Report from Dagstuhl Seminar 18121

Machine Learning and Model Checking Join Forces

Edited by

Nils Jansen[1], Joost-Pieter Katoen[2], Pushmeet Kohli[3], and Jan Kretinsky[4]

1 **Radboud University Nijmegen, NL,** n.jansen@science.ru.nl
2 **RWTH Aachen University, DE,** katoen@cs.rwth-aachen.de
3 **Google DeepMind – London, GB,** pushmeet@google.com
4 **TU München, DE,** jan.kretinsky@tum.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 18121 "Machine Learning and Model Checking Join Forces". This Dagstuhl Seminar brought together researchers working in the fields of machine learning and model checking. It helped to identify new research topics on the one hand and to help with current problems on the other hand.

Seminar March 18–23, 2018 – https://www.dagstuhl.de/18121
2012 ACM Subject Classification Theory of computation → Machine learning theory, Hardware → Model checking
Keywords and phrases artificial intelligence, cyber-physical systems, formal methods, formal verification, logics, machine learning, model checking, quantitative verification, safety-critical systems
Digital Object Identifier 10.4230/DagRep.8.3.74
Edited in cooperation with Alexis Linard

1 Executive Summary

Nils Jansen
Joost-Pieter Katoen
Pushmeet Kohli
Jan Kretinsky

This Dagstuhl Seminar aimed at bringing together researchers working in the fields of machine learning and model checking. Growing application areas for machine learning, such as autonomous driving, require the exclusion or likely avoidance of unsafe behaviors. An important question is then, how confidence in system behaviors obtained from machine learning can be transferred to formal verification. Vice versa, industrial usage of model checking still suffers from scalability issues for large applications. Leveraging the capabilities of machine learning to assess large data sets will help to enable the verification for more realistic systems.

Machine Learning and Model Checking Join Forces, *Dagstuhl Reports*, Vol. 8, Issue 03, pp. 74–93
Editors: Nils Jansen, Joost-Pieter Katoen, Pushmeet Kohli, and Jan Kretinsky

Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Based on the concrete discussions and inputs from all the participants, we identified the following topics as great challenges to the combination of the fields of machine learning and model checking.

- Safety Verification of Deep Neural Networks
- Formal Program Synthesis and Analysis using Machine Learning
- Representation of Strategies and Controllers
- Explainable Artificial Intelligence
- Challenges for Machine Learning in Motion Planning
- Guarantees on Reinforcement Learning in Verification
- Social and Legal Issues in Artificial Intelligence
- Exploiting Weaknesses in Reinforcement Learning

2 Table of Contents

Executive Summary
Nils Jansen, Joost-Pieter Katoen, Pushmeet Kohli, and Jan Kretinsky 74

Overview of Talks

Formal verification of complex systems: model-based and data-driven methods
Alessandro Abate 78

Shield Synthesis
Roderick Bloem 79

Statistical Parameter Verification of Stochastic Models
Luca Bortolussi 79

Learning to Represent Programs with Graphs
Marc Brockschmidt 80

A Unified View of Piecewise Linear Neural Network Verification
Rudy Bunel and Pushmeet Kohli 81

Managing and Exploiting Uncertainty for Fast Approximate Computations
Michael Carbin 81

Towards Correct-by-Construction Probabilistic Inference
Michael Carbin 82

A dual approach to scalable verification of neural networks
Krishnamurthy Dvijotham 82

Machine Learning and Formal Methods for Assessing Slope Stability
Rüdiger Ehlers 83

Explainable RNNs: Modeling, Learning and Verification
Radu Grosu 83

Government & Industry Perspectives, Cultural Challenges, & Applications for Model Checking & Machine Learning
Laura Humphrey 83

Motion Planning under Uncertainty and Partial Observability
Nils Jansen 84

Bayes meets Dijkstra Exact Inference by Program Verification
Joost-Pieter Katoen 85

Towards Robust and Explainable Artificial Intelligence
Pushmeet Kohli 85

Guarantees in model checking and machine learning
Jan Kretinsky 85

Verification, Analysis, Synthesis Optimization using UPPAAL Stratego
Kim Guldstrand Larsen 86

Learning Adaptive Maintenance Policies for Cyber-Physical Systems
Alexis Linard 86

Graph-Based Reductions for Model Checking and Learning MDPs
Guillermo A. Pérez 87

Using Machine Learning Techniques for Verification of Configuration Files
Ruzica Piskac 88

Verification and Design of Rectifier Networks as Controllers
Hasan Poonawala 88

A gentle introduction to games played on graphs
Jean-François Raskin 89

An introductory tutorial to Bayesian Machine learning and Gaussian Processes
Guido Sanguinetti 89

Learning a SAT Solver from Single-Bit Supervision
Daniel Selsam 89

Oracle-Guided Synthesis of Machine Learning Models
Sanjit A. Seshia 90

Interpretability and Expressiveness of the ML/Synthesis boundary
Armando Solar-Lezama 90

Adversarial Risk and the Dangers of Evaluating Against Weak Attacks
Jonathan Uesato and Pushmeet Kohli 91

Active learning of state machines
Frits Vaandrager 91

Learning from Demonstrations with High-Level Side Information
Min Wen, Ivan Papusha, and Ufuk Topcu 91

Participants 93

3 Overview of Talks

3.1 Formal verification of complex systems: model-based and data-driven methods

Alessandro Abate (University of Oxford, GB)

Two known shortcomings of standard techniques in formal verification are the limited capability to provide system-level assertions, and the scalability to large-scale, complex models, such as those needed in Cyber-Physical Systems (CPS) applications. Using data, which nowadays is becoming ever more accessible, has the potential to mitigate such limitations. However, this notoriously leads to a lack of formal proofs that are needed in safety-critical systems.

