	HS 2	HS 3
	Thursday, March 14	
08:30 – 08:50	Registration	
08:50 – 09:00	Welcoming Speech (HS 3)	
09:00 – 10:00	**Filip Mazowiecki (LaBRI, University of Bordeaux)** **The Reachability Problem for Petri Nets is Not Elementary** (HS 3)	
10:00 – 10:30	**Coffee break**	
10:30 – 11:15	Daniel Stan (Universität des Saarlandes) Syntactic Partial Order Compression for Probabilistic Reachability	Christina Mika-Michalski (Universität Duisburg-Essen) Fixpoint Games on Continuous Lattices
11:15 – 12:00	Sebastian Kenter (WWU Münster) Expressing lock-sensitive reachability in graph structures	Dennis Nolte (Universität Duisburg-Essen) Materialization matters: Investigating abstract graph rewriting
12:00 – 13:00	**Lunch**	
13:00 – 13:45	Hendrik Göttmann (TU Darmstadt) Compositional Input/Output Conformance Testing of Live Timed Systems with Silent Transitions	Jens Gutsfeld (WWU Münster) An Extension of CTL* for Recursive Programs
13:45 – 15:15	Christoph Matheja (RWTH Aachen) *Tutorial*: An Introduction to Weakest Preexpectation Reasoning and Quantitative Separation Logic	Uwe Nestmann (TU Berlin) *Discussion Session*: Formal Verification of Distributed Algorithms
15:15 – 15:45	**Coffee break**	
15:45 – 16:30	Lars Luthmann (TU Darmstadt) Checking Bisimulation of Parametric Timed Automata	Ronja Enseleit (HU Berlin) Soundness preserving asynchronous composition of workflow nets with interfaces
16:30 – 17:15	Harsh Beohar (Universität Duisburg-Essen) Bisimulation maps in presheaf categories	Johannes Gareis (Uni Bamberg) Modal Interface Automata with Data

	Friday, March 15	
09:00 – 10:00	**Karoliina Lehtinen (University of Liverpool)** **Quasi-polynomial automata for playing with parity games** (HS 3)	
10:00 – 10:30	**Coffee break**	
10:30 – 11:15	Sebastian Wolff (TU Braunschweig) Decoupling Lock-Free Data Structures from Memory Reclamation for Static Analysis	Philipp Meyer (TU München) Complexity of analyzing workflow nets with resources, costs and time
11:15 – 12:00	Stefan Jaax (TU München) Expressive Power of Broadcast Protocols	
12:00 – 13:00	**Lunch**	
13:00 – 13:30	Business Meeting (HS 3)	

Filip Mazowiecki (LaBRI, University of Bordeaux)

The Reachability Problem for Petri Nets is Not Elementary

Petri nets, also known as vector addition systems, are a long established and widely used model of concurrent processes. The complexity of their reachability problem is one of the most prominent open questions in the theory of verification. That the reachability problem is decidable was established by Mayr in his seminal STOC 1981 work, and the currently best upper bound is non-primitive recursive cubic-Ackermannian of Leroux and Schmitz from LICS 2015. We show that the reachability problem is not elementary. Until this work, the best lower bound has been exponential space, due to Lipton in 1976.

Karoliina Lehtinen (University of Liverpool)

Quasi-polynomial automata for playing with parity games.

Parity games are infinite two-player games, which are used in verification, automata theory, and reactive synthesis. Solving parity games -- that is, deciding which player has a winning strategy -- is one of the few problems known to be in both UP and co-UP yet not known to be in P. So far, the quest for a polynomial algorithm has lasted over 25 years.

Since then, several different quasi-polynomial algorithms have been published. I will present an automata-theoretic quasi-polynomial solution, which relates the algorithmic complexity of parity games to the complexity of the logics and automata we use to manipulate them.

In 2017, a major breakthrough occurred: parity games were shown to be solvable in quasi-polynomial time.

