

LearnLib Tutorial: From Finite Automata to Register Interface Programs

Falk Howar¹, Malte Isberner², Maik Merten², and Bernhard Steffen²

¹ Carnegie Mellon University, Silicon Valley Campus,
Mountain View, CA

`falk.howar@tu-dortmund.de`

² Technical University Dortmund, Chair for Programming Systems, Dortmund,
D-44227, Germany

`{malte.isberner|maik.merten|steffen}@cs.tu-dortmund.de`

1 Motivation

In the past decade, active automata learning, an originally merely theoretical enterprise, got attention as a method for dealing with black-box or third party systems. Applications ranged from the support of formal verification, e.g. for assume guarantee reasoning [4], to usage of learned models as the basis for regression testing. In the meantime, a number of approaches exploiting active learning for validation [17,20,6,7,2,1] emerged.

Today, active automata learning is on the verge of becoming a valuable asset in bringing formal methods to systems lacking formal descriptions (e.g., the huge class of legacy systems): This edition of ISoLA alone features a track on active learning in formal verification [16], one on model-based testing and model inference [12], this tutorial, and is co-located with the STRESS summer school,³ where active automata learning is part of the curriculum.

In particular when dealing with black-box systems, i.e., systems that can be observed, but for which no or little knowledge about the internal structure or even their intent is available, *active* automata learning can be considered as a key technology due to its test-based approach to model inference. However, the test-based interaction introduces a number of challenges when using active automata learning to infer models of real world systems, which have been summarized in [21]:

A: Interacting with real systems

The interaction with a realistic target system comes with two problems. The technical problem of establishing an adequate interface that allows one to apply test cases for realizing so-called membership queries, and a conceptual problem of bridging the gap between the abstract learned model and the concrete runtime scenario.

B: Membership Queries

³ <http://info.santoslabor.org/event/stress2012>

Whereas small learning experiments typically require only a few hundred membership queries, learning realistic systems may easily require several orders of magnitude more.

C: Reset

Active learning requires membership queries to be independent. Whereas this is no problem for simulated system, this may be quite problematic in practice.

D: Parameters and value domains

Active learning classically is based on abstract communication alphabets. Parameters and interpreted values are only treated to an extent expressible within the abstract alphabet. In practice, this typically is not sufficient, not even for systems as simple as communication protocols, where, e.g., increasing sequence numbers must be handled, or where authentication requires matching user/password combinations.

E: Equivalence Queries

Equivalence queries compare a learned hypothesis model with the target system for language equivalence and, in case of failure, return a counterexample exposing a difference. In practice, equivalence queries will have to be approximated using membership queries. Methods from conformance testing have been suggested as approximations but are in general too expensive to be feasible for industry scale applications.

The tutorial discusses all these issues along a number of practical examples. In particular, using the LearnLib [19,18,15,13], a flexible automata learning framework, it provides hands-on experience on

- Challenge A: It is discussed how test drivers can be created for the LearnLib. Starting with the construction of application-specific test drivers by hand, it is discussed how a generic test driver can be employed by means of configuration. This configuration can be (semi-)automatically created by analysis of the target system's interface [14].
- Challenge E: Here we illustrate how a more global view of the learning process that more closely coalesces the two intertwined learning phases of model construction and model validation improves both, intuition and efficiency. This approach was applied successfully as part of the winning contribution of the ZULU competition [5,9] where it clearly dominated other methods of searching for counterexamples.
- Challenge D: Here two approaches are discussed and applied [22], alphabet abstraction refinement [10], and the explicit treatment of data flow using the more powerful modeling format of register automata [3,8,22].

Participants are invited to experience the impact of all these methods on their own laptop using their own LearnLib installation.⁴