This talk covers research which addresses these shortcomings, by bringing model-based and data-driven methods together, which can help pushing the envelope of existing algorithms and tools in formal verification.

In the first part of the talk, I will discuss a new, formal, measurement-driven and model-based automated technique, for the quantitative verification of systems with partly unknown dynamics. I will formulate this setup as a data-driven Bayesian inference problem, formally embedded within a quantitative, model-based verification procedure. I argue that the approach can be applied to complex physical systems (e.g., with spatially continuous variables), driven by external inputs and accessed under noisy measurements.

In the later part of the talk, I will concentrate on systems represented by models that are probabilistic with heterogeneous dynamics (continuous/discrete, i.e. hybrid, as well as nonlinear). Such stochastic hybrid models (SHS) are a natural mathematical framework for CPS. With focus on model-based verification procedures, I will provide algorithms for quantitative model checking of temporal specifications on SHS with formal guarantees. This is attained via the development of formal abstraction techniques based on quantitative approximations.

Theory is complemented by algorithms, all packaged in a software tool that is available to users, and applied in the domain of Smart Energy.

References

1 E. Polgreen, V. B. Wijesuriya, S. Haesaert and A. Abate, "Automated Experiment Design for Efficient Verification of Parametric Markov Decision Processes," QEST17, LNCS 10503, pp. 259–274, 2017.
2 E. Polgreen, V. B. Wijesuriya, S. Haesert and A. Abate, "Data-efficient Bayesian verification of parametric Markov chains," QEST16, LNCS 9826, B. Van Houdt and G. Agha (Eds.), pp. 35–51, 2016.
3 S. Haesaert, P. M. J. V. d. Hof, and A. Abate, "Data-driven and Model-based Verification via Bayesian Identification and Reachability Analysis," Automatica, vol. 79, pp. 115–126, May 2017.

3.2 Shield Synthesis

Roderick Bloem (TU Graz, AT)

Shield synthesis is an approach to enforce safety properties at runtime. A shield monitors the system and corrects any erroneous output values instantaneously. The shield deviates from the given outputs as little as it can and recovers to hand back control to the system as soon as possible. In the first part of this paper, we consider shield synthesis for reactive hardware systems. First, we define a general framework for solving the shield synthesis problem. Second, we discuss two concrete shield synthesis methods that automatically construct shields from a set of safety properties: (1) k-stabilizing shields, which guarantee recovery in a finite time. (2) Admissible shields, which attempt to work with the system to recover as soon as possible. Next, we discuss an extension of k-stabilizing and admissible shields, where erroneous output values of the reactive system are corrected while liveness properties of the system are preserved. Finally, we give experimental results for both synthesis methods. In the second part of the paper, we consider shielding a human operator instead of shielding a reactive system: the outputs to be corrected are not initiated by a system but by a human operator who works with an autonomous system. The challenge here lies in giving simple and intuitive explanations to the human for any interferences of the shield. We present results involving mission planning for unmanned aerial vehicles.

3.3 Statistical Parameter Verification of Stochastic Models

Luca Bortolussi (University of Trieste, IT)

Joint work of Luca Bortolussi, Guido Sanguinetti et al.
Main reference Luca Bortolussi, Dimitrios Milios, Guido Sanguinetti: "Smoothed model checking for uncertain Continuous-Time Markov Chains", Inf. Comput., Vol. 247, pp. 235–253, 2016.
URL http://dx.doi.org/10.1016/j.ic.2016.01.004

Parametric verification and parameter syntesis are fundamental tools to apply formal methods to the design of Cyber-Physical and complex systems. The biggest challenge in this area is scalability to realistic stochastic models of those systems. Recently, a parametric verification has been tackled by a statistical approach grounded in Bayesian Machine Learning techniques, namely Gaussian Processes. The method, called smoothed Model Checking [1], tackles parametric verification of linear time properties of black box statistical models, as a function of model or property parameters, under mild conditions on continuity on parameters of the satisfaction probability. It requires simulation data – substantially the truth value of the property of interest at a small number of parameters points of the parameter space, and only few simulations per point. Being Bayesian, it provides not only an estimate of the satisfaction probability, but also uncertainty estimates at each point. This approach has been leveraged to efficiently solve several tasks, like parameter synthesis [2], system design [4], counterexample generation [6], requirement synthesis [5], parameter estimation from Boolean observations [3], combining it with active learning ideas.