Christoph Matheja (RWTH Aachen)

Tutorial: An Introduction to Weakest Preexpectation Reasoning and Quantitative Separation Logic

We give a short introduction to classical weakest precondition reasoning à la Dijkstra. This technique allows for reasoning about nonprobabilistic programs at the level of predicates. We then show how this technique can be extended to weakest preexpectation reasoning à la Kozen / McIver & Morgan. This latter technique allows for reasoning about probabilistic programs at the level of quantities which evaluate to real numbers instead of Boolean values.

We then introduce Quantitative Separation Logic (QSL). In contrast to classical separation logic, QSL employs quantities instead of predicates. The connectives of classical separation logic, separating conjunction and separating implication, are lifted from predicates to quantities. This extension is conservative: Both connectives are backward compatible to their classical analogs and obey the same laws, e.g. modus ponens, adjointness of separating conjunction and implication, etc.

Sebastian Kenter (Westfälische Wilhelms-Universität (WWU) Münster)

Expressing lock-sensitive reachability in graph structures

Reachability problems for parallel programs in the presence of locks are close to the border of decidability. We consider a variant of dynamic pushdown networks as a novel and flexible model for parallel programs with recursion and unboundedly many locks. Exploiting the decidability of monadic second-order satisfiability on a set of graphs generated by a hyperedge replacement grammar (HRG), we show that reachability is decidable for this model. To this end, we use augmented execution trees that indicate compatibility of executions with the locking mechanism, and state an HRG that generates those graphs. In this talk we will focus on presenting in a modular manner the reasoning why such graphs can capture the desired locking behaviour.

Jens Gutsfeld (Westfälische Wilhelms-Universität (WWU) Münster)

An Extension of CTL for Recursive Programs*

Pushdown systems offer a natural model for verifying recursive programs. The logic CaRet, which extends LTL by modalities for navigation over calls and returns of procedures, has been a starting point for new logics for pushdown systems for more than a decade. Last year, we presented a branching time variant of CaRet (BranchCaRet) which extends CTL by CaRet-like modalities. This year, we unify both logics in a CTL*-like extension called BranchCaRet* that subsumes both CaRet and BranchCaRet. We discuss properties expressible in this logic as well as a model checking algorithm and complexity results.

Lars Luthmann (Real-Time Systems Lab, TU Darmstadt)

Checking Bisimulation of Parametric Timed Automata

Timed automata (TA) constitute a well-established formalism for the specification and automated analysis of discrete state/continuous time reactive system behaviors. Concerning the problem of comparing the real-time behaviors of a candidate implementation against a specification, both given as TA, it is a well-known fact that timed trace equivalence is undecidable, whereas timed bisimulation is decidable. However, the inherent limitations of expressiveness of TA in their basic form are a serious limiting factor of applying TA in practice, and a wide range of TA extensions have been proposed in the recent past to overcome these limitations. Amongst others, parametric timed automata (PTA) generalize timed automata by parametric timing constraints with freely-adjustable time boundaries instead of constant values as in TA. Unfortunately, many interesting semantic properties considered for TA, including timed bisimilarity, become undecidable for PTA in general. In this talk, we first revisit, and further elaborate, timed bisimulation checking for TA and then generalize the approach to (inherently incomplete) parametric bisimulation checking for PTA. We further investigate decidable sub-classes of PTA.

Hendrik Göttmann (Real-Time Systems Lab, TU Darmstadt)

Compositional Input/Output Conformance Testing of Live Timed Systems with Silent Transitions