⁴ <http://www.learnlib.de>

References

1. Fides Aarts, Julien Schmaltz, and Frits W. Vaandrager. Inference and Abstraction of the Biometric Passport. In Margaria and Steffen [11], pages 673–686.
2. Therese Bohlin and Bengt Jonsson. Regular Inference for Communication Protocol Entities. Technical report, Department of Information Technology, Uppsala University, Sweden, 2009.
3. S. Cassel, F. Howar, B. Jonsson, M. Merten, and B. Steffen. A Succinct Canonical Register Automaton Model. In *ATVA*, volume 6996 of *Lecture Notes in Computer Science*, pages 366–380. Springer Verlag, 2011.
4. Jamieson M. Cobleigh, Dimitra Giannakopoulou, and Corina S. Pasareanu. Learning Assumptions for Compositional Verification. In *Proc. TACAS '03, 9th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, volume 2619 of *Lecture Notes in Computer Science*, pages 331–346. Springer Verlag, 2003.
5. David Combe, Colin de la Higuera, and Jean-Christophe Janodet. Zulu: An Interactive Learning Competition. In Anssi Yli-Jyrä, András Kornai, Jacques Sakarovitch, and Bruce W. Watson, editors, *FSMNLP*, volume 6062 of *Lecture Notes in Computer Science*, pages 139–146. Springer, 2009.
6. Andreas Hagerer, Hardi Hungar, Oliver Niese, and Bernhard Steffen. Model Generation by Moderated Regular Extrapolation. *Lecture Notes in Computer Science*, pages 80–95, 2002.
7. Andreas Hagerer, Tiziana Margaria, Oliver Niese, Bernhard Steffen, Georg Brune, and Hans-Dieter Ide. Efficient Regression Testing of CTI-Systems: Testing a Complex Call-center Solution. *Annual Review of Communication, Int.Engineering Consortium (IEC)*, 55:1033–1040, 2001.
8. Falk Howar, Bernhard Steffen, Bengt Jonsson, and Sofia Cassel. Inferring Canonical Register Automata. In *VMCAI 2012*, to appear.
9. Falk Howar, Bernhard Steffen, and Maik Merten. From ZULU to RERS – Lessons Learned in the ZULU Challenge. In Margaria and Steffen [11], pages 687–704.
10. Falk Howar, Bernhard Steffen, and Maik Merten. Automata Learning with Automated Alphabet Abstraction Refinement. In *Twelfth International Conference on Verification, Model Checking, and Abstract Interpretation*, 2011.
11. Tiziana Margaria and Bernhard Steffen, editors. *Leveraging Applications of Formal Methods, Verification, and Validation - 4th International Symposium on Leveraging Applications, ISoLA 2010, Heraklion, Crete, Greece, October 18-21, 2010, Proceedings, Part I*, volume 6415 of *Lecture Notes in Computer Science*. Springer, 2010.
12. Karl Meinke and Neil Walkinshaw. Model-based Testing and Model Inference. In *ISoLA*, 2012.
13. Maik Merten, Falk Howar, Bernhard Steffen, Sofia Cassel, and Bengt Jonsson. Demonstrating Learning of Register Automata. In Cormac Flanagan and Barbara König, editors, *TACAS*, volume 7214 of *Lecture Notes in Computer Science*, pages 466–471. Springer, 2012.
14. Maik Merten, Malte Isberner, Falk Howar, Bernhard Steffen, and Tiziana Margaria. Automated Learning Setups in Automata Learning. In *ISoLA*, 2012.
15. Maik Merten, Bernhard Steffen, Falk Howar, and Tiziana Margaria. Next Generation LearnLib. In *17th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2011*, 2011.
16. Corinna Pasareanu and Mihaela Bobaru. Learning Techniques for Software Verification and Validation. In *ISoLA*, 2012.

17. D. Peled, M. Y. Vardi, and M. Yannakakis. Black Box Checking. *Journal of Automata, Languages and Combinatorics*, 7(2):225–246, 2002.
18. Harald Raffelt and Bernhard Steffen. LearnLib: A Library for Automata Learning and Experimentation. In Luciano Baresi and Reiko Heckel, editors, *FASE*, volume 3922 of *Lecture Notes in Computer Science*, pages 377–380. Springer, 2006.
19. Harald Raffelt, Bernhard Steffen, Therese Berg, and Tiziana Margaria. LearnLib: A Framework for Extrapolating Behavioral Models. *Int. J. Softw. Tools Technol. Transf.*, 11(5):393–407, 2009.
20. Muzammil Shahbaz, Keqin Li, and Roland Groz. Learning Parameterized State Machine Model for Integration Testing. In *Proc. 31st Annual Int. Computer Software and Applications Conf.*, volume 2, pages 755–760, Washington, DC, USA, 2007. IEEE Computer Society.
21. B. Steffen, F. Howar, and M. Merten. Introduction to Active Automata Learning from a Practical Perspective. In *SFM*, volume 6659 of *Lecture Notes in Computer Science*, pages 256–296. Springer, 2011.
22. Bernhard Steffen, Falk Howar, and Malte Isberner. Active Automata Learning: From DFAs to Interface Programs and Beyond. In *ICGI*, 2012.