References

1 L. Bortolussi, D. Milios, and G. Sanguinetti, "Smoothed model checking for uncertain Continuous-Time Markov Chains", Information and Computation, vol. 247, pp. 235–253, Apr. 2016.
2 L. Bortolussi and S. Silvetti, "Bayesian Statistical Parameter Synthesis for Linear Temporal Properties of Stochastic Models", in Tools and Algorithms for the Construction and Analysis of Systems, vol. 10806, D. Beyer and M. Huisman, Eds. Cham: Springer International Publishing, 2018, pp. 396–413
3 L. Bortolussi and G. Sanguinetti, "Learning and Designing Stochastic Processes from Logical Constraints", Logical Methods in Computer Science, vol. 11, no. 2, 2015.
4 E. Bartocci, L. Bortolussi, L. Nenzi, and G. Sanguinetti, "System design of stochastic models using robustness of temporal properties", Theoretical Computer Science, vol. 587, pp. 3–25, Jul. 2015.
5 E. Bartocci, L. Bortolussi, and G. Sanguinetti, "Data-driven statistical learning of temporal logic properties", in Formal Modeling and Analysis of Timed Systems, Springer, 2014, pp. 23–37
6 S. Silvetti, A. Policriti, L. Bortolussi, "An Active Learning Approach to the Falsification of Black Box Cyber-Physical Systems". IFM 2017: 3–17

3.4 Learning to Represent Programs with Graphs

Marc Brockschmidt (Microsoft Research UK – Cambridge, GB)

Joint work of Marc Brockschmidt, Mahmoud Khademi, Miltos Allamanis
Main reference Miltiadis Allamanis, Marc Brockschmidt, Mahmoud Khademi: "Learning to Represent Programs with Graphs", CoRR, Vol. abs/1711.00740, 2017.
URL http://arxiv.org/abs/1711.00740

Learning tasks on source code (i.e. , formal languages) have been considered re- cently, but most work has tried to transfer natural language methods and does not capitalize on the unique opportunities offered by code's known sematics. For example, long-range dependencies induced by using the same variable or function in distant locations are often not considered. We propose to use graphs to represent both the syntactic and semantic structure of code and use graph-based deep learning methods to learn to reason over program structures. In this work, we present how to construct graphs from source code and how to scale Gated Graph Neural Networks training to such large graphs. We evaluate our method on two tasks: VarNaming , in which a network attempts to predict the name of a variable given its usage, and VarMisuse, in which the network learns to reason about selecting the correct variable that should be used at a given program location. Our comparison to methods that use less structured program representations shows the advantages of modeling known structure, and suggests that our models learn to infer meaningful names and to solve the VarMisuse task in many cases. Additionally, our testing showed that VarMisuse identifies a number of bugs in mature open-source projects.

3.5 A Unified View of Piecewise Linear Neural Network Verification

Rudy Bunel (University of Oxford, GB) and Pushmeet Kohli (Google DeepMind – London, GB)

Joint work of Rudy Bunel, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, M. Pawan Kumar
Main reference Rudy Bunel, Ilker Turkaslan, Philip H. S. Torr, Pushmeet Kohli, M. Pawan Kumar: "Piecewise Linear Neural Network verification: A comparative study", CoRR, Vol. abs/1711.00455, 2017.
URL http://arxiv.org/abs/1711.00455

The success of Deep Learning and its potential use in many safety-critical applications has motivated research on formal verification of Neural Network (NN) models. Despite the reputation of learned NN models to behave as black boxes and the theoretical hardness of proving their properties, researchers have been successful in verifying some classes of models by exploiting their piecewise linear structure. To facilitate progress on this crucial area, we make two key contributions. First, we present a unified framework that encompasses previous methods. This analysis results in the identification of new methods that combine the strengths of multiple existing approaches. Second, we propose a new data set of benchmarks which includes a collection of previously released testcases. We use the benchmark to provide the first experimental comparison of the algorithms.

3.6 Managing and Exploiting Uncertainty for Fast Approximate Computations

Michael Carbin (MIT – Cambridge, US)

Joint work of Michael Carbin, Brett Boston, Zoe Gong

Many modern applications implement large-scale computations (e.g., machine learning, big data analytics, and financial analysis) in which there is a natural trade-off between the quality of the results that the computation produces and the performance and cost of executing the computation.

Exploiting this fact, researchers have recently developed a variety of new mechanisms that automatically change the structure and execution of an application to enable it to meet its performance requirements. Examples of these mechanisms include skipping portions of the application's computation and executing the application on fast and/or energy-efficient unreliable hardware systems whose operations may silently produce incorrect results.

In this talk, I survey a variety of these new mechanisms as well as present how program verification and analysis makes it possible to verify the safety, security, and accuracy of the approximate applications that these mechanisms produce.

References

1 Leto: Verifying Programs Under Custom Application-Specific Execution Models Brett Boston, Zoe Gong, and Michael Carbin https://arxiv.org/pdf/1805.06090.pdf
2 Verifying Quantitative Reliability for Programs that Execute on Unreliable Hardware Michael Carbin, Sasa Misailovic, and Martin C. Rinard. OOSPLA 2013 https://dl.acm.org/citation.cfm?id=2509546

3 Chisel: Reliability- and Accuracy-Aware Optimization of Approximate Computational Kernels Sasa Misailovic, Michael Carbin, Sara Achour, Zichao Qi, and Martin C. Rinard. OOPSLA 2014 https://dl.acm.org/citation.cfm?id=2660231
4 Proving Acceptability Properties of Relaxed Nondeterministic Approximate Programs Michael Carbin, Deokhwan Kim, Sasa Misailovic, and Martin C. Rinard. PLDI 2012 https://dl.acm.org/citation.cfm?id=2254086

3.7 Towards Correct-by-Construction Probabilistic Inference

Michael Carbin (MIT – Cambridge, US)

Researchers have recently proposed several systems that ease the process of performing Bayesian probabilistic inference. These include systems for automatic inference algorithm synthesis as well as stronger abstractions for manual algorithm development. However, existing systems whose performance relies on the developer manually constructing a part of the inference algorithm have limited support for reasoning about the correctness of the resulting algorithm.

