Learning Systems: Machine-Learning in Software Products and Learning-Based Analysis of Software Systems

Special Track at ISoLA 2016

Falk Howar[1(✉)], Karl Meinke[2], and Andreas Rausch[1]

[1] Clausthal University of Technology, Clausthal-Zellerfeld, Germany
falk.howar@tu-clausthal.de

[2] KTH Royal Institute of Technology, Stockholm, Sweden

We are entering the age of learning systems! On the one hand, we are surrounded by devices that learn from our behavior [3]: household appliances, smart phones, wearables, cars, etc.—the most recent prominent example being Tesla Motor's autopilot that learns from human drivers. On the other hand, man-made systems are becoming ever more complex, requiring us to learn the behavior of these systems: Learning-based testing [8,13,17], e.g., has been proposed as a method for testing the behavior of systems systematically without models and at a high level of abstraction. Promising results have been obtained here using active automata learning technology in verification [6,16] and testing [1,8]. At the same time, active automata learning has been extended to support the inference of program structures [5,10] (it was first introduced for regular languages).

Advances in both areas raise the same questions cornering properties of inferred models: *How accurate are the descriptions that can be obtained of some behavior?* and: *How can we reason about and assure the safety of such systems?* This track aims at bringing together practitioners and researchers to explore the practical impact and challenges associated with using learning-based products as well as learning-based approaches in the analysis and verification of software. The track continues a series of special tracks focused on the application of automata learning techniques in testing and verification at ISoLA conferences [7,15] and [9]. This year's special track has three contributions and two tutorials.

The first contribution *"ALEX: Mixed-Mode Learning of Web Applications at Ease"* by Alexander Bainczyk, Malte Isberner, Tiziana Margaria, Johannes Neubauer, Alexander Schieweck, and Bernhard Steffen [2] (in this volume) presents ALEX, a Browser-based tool that enables non-programmers to fully automatically infer models of other Web applications via active automata learning. These models can be used for documentation, testing, and verification of such applications. ALEX guides a user in setting up dedicated learning scenarios, and invites her to experiment with the available options in order to infer models at adequate levels of abstraction. Characteristic for ALEX is its support for mixed-mode learning: Rest and Web services can be executed simultaneously in one learning experiment, which is ideal when trying to compare back-end and front-end functionality of a Web application. The authors present results from

T. Margaria and B. Steffen (Eds.): ISoLA 2016, Part II, LNCS 9953, pp. 651–654, 2016.
DOI: 10.1007/978-3-319-47169-3_50

an evaluation of ALEX in a comparative study with 140 undergraduate students. The contribution documents recent advances in the usability of learning-based analysis tools.

The second contribution *"Assuring the Safety of Advanced Driver Assistance Systems through a Combination of Simulation and Runtime Monitoring"* by Malte Mauritz, Falk Howar and Andreas Rausch [12] (in this volume) addresses one of the open challenges in the domain of autonomous driving: the lack of established and cost-efficient approaches for assuring the safety of advanced driver assistance systems. The authors present a method for ensuring that an advanced driver assistance system satisfies its safety requirements at runtime and operates within safe limits that were tested in simulations. This can be the basis for reducing the cost of quality assurance by transferring a significant part of the testing effort from road tests to (system-level) simulations. The approach utilizes runtime monitors that are generated from safety requirements and trained (i.e., learned) using simulated test cases. The contribution shows that relevant driving scenarios can be learned from data recorded in road tests. It presents an interesting usecase for data analysis and learning in the realm of safety assurance.

The third contribution *"Enhancement of an adaptive HEV operating strategy using machine learning algorithms"* by Mark Schudeleit, Meng Zhang, Xiaofei Qi, Ferit Küçükay, and Andreas Rausch [18] (in this volume) presents two approaches for reducing CO_2 emissions of a hybrid electric vehicle. The first approach is an adaptive heuristic operating strategy. It classifies current driving style and driving environment into predefined categories and chooses a corresponding strategy for switching between combustion and electric engine. The second approach optimizes this adaptive operating strategy for individual drivers using multigene symbolic regression and supervised machine learning. This contribution demonstrates the potential positive impact of learning products that can be trained to work optimally with their respective user.

The first tutorial *"Learning-based Testing of Procedural and Reactive Systems"* will be given by Karl Meinke from KTH Royal Institute of Technology in Sweden. Learning-based testing (LBT) is an emerging paradigm for black-box requirements testing that is based on combining machine learning with model checking [8,13,17]. The basic idea is to incrementally reverse engineer an abstract model of a system under test (SUT) by using machine learning techniques applied to black-box test cases and their results. Test verdict generation (pass/fail/warning) is fully automatic, based on a simple equality test. So a high degree of test automation is achieved. In practice many thousands of test cases per hour can be executed, with greater effectiveness than random testing. LBT is a general paradigm that can be applied to any class of software systems for which there exist efficient machine learning and model checking algorithms. We can illustrate this generality with research on testing: (1) imperative "C"-style programs against Hoare style pre- and postconditions, (2) reactive systems, based on automata learning algorithms and temporal logic model checkers, and (3) hybrid automata, based on combining methods from (1) and (2). The tutorial will address practical aspects of the methodology using the tool LBTest [14].