Input/output conformance testing theories (e. g., /ioco/) are concerned with formally defining when observable output behaviors of an implementation under test conform to those permitted by a specification. Thereupon, several timed extensions of /ioco/, called /tioco/, have been proposed, further taking into account the permitted delay between input/output actions. In this talk, we propose an improved version of /tioco/, called /live timed ioco/ (/ltioco/), tackling various essential flaws of existing notions. First, /tioco/ requires input-enabledness for implementations which is problematic in a timed setting, enabling live-locks perpetually preventing systems from progressing. Second, a reasonable definition of quiescence (i. e., observable absence of any outputs) for timed systems has also to be done with care: unlike /tioco/, /ltioco/ distinguishes safe outputs being allowed to happen, from live outputs being enforced to happen within a certain time period thus imposing two different facets of quiescence. Finally, /tioco/ is frequently defined on Timed Input/Output Labeled Transition Systems (TIOLTS), a semantic representation of Timed Input/Output Automata (TIOA). However, TIOLTS are, by definition, infinitely branching. In contrast, we define /ltioco/ on input/output zone graphs, a finite representation of TIOA semantics. Additionally, we investigate compositionality of /ltioco/ with respect to parallel composition. We especially distinguish between the composition of automata with disjoint clock sets, and automata with intersecting clock sets, i. e., some clocks may be shared between the automata.

Sebastian Wolff (TU Braunschweig)

Decoupling Lock-Free Data Structures from Memory Reclamation for Static Analysis

Verification of concurrent data structures is one of the most challenging tasks in software verification. The topic has received considerable attention over the course of the last decade. Nevertheless, human-driven techniques remain cumbersome and notoriously difficult while automated approaches suffer from limited applicability. The main obstacle for automation is the complexity of concurrent data structures. This is particularly true in the absence of garbage collection. The intricacy of lock-free memory management paired with the complexity of concurrent data structures makes automated verification prohibitive.

In this work we present a method for verifying concurrent data structures and their memory management separately. We suggest two simpler verification tasks that imply the correctness of the data structure. The first task establishes an over-approximation of the reclamation behavior of the memory management. The second task exploits this over-approximation to verify the data structure without the need to consider the implementation of the memory management itself. To make the resulting verification tasks tractable for automated techniques, we establish a second result. We show that a verification tool needs to consider only executions where a single memory location is reused. We implemented our approach and were able to verify linearizability of Michael&Scott's queue and the DGLM queue for both hazard pointers and epoch-based reclamation. To the best of our knowledge, we are the first to verify such implementations fully automatically.

Dennis Nolte (Universität Duisburg-Essen)

Materialization matters: Investigating abstract graph rewriting

In this talk we will have a look on an abstract semantics for double-pushout rewriting of graphs. The focus is on the so-called materialization of left-hand sides from abstract graphs, a central concept in abstract rewriting. Furthermore, we introduce an extension of the framework by enriching graphs with annotations and give a characterization of strongest post-conditions.

Christina Mika-Michalski (Universität Duisburg-Essen)

Fixpoint Games on Continuous Lattices

In model-checking specifications are represented by mu-calculus formulas. Such formulas can be transformed into systems of fixpoint equations. Inspired by recent work on lattice-theoretic progress measures, we develop a game-theoretical approach to the solution of systems of monotone equations over lattices, where for each single equation either the least or greatest solution is taken. A simple parity game, referred to as fixpoint game, is defined that provides a correct and complete characterisation of the solution of systems of equations over continuous lattices, a quite general class of lattices widely used in semantics.

Harsh Beohar (Universität Duisburg-Essen)

Bisimulation maps in presheaf categories

The category of presheaves over a (small) category is a suitable semantic universe to study behaviour of various dynamical systems. In particular, presheaves can be used to record the executions of a system and their morphisms correspond to simulation maps for various kinds of state-based systems. In this talk, I'll introduce a notion of bisimulation map between presheaves (or executions) to capture well known behavioural equivalences in an abstract way. As an application, I'll show how to derive the characterisation of ∀-fair bisimulation in this framework.

Johannes Gareis (Otto-Friedrich-Universität Bamberg)

Modal Interface Automata with Data

Interface theories based on Interface Automata (IA) are formalisms for the component-based specication of concurrent systems. In our previous work, we showed how de Alfaro and Henzinger's original IA theory can be conservatively extended by shared memory data. Modal Interface Automata also belong to a family of interface theories that combine Interface Automata (IA) with Modal Transition Systems. Now, we want to show how Modal Interface Automata can be extended by means of data.

Ronja Enseleit (Humboldt-Universität zu Berlin)

Soundness preserving asynchronous