In this talk, I'll present Shuffle, a programming language for manually developing inference procedures that 1) enforces the basic rules of probability theory, 2) enforces the statistical dependencies of the algorithm's corresponding probabilistic model, and 3) generates an optimized implementation. We have used Shuffle to develop inference algorithms for several standard probabilistic models. Our results demonstrate that Shuffle enables a developer to deliver correct and performant implementations of these algorithms.

References

1 Verifying Handcoded Probabilistic Inference Procedures Eric Atkinson, Cambridge Yang, and Michael Carbin https://arxiv.org/abs/1805.01863

3.8 A dual approach to scalable verification of neural networks

Krishnamurthy Dvijotham (Google UK, GB)

We present a novel approach to verifying input-output properties of neural networks. Our approach relies on dualizing an adversarial optimization problem that seeks to find the maximum violation of the property being verified. The dual problem provides an upper bound on the maximum violation, which, if smaller than zero, acts as a certificate of the property being true. We show that this approach can handle networks with arbitrary feedforward architectures and activation functions. Numerical experiments show that our approach can compute tight upper bounds on the maximum error rate of a neural network classifier under bounded adversarial perturbations in the infinity norm and also handle more complex specifications in a computationally tractable fashion.

3.9 Machine Learning and Formal Methods for Assessing Slope Stability

Rüdiger Ehlers (Universität Bremen, DE)

Joint work of Timo Hartmann, Cormac Reale, and other participants of the EU/H2020 project SAFE-10-T

This talk provided a summary of the *slope stability estimation problem* (dealt with in the EU/H2020 project *SAFE-10-T*) from the computer science perspective. As such estimations are safety-critical, solving the problem not only asks for utilizing the capabilities of modern machine learning approaches to infer models from data, but also for the correctness guarantees that are commonly given by techniques from the area of formal methods. The focus of the talk was on presenting the problem and what properties of the learned models need to be verified. The results of a naive application of neural network learning show that learned models do not automatically have the requested properties, and smarter approaches to combining machine learning and formal verification are likely to be useful for solving the problem.

3.10 Explainable RNNs: Modeling, Learning and Verification

Radu Grosu (TU Wien, AT)

Recurrent neural networks have recently achieved considerable success in matching and in many cases surpassing state-of-the-art robotic controllers. However, they have important deficiencies, that make them inappropriate for safety-critical applications: interpretability, size, and robustness to adversarial attacks. In this talk we present a biophysical alternative that does not suffer from such deficiencies.

3.11 Government & Industry Perspectives, Cultural Challenges, & Applications for Model Checking & Machine Learning

Laura Humphrey (AFRL – Wright Patterson, US)

Government & industry are heavily focused on the development of autonomous systems. However, verification & validation of autonomous systems remains a challenge because the space of behaviors autonomous systems can exhibit is orders of magnitude larger than current systems, and they are expected to be able to modify their behavior in response to new situations through approaches like machine learning. Current research in formal methods is focused on how to adapt approaches such as model checking to handle complex systems that incorporate machine learning. However, even if this can be done, many in government & industry do not have a background in formal methods or even discrete mathematics, leading to cultural challenges in the adoption of formal methods. This talk aims to provide

an overview of government & industry perspectives and cultural challenges with respect to verification & validation of autonomous systems. It also presents some potential application problems involving cooperative control of unmanned aerial vehicles, successes in which would help provide concrete evidence to government & industry that model checking and machine learning can be used for design and verification of autonomous systems.

References

1 D. Ahner and C. Parson. Workshop report: Test and evaluation of autonomous systems. Technical report, STAT T&E Center of Excellence, 2016.
2 M. Clark. Autonomy Community of Interest (COI) Test and Evaluation, Verification and Validation (TEVV) working group: Technology investment strategy 2015 – 2018. Technical report, Office of the Assistant Secretary of Defense For Research & Engineering, 2015.

3.12 Motion Planning under Uncertainty and Partial Observability

Nils Jansen (Radboud University Nijmegen, NL)

Joint work of Nils Jansen, Joost-Pieter Katoen, Ufuk Topcu, Sebastian Junges, Ralf Wimmer, Bernd Becker, Leonore Winterer, Christian Dehnert, Steve Carr, Jie Fu

The subject of this talk are motion planning problems where agents move inside environments that are subject to uncertainties and potentially not fully observable. The goal is to compute a strategy or a set of strategies for an agent that is guaranteed to satisfy certain safety or performance specifications. Such problems are naturally modeled by Markov decision processes (MDPs) or partially observable MDPs (POMDPs). We discuss several technical approaches, ranging from the computation of permissive strategies that guarantee safe reinforcement learning in unknown environments, a game-based abstraction framework for POMDPs, as well as the utilization of parameter synthesis for Markov chains to compute randomized strategies for POMDPs. We also consider preliminary work on actively including humans into verification and synthesis processes, and what challenges arise.