The second tutorial *"Active Automata Learning with LearnLib"* will be given by Falk Howar from Clausthal University of Technology in Germany. The tutorial will provide an introduction to active learning of Mealy machines, an automata model particularly suited for modeling the behavior of realistic reactive systems. Active learning is characterized by its alternation of an exploration phase and a testing phase. During exploration phases, so-called membership queries are used to construct hypothesis models of a system under learning. In testing phases, so-called equivalence queries are used to compare respective hypothesis models to the actual system. These two phases are iterated until a valid model of the target system is produced. The tutorial will demonstrate this simple algorithmic pattern using LearnLib [11] and its extension to register automata [4]. It will also address the underlying correctness argument, its limitations, and, in particular, methods to overcome apparent hurdles for practical application. This comprises ways to address real world applications, as well as the treatment of infinite data domains by abstraction refinement.

References

1. Aarts, F., Kuppens, H., Tretmans, G.J., Vaandrager, F.W., Verwer, S.: Learning and testing the bounded retransmission protocol. In: Heinz, J., de la Higuera, C., Oates, T. (eds.) Proceedings of 11th International Conference on Grammatical Inference (ICGI 2012), 5–8 September 2012. JMLR Workshop and Conference Proceedings, vol. 21. pp. 4–18. University of Maryland, College Park (2012)
2. Bainczyk, A., Isberner, M., Margaria, T., Neubauer, J., Schieweck, A., Steffen, B.: ALEX: mixed-mode learning of web applications at ease. In: ISoLA 2016 (2016)
3. Bosch, J., Olsson, H.H.: Data-driven continuous evolution of smart systems. In: Proceedings of the 11th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2016, pp. 28–34. ACM, New York (2016)
4. Cassel, S., Howar, F., Jonsson, B.: RALib: a LearnLib extension for inferring EFSMs. In: DIFTS 2015 at FMCAD 2015 (2015) (published online)
5. Cassel, S., Howar, F., Jonsson, B., Steffen, B.: Learning extended finite state machines. In: Giannakopoulou, D., Salaün, G. (eds.) SEFM 2014. LNCS, vol. 8702, pp. 250–264. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10431-7_18
6. Cobleigh, J.M., Giannakopoulou, D., Păsăreanu, C.S.: Learning assumptions for compositional verification. In: Garavel, H., Hatcliff, J. (eds.) TACAS 2003. LNCS, vol. 2619, pp. 331–346. Springer, Heidelberg (2003). doi:10.1007/3-540-36577-X_24
7. Giannakopoulou, D., Păsăreanu, C.S.: Learning techniques for software verification and validation – special track at ISoLA 2010. In: Margaria, T., Steffen, B. (eds.) ISoLA 2010, Part I. LNCS, vol. 6415, pp. 640–642. Springer, Heidelberg (2010). doi:10.1007/978-3-642-16558-0_51
8. Hagerer, A., Hungar, H.: Model generation by moderated regular extrapolation. In: Kutsche, R.-D., Weber, H. (eds.) FASE 2002. LNCS, vol. 2306, pp. 80–95. Springer, Heidelberg (2002). doi:10.1007/3-540-45923-5_6
9. Howar, F., Steffen, B.: Learning models for verification and testing — special track at ISoLA 2014 Track Introduction. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014, Part I. LNCS, vol. 8802, pp. 199–201. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45234-9_14

10. Isberner, M., Howar, F., Steffen, B.: Learning register automata: from languages to program structures. Mach. Learn. **96**(1–2), 65–98 (2014)
11. Isberner, M., Howar, F., Steffen, B.: The open-source LearnLib. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9206, pp. 487–495. Springer, Heidelberg (2015). doi:10.1007/978-3-319-21690-4_32
12. Mauritz, M., Howar, F., Rausch, A.: Assuring the safety of advanced driver assistance systems through a combination of simulation and runtime monitoring. In: ISoLA 2016 (2016)
13. Meinke, K., Sindhu, M.A.: Incremental learning-based testing for reactive systems. In: Gogolla, M., Wolff, B. (eds.) TAP 2011. LNCS, vol. 6706, pp. 134–151. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21768-5_11
14. Meinke, K., Sindhu, M.A.: Lbtest: a learning-based testing tool for reactive systems. In: Sixth IEEE International Conference on Software Testing, Verification and Validation, ICST 2013, Luxembourg, Luxembourg, 18–22 March 2013, pp. 447–454 (2013)
15. Pasareanu, C.S., Bobaru, M.: Learning techniques for software verification and validation. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012, Part I. LNCS, vol. 7609, pp. 505–507. Springer, Heidelberg (2012). doi:10.1007/978-3-642-34026-0_37
16. Peled, D., Vardi, M.Y., Yannakakis, M.: Black box checking. J. Automata Lang. Comb. **7**(2), 225–246 (2002)
17. Raffelt, H., Merten, M., Steffen, B., Margaria, T.: Dynamic testing via automata learning. Int. J. Softw. Tools Technol. Transfer **11**(4), 307–324 (2009)
18. Schudeleit, M., Zhang, M., Qi, X., Küçükay, F., Rausch, A.: Enhancement of an adaptive hev operating strategy using machine learning algorithms. In: ISoLA 2016 (2016)