Users also have the option to double-check the results before they get published to fix any bugs that might occur while running the code within the evaluation system. To help users with finding bugs, a log file is provided of the entire submission run. As the automatic evaluation system gives immediate feedback on the reproducibility of the tools in both, in case the repeatability fails and the benchmark results and times of their submission, tool authors can revise their submissions as desired. Thus, the effort the authors put into the competition is also valued in the repeatability evaluation.

In this year's iteration, we also improved how authors can extract results for the individual reports. All results can now be exported via the submission website[7], with customization options for table formats and visible columns. Additionally, we provide a tool that automatically converts figures into LaTeX TikZ/PGF format[8], enabling consistent, high-quality vector graphics in reports.

In ARCH-COMP, users usually know all the specifications of each benchmark in advance, and they can fine-tune their tools to obtain optimal results. However, this expert knowledge is not available when the tool is used externally, and default settings are usually chosen. As in last year's iteration of ARCH-COMP, we also experimented in the AFF category by having secret benchmarks that are not known to the participants of this category to avoid fine-tuning their tools. The specifications of the secret benchmarks are only added prior to the evaluation of the submission in a standardized format. Details of this format are given in the AFF category report published alongside this report.

[4] Archive: https://gitlab.com/goranf/ARCH-COMP
[5] Evaluation System of ARCH-COMP: https://arch.repeatability.cps.cit.tum.de
[6] Archive: https://gitlab.com/goranf/ARCH-COMP
[7] Details on results export and TikZ figure generation: https://arch.repeatability.cps.cit.tum.de/arch-report/information
[8] LaTeX TikZ/PGF: https://tikz.dev/

4 Conclusion and Outlook

This report summarizes the repeatability evaluation for the 9th edition of the competition on formal verification of continuous and hybrid systems (ARCH-COMP'25), held as part of the ARCH'25 workshop. Detailed category reports are available in the proceedings[9] and on the ARCH website[10]. All documentation, benchmarks, and execution scripts related to the repeatability evaluation are archived online. Looking forward, we aim to include every ARCH-COMP category in the automated evaluation system to ensure that all results are fully reproducible. We are also refining the system by addressing issues identified during this year's run. Overall, the automated evaluation system has streamlined the process and contributes significantly to the community by enabling one-click reproducibility.

Acknowledgments

The author gratefully acknowledges financial support from the project FAI funded by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under project number 286525601. Special thanks go to Volkan Şahin for his significant contributions to the development and maintenance of the evaluation system's website.

A Hardware Specification

This year, we run all tools on the same hardware using tool-specific docker images. The specification of the server used for the evaluation is given below:

- Processor: AMD EPYC 7742 64-Core
- Memory: 995 GB
- OS: Ubuntu 22.04
- Docker: 20.10.21

References

[1] M. Althoff. An introduction to CORA 2015. In *Proceedings of the Workshop on Applied Verification for Continuous and Hybrid Systems*, pages 120–151, 2015.

[2] Sergiy Bogomolov, Marcelo Forets, Goran Frehse, Kostiantyn Potomkin, and Christian Schilling. JuliaReach: A toolbox for set-based reachability. In *Proceedings of the ACM International Conference on Hybrid Systems: Computation and Control*, pages 39–44, 2019.

[3] Julien Alexandre dit Sandretto and Alexandre Chapoutot. Validated explicit and implicit Runge–Kutta methods. *Reliable Computing*, 22(1):79–103, 2016.

[4] Akash Harapanahalli, Saber Jafarpour, and Samuel Coogan. immrax: A parallelizable and differentiable toolbox for interval analysis and mixed monotone reachability in JAX. *IFAC-PapersOnLine*, 58(11):75–80, 2024.

[5] Taylor T. Johnson. ARCH-COMP17 repeatability evaluation report. In *ARCH@ CPSWeek*, pages 175–180, 2017.

[9] ARCH proceedings: https://cps-vo.org/group/ARCH/proceedings
[10] ARCH website: https://cps-vo.org/group/ARCH

[6] Taylor T. Johnson. ARCH-COMP23 repeatability evaluation report. In *Proceedings of 10th International Workshop on Applied Verification of Continuous and Hybrid Systems*, volume 96, pages 189–195, 2023.

[7] Diego Manzanas Lopez, Sung Woo Choi, Hoang-Dung Tran, and Taylor T. Johnson. NNV 2.0: The neural network verification tool. In *35th International Conference on Computer-Aided Verification*, July 2023.

[8] Huanhuan Sheng, Alexander Bentkamp, and Bohua Zhan. HHLPy: practical verification of hybrid systems using hoare logic. In *International Symposium on Formal Methods*, pages 160–178, 2023.

[9] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems*, 31, 2018.