Automata Learning with an Incomplete Teacher (Artifact)

Mark Moeller
Cornell University, Ithaca, NY, USA

Thomas Wiener
Cornell University, Ithaca, NY, USA

Alaia Solko-Breslin
University of Pennsylvania, Philadelphia, PA, USA

Caleb Koch
Stanford University, Stanford, CA, USA

Nate Foster
Cornell University, Ithaca, NY, USA

Alexandra Silva
Cornell University, Ithaca, NY, USA

Abstract

We provide an implementation of the automata learning software described in the associated ECOOP article. In particular, the artifact is a Docker image with the source code for nerode and `nerode-learn`, along with the scripts and benchmark inputs needed to reproduce the experiments described in the paper.

2012 ACM Subject Classification Theory of computation → Active learning

Keywords and phrases Finite Automata, Active Learning, SMT Solvers

Digital Object Identifier 10.4230/DARTS.9.2.21

Funding This material is based upon work supported by the Office of Naval Research under Contract No. N68335-22-C-0411.

Acknowledgements We thank Marijn Heule, Martin Leucker, and Arlindo Oliveira for their efforts in providing us access to their code and benchmarks. We also thank Akshat Singh and Sheetal Athrey, with whom this project began as an undergraduate research project.

Related Article Mark Moeller, Thomas Wiener, Alaia Solko-Breslin, Caleb Koch, Nate Foster, and Alexandra Silva, "Automata Learning with an Incomplete Teacher", in 37th European Conference on Object-Oriented Programming (ECOOP 2023), LIPIcs, Vol. 263, pp. 21:1–21:30, 2023.
`https://doi.org/10.4230/LIPIcs.ECOOP.2023.21`

Related Conference 37th European Conference on Object-Oriented Programming (ECOOP 2023), July 17–21, 2023, Seattle, Washington, United States

Evaluation Policy The artifact has been evaluated as described in the ECOOP 2023 Call for Artifacts and the ACM Artifact Review and Badging Policy.

1 Scope

This artifact contains the OCaml implementation described in Section 8 of the associated ECOOP paper. The Docker image includes everything required to build the software and reproduce the experiments in the evaluation of Section 9 of the paper.

Dagstuhl Artifacts Series
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,
Dagstuhl Publishing, Germany

2 Getting Started

After downloading the artifact image, in order to start it, run:

```
$ docker load < imat-artifact-image.tar.gz
$ docker run -it imat/base
```

At this point, there is a README in the current working directory of the running Docker image that will guide a user in the basics of the software and reproducing the results of the paper.

3 Content

The artifact package includes:

- Source code for `nerode`, a package for various constructions of finite automata and regular expressions.
- Source code for `nerode-learn`, a package for implementing learning algorithms for automata, including the learner described in the paper.
- A package for interacting with the Z3 SMT solver [1] from an OCaml program.
- The dependencies required for compiling and running the above.
- Benchmarks (i.e., example inputs) due to Lee, So, and Oh [2], and Oliveira and Silva [3].

4 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the current version of the source code for `nerode` and `nerode-learn` is also available on GitHub at: `https://github.com/cornell-pl/nerode-public`.

5 Tested platforms

There are no special system requirements beyond those required to run Docker. The uncompressed Docker image is about 3.36 GiB. The artifact was tested using Docker version 20.10.21. The other relevant dependencies are contained in the image. The dependencies for the source itself are maintained current in the `opam` files for each library in the public GitHub repository.

6 License

Both `nerode` and `nerode-learn` are available under the Apache v2.0 license.

7 MD5 sum of the artifact

4365265d0d7390aa915d8aee84cdd1cc

8 Size of the artifact

1.47 GiB

References

1 Leonardo Mendonça de Moura and Nikolaj S. Bjørner. Z3: an efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008. doi:10.1007/978-3-540-78800-3_24.

2 Mina Lee, Sunbeom So, and Hakjoo Oh. Synthesizing regular expressions from examples for introductory automata assignments. In *Proceedings of the 2016 ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences*, GPCE 2016, pages 70–80, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2993236.2993244.

3 Arlindo Oliveira and J.P.M. Silva. Efficient algorithms for the inference of minimum size dfas. *Machine Learning*, 44, July 2001.