References

1 Sebastian Junges, Nils Jansen, Ralf Wimmer, Tim Quatmann, Leonore Winterer, Joost-Pieter Katoen, and Bernd Becker. Finite-state Controllers of POMDPs via Parameter Synthesis. In *UAI*, 2018. to appear.
2 Leonore Winterer, Sebastian Junges, Ralf Wimmer, Nils Jansen, Ufuk Topcu, Joost-Pieter Katoen, and Bernd Becker. Motion planning under Partial observability using Game-based Abstraction. In *CDC*, pages 2201–2208. IEEE, 2017.
3 Steven Carr, Nils Jansen, Ralf Wimmer, Jie Fu, and Ufuk Topcu. Human-in-the-loop synthesis for partially observable markov decision processes. In *ACC*, 2018. to appear.
4 Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu, and Joost-Pieter Katoen. Safety-constrained reinforcement learning for MDPs. In *TACAS*, volume 9636 of *LNCS*, pages 130–146. Springer, 2016.
5 Shashank Pathak, Erika Ábrahám, Nils Jansen, Armando Tacchella, and Joost-Pieter Katoen. A greedy approach for the efficient repair of stochastic models. In *NFM*, volume 9058 of *LNCS*, pages 295–309. Springer, 2015.
6 Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Synthesis in pMDPs: A tale of 1001 parameters. *CoRR*, abs/1803.02884, 2018.

3.13 Bayes meets Dijkstra Exact Inference by Program Verification

Joost-Pieter Katoen (RWTH Aachen University, DE)

Joint work of Kevin Batz, Benjamin Kaminski and Christoph Matheaa
Main reference Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja: "How long, O Bayesian network, will I sample thee? – A program analysis perspective on expected sampling times", in Proc. of the Programming Languages and Systems – 27th European Symposium on Programming, ESOP 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 10801, pp. 186–213, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-319-89884-1_7

In this talk, I will give a perspective on inference in Bayes' networks (BNs) using program verification. I will argue how weakest precondition reasoning a la Dijkstra can be used for exact inference (and more). As exact inference is NP-complete, inference is typically done by means of simulation. I will show how by means of wp-reasoning exact expected sampling times of BNs can be obtained in a fully automated fashion. An experimental evaluation on BN benchmarks demonstrates that very large expected sampling times (in the magnitude of millions of years) can be inferred within less than a second. This provides a means to decide whether sampling-based methods are appropriate for a given BN. The key ingredients are to reason at program code in a compositional manner.

3.14 Towards Robust and Explainable Artificial Intelligence

Pushmeet Kohli (Google DeepMind – London, GB)

Deep learning has led to rapid progress being made in the field of machine learning and artificial intelligence, leading to dramatically improved solutions of many challenging problems such as image understanding, speech recognition, and automatic game playing. Despite these remarkable successes, researchers have observed some intriguing and troubling aspects of the behaviour of these models. A case in point is the presence of adversarial examples which make learning based systems fail in unexpected ways. Such behaviour and the difficultly of interpreting the behaviour of neural networks is a serious hindrance in the deployment of these models for safety-critical applications. In this talk, I will review the challenges in developing models that are robust and explainable and discuss the opportunities for collaboration between the formal methods and machine learning communities.

3.15 Guarantees in model checking and machine learning

Jan Kretinsky (TU München, DE)

We survey various kinds of combining model-checking and machine-learning algorithms, e.g. [1, 2, 3] and the guarantees on each of the two components as well as the result. We discuss the interest in guarantees from perspectives of both communities.

References

1 Tomas Brazdil, Krishnendu Chatterjee, Martin Chmelik, Vojtech Forejt, Jan Kretinsky, Marta Z. Kwiatkowska, David Parker, Mateusz Ujma: *Verification of Markov Decision Processes Using Learning Algorithms*. ATVA 2014.
2 Tomas Brazdil, Krishnendu Chatterjee, Martin Chmelik, Andreas Fellner, Jan Kretinsky: *Counterexample Explanation by Learning Small Strategies in Markov Decision Processes*. CAV 2015.
3 Tomas Brazdil, Krishnendu Chatterjee, Jan Kretinsky, Viktor Toman: *Strategy Representation by Decision Trees in Reactive Synthesis*. TACAS 2018.

3.16 Verification, Analysis, Synthesis Optimization using UPPAAL Stratego

Kim Guldstrand Larsen (Aalborg University, DK)

I will present the framework of stochastic Timed Hybrid Automata and Games and show how the tools UPPAAL, UPPAAL SMC and UPPAAL Stratego allows to perform model checking providing in particular timing guarantees, performance evaluation as well as the ability to synthesize sage and near-optimal control strategies.

For the synthesis we show that the underlying simulation-based methods underlying UPPAAL Stratego including run-based reinforcement learning, Q- and M-learning.

A number of applications (floor heating, adaptive cruise control and intelligent traffic light) will be given.

3.17 Learning Adaptive Maintenance Policies for Cyber-Physical Systems

Alexis Linard (Radboud University Nijmegen, NL)

Joint work of Alexis Linard, Marcos L. P. Bueno
Main reference Alexis Linard, Marcos L. P. Bueno: "Towards Adaptive Scheduling of Maintenance for Cyber-Physical Systems", in Proc. of the Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques – 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 9952, pp. 134–150, 2016.
URL http://dx.doi.org/10.1007/978-3-319-47166-2_9

Scheduling and control of Cyber-Physical Systems (CPS) are becoming increasingly complex, requiring the development of new techniques that can effectively lead to their advancement. This is also the case for failure detection and scheduling component replacements. The large number of factors that influence how failures occur during operation of a CPS may result in maintenance policies that are time-monitoring based, which can lead to suboptimal scheduling of maintenance. We investigate [1] how to improve maintenance scheduling of such complex embedded systems, by means of monitoring in real-time the critical components and dynamically adjusting the optimal time between maintenance actions. The proposed technique relies on machine learning classification models in order to classify component

failure cases vs. non-failure cases, and on real-time updating of the maintenance policy of the sub-system in question. We modeled our simulations in Uppaal, a model checking tool. The results obtained from the domain of printers show that a model that is responsive to the environmental changes can enable consumable savings, while keeping the same product quality, and thus be relevant for industrial purposes.

References

1 Linard, A.; de Paula Bueno, M. L.: Towards adaptive scheduling of maintenance for cyber-physical systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2016, Part I. LNCS, vol. 9952, pp. 134–150. Springer, Cham (2016)

3.18 Graph-Based Reductions for Model Checking and Learning MDPs

Guillermo A. Pérez (Free University of Brussels, BE)

Joint work of Suda Bharadwaj, Stephane Le Roux, Guillermo A. Pérez, Ufuk Topcu

We study the never-worse relation (NWR) for Markov decision processes with an infinite-horizon reachability objective. A state q is never worse than a state p if the maximal probability of reaching the target set of states from p is at most the same value from q, regardless of the probabilities labelling the transitions. Extremal-probability states, end components, and essential states are all special cases of the equivalence relation induced by the NWR. Using the NWR, states in the same equivalence class can be collapsed. Then, actions leading to sub-optimal states can be removed.

Our main results are as follows.

1. We show that the natural decision problem associated to computing the NWR is coNP-complete.
2. We also give a polynomial-time iterative algorithm to under-approximate the NWR.

Among other applications, NWR-based MDP reductions can be seen as a pre-processing of MDPs before model checking or as a way to reduce the number of experiments required to obtain a good approximation of an unknown MDP.

References

1 Stéphane Le Roux, Guillermo A. Pérez: The Complexity of Graph-Based Reductions for Reachability in Markov Decision Processes. FoSSaCS 2018: 367–383
2 Suda Bharadwaj, Stéphane Le Roux, Guillermo A. Pérez, Ufuk Topcu: Reduction Techniques for Model Checking and Learning in MDPs. IJCAI 2017: 4273–4279

3.19 Using Machine Learning Techniques for Verification of Configuration Files

Ruzica Piskac (Yale University – New Haven, US)

Joint work of Ruzica Piskac, Mark Santolucito, Ennan Zhai
Main reference Mark Santolucito, Ennan Zhai, Rahul Dhodapkar, Aaron Shim, Ruzica Piskac: "Synthesizing configuration file specifications with association rule learning", PACMPL, Vol. 1(OOPSLA), pp. 64:1–64:20, 2017.
URL http://dx.doi.org/10.1145/3133888

In this talk we show how to learn specification, using verification for configuration files, when the given examples is actually a set of configuration files. Software failures resulting from configuration errors have become commonplace as modern software systems grow increasingly large and more complex. The lack of language constructs in configuration files, such as types and grammars, has directed the focus of a configuration file verification towards building post-failure error diagnosis tools. We describe a framework which analyzes data sets of correct configuration files and derives rules for building a language model from the given data set. The resulting language model can be used to verify new configuration files and detect errors in them.

References

1 Mark Santolucito, Ennan Zhai, Rahul Dhodapkar, Aaron Shim, Ruzica Piskac: Synthesizing configuration file specifications with association rule learning. PACMPL 1(OOPSLA): 64:1–64:20 (2017)
2 Mark Santolucito, Ennan Zhai, Ruzica Piskac: Probabilistic Automated Language Learning for Configuration Files. CAV (2) 2016: 80–87

3.20 Verification and Design of Rectifier Networks as Controllers

Hasan Poonawala (Univ. of Texas at Austin, US)

Joint work of Hasan Poonawala, Ufuk Topcu

Robotic systems must operate autonomously in environments that are partially known, by relying on complex sensor measurements for control and decision making. A common approach for dealing with this scenario is to design controllers from previously collected sensor data using machine learning. The interaction of dynamics and machine learning errors can lead to suboptimal or even unsafe behavior, such as crashes of autonomous mobile robots. I describe methods to model control strategies that use rectifier networks (a popular type of deep learning architecture) for converting sensor measurements into control signals. The closed-loop model is a piece-wise linear (PWL) continuous-time dynamical system, whose safety and stability properties we can verify using PWL Lyapunov functions and PWL barrier certificates, by solving a linear program. More interestingly, we can design the rectifier network's parameters, by solving a bilinear program. We present an example involving navigation of a mobile robot using different optical sensors. The Lyapunov functions and barrier functions in these examples are chosen by hand. Ideally, we would like to automatically choose these functions based on the closed-loop dynamics, without human intervention. I discuss the challenges to developing such an automatic procedure, and avenues for applications of ideas from model checking of hybrid systems to this task.

3.21 A gentle introduction to games played on graphs

Jean-François Raskin (Free University of Brussels, BE)

This talk gives a quick overview of the models and concepts used for reactive synthesis. It reviews notions of game graphs, infinite duration games, omega-regular winning objectives, strategies, and it gives elements of the algorithms underlying the synthesis of winning strategies. Finally, it considers how two-player games can be combined with Markov Decision Processes to provide models and algorithms able to synthesize strategies that enforce some key properties with certainty and good expectation for other soft properties.

3.22 An introductory tutorial to Bayesian Machine learning and Gaussian Processes

Guido Sanguinetti (University of Edinburgh, GB)

In this talk, I give a tutorial overview of Bayesian machine learning methods, with a particular focus on Gaussian Processes, a nonparameteric Bayesian model for regression which works by imposing a prior distribution directly on a space of function. The talk is preparatory to the material covered by Luca Bortolussi on his talk on smoothed Model Checking.

3.23 Learning a SAT Solver from Single-Bit Supervision

Daniel Selsam (Stanford University, US)

Joint work of Daniel Selsam, Matthew Lamm, Benedikt Bunz, Percy Liang, Leonardo de Moura, David L. Dill
Main reference Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, David L. Dill: "Learning a SAT Solver from Single-Bit Supervision", CoRR, Vol. abs/1802.03685, 2018.
URL http://arxiv.org/abs/1802.03685

We present NeuroSAT, a message passing neural network that learns to solve SAT problems after only being trained as a classifier to predict satisfiability. Although it is not competitive with state-of-the-art SAT solvers, NeuroSAT can solve problems that are substantially larger and more difficult than it ever saw during training by simply running for more iterations. Moreover, NeuroSAT generalizes to novel distributions; after training only on random SAT problems, at test time it can solve SAT problems encoding graph coloring, clique detection, dominating set, and vertex cover problems, all on a range of distributions over small random graphs.

3.24 Oracle-Guided Synthesis of Machine Learning Models

Sanjit A. Seshia (University of California – Berkeley, US)

Main reference Susmit Jha, Sanjit A. Seshia: "A theory of formal synthesis via inductive learning", Acta Inf., Vol. 54(7), pp. 693–726, 2017.
URL http://dx.doi.org/10.1007/s00236-017-0294-5
Main reference Sanjit A. Seshia, Dorsa Sadigh: "Towards Verified Artificial Intelligence", CoRR, Vol. abs/1606.08514, 2016.
URL http://arxiv.org/abs/1606.08514

We consider the problem of designing machine learning models used within a larger system that must satisfy a formal specification, a step towards the goal of verified artificial intelligence (AI) [4]. This problem is an instance of a class of problems termed as formal inductive synthesis [5]. An illustrative example is the use of deep neural networks for perception in an autonomous driving system. We present a compositional falsification approach that combines a falsifier for cyber-physical system (CPS) models with a machine learning (ML) analyzer that performs a more detailed analysis of a machine learning model [1]. The ML analyzer performs semantic transformations to input data (images) to generate new data so as to find system-level counterexamples (e.g. safety violations). We show how retraining the models with generated images can both improve accuracy and eliminate system-level counterexamples [2]. Such counterexample-guided retraining is an instance of oracle-guided inductive synthesis, and may also be seen as a "semantic" approach to adversarial machine learning [3]. We describe our results using oracle-guided synthesis of ML models for autonomous driving.

References

1 Tommaso Dreossi, Alexandre Donze, and Sanjit A. Seshia. *Compositional Falsification of Cyber-Physical Systems with Machine Learning Components*. Proc. NASA Formal Methods Symposium (NFM), May 2017.
2 Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Keutzer, Alberto Sangiovanni-Vincentelli, and Sanjit A. Seshia. *Counterexample-Guided Data Augmentation*. Proc. International Joint Conference on Artificial Intelligence (IJCAI), July 2018.
3 Tommaso Dreossi, Somesh Jha, and Sanjit A. Seshia. *Semantic Adversarial Deep Learning*. In 30th International Conference on Computer Aided Verification (CAV), July 2018.
4 Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. *Towards Verified Artificial Intelligence*. ArXiv e-prints, July 2016.
5 Susmit Jha and Sanjit A. Seshia. *A Theory of Formal Synthesis via Inductive Learning*. Acta Informatica, 54(7):693–726, 2017.

3.25 Interpretability and Expressiveness of the ML/Synthesis boundary

Armando Solar-Lezama (MIT – Cambridge, US)

The talk describes some recent work applying ideas from synthesis and FM to problems in ML, such as interpreting a decision made by a neural network, as well as applying ideas from ML to make synthesis more efficient and expressive.

3.26 Adversarial Risk and the Dangers of Evaluating Against Weak Attacks

Jonathan Uesato (Google DeepMind – London, GB) and Pushmeet Kohli (Google DeepMind – London, GB)

Joint work of Jonathan Uesato, Brendan O'Donoghue, Aäron van den Oord, Pushmeet Kohli
Main reference Jonathan Uesato, Brendan O'Donoghue, Aäron van den Oord, Pushmeet Kohli: "Adversarial Risk and the Dangers of Evaluating Against Weak Attacks", CoRR, Vol. abs/1802.05666, 2018.
URL http://arxiv.org/abs/1802.05666

This paper investigates recently proposed approaches for defending against adversarial examples and evaluating adversarial robustness. The existence of adversarial examples in trained neural networks reflects the fact that expected risk alone does not capture the model's performance against worst-case inputs. We motivate the use of *adversarial risk* as an objective, although it cannot easily be computed exactly. We then frame commonly used attacks and evaluation metrics as defining a tractable surrogate objective to the true adversarial risk. This suggests that models may be *obscured* to adversaries, by optimizing this surrogate rather than the true adversarial risk. We demonstrate that this is a significant problem in practice by repurposing gradient-free optimization techniques into adversarial attacks, which we use to decrease the accuracy of several recently proposed defenses to near zero. Our hope is that our formulations and results will help researchers to develop more powerful defenses.

3.27 Active learning of state machines

Frits Vaandrager (Radboud University Nijmegen, NL)

Main reference Frits W. Vaandrager: "Model learning", Commun. ACM, Vol. 60(2), pp. 86–95, 2017.
URL http://dx.doi.org/10.1145/2967606

In this tutorial, I review the basic theory of active learning of state machines and recent applications in which this theory was used to learn models of (and find bugs in) smart cards, implementations of network protocols, and embedded systems controllers. I discuss some recent results and outline research challenges.

3.28 Learning from Demonstrations with High-Level Side Information

Min Wen (University of Pennsylvania – Philadelphia, US), Ivan Papusha, and Ufuk Topcu (University of Texas – Austin, US)

Main reference Min Wen, Ivan Papusha, Ufuk Topcu: "Learning from Demonstrations with High-Level Side Information", in Proc. of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pp. 3055–3061, ijcai.org, 2017.
URL http://dx.doi.org/10.24963/ijcai.2017/426

We consider the problem of learning from demonstration, where extra side information about the demonstration is encoded as a co-safe linear temporal logic formula. We address two known limitations of existing methods that do not account for such side information. First, the

policies that result from existing methods, while matching the expected features or likelihood of the demonstrations, may still be in conflict with high-level objectives not explicit in the demonstration trajectories. Second, existing methods fail to provide a priori guarantees on the out-of-sample generalization performance with respect to such high-level goals. This lack of formal guarantees can prevent the application of learning from demonstration to safetycritical systems, especially when inference to state space regions with poor demonstration coverage is required. In this work, we show that side information, when explicitly taken into account, indeed improves the performance and safety of the learned policy with respect to task implementation. Moreover, we describe an automated procedure to systematically generate the features that encode side information expressed in temporal logic.

Participants

- Alessandro Abate
University of Oxford, GB
- Erika Abraham
RWTH Aachen University, DE
- Ezio Bartocci
TU Wien, AT
- Roderick Bloem
TU Graz, AT
- Luca Bortolussi
University of Trieste, IT
- Tomáš Brázdil
Masaryk University – Brno, CZ
- Marc Brockschmidt
Microsoft Research UK – Cambridge, GB
- Rudy Bunel
University of Oxford, GB
- Michael Carbin
MIT – Cambridge, US
- Rayna Dimitrova
University of Leicester, GB
- Krishnamurthy Dvijotham
Google UK – London, GB
- Rüdiger Ehlers
Universität Bremen, DE
- Andreas Berre Eriksen
Aalborg University, DK
- Radu Grosu
TU Wien, AT
- Arnd Hartmanns
University of Twente, NL
- Laura Humphrey
AFRL – Wright Patterson, US
- Manfred Jaeger
Aalborg University, DK
- Nils Jansen
Radboud University Nijmegen, NL
- Sebastian Junges
RWTH Aachen University, DE
- Joost-Pieter Katoen
RWTH Aachen University, DE
- Pushmeet Kohli
Google DeepMind – London, GB
- Jan Kretinsky
TU München, DE
- Kim Guldstrand Larsen
Aalborg University, DK
- Alexis Linard
Radboud University Nijmegen, NL
- Tobias Meggendorfer
TU München, DE
- Daniel Neider
MPI-SWS – Kaiserslautern, DE
- Guillermo A. Pérez
Free University of Brussels, BE
- Ruzica Piskac
Yale University – New Haven, US
- Hasan Poonawala
Univ. of Texas at Austin, US
- Pavithra Prabhakar
Kansas State University – Manhattan, US
- Jean-Francois Raskin
Free University of Brussels, BE
- Guido Sanguinetti
University of Edinburgh, GB
- Daniel Selsam
Stanford University, US
- Sanjit A. Seshia
University of California – Berkeley, US
- Armando Solar-Lezama
MIT – Cambridge, US
- Ufuk Topcu
University of Texas – Austin, US
- Jana Tumova
KTH Royal Institute of Technology – Stockholm, SE
- Jonathan Uesato
Google DeepMind – London, GB
- Frits Vaandrager
Radboud University Nijmegen, NL
- Min Wen
University of Pennsylvania – Philadelphia, US
- Leonore Winterer
Universität Freiburg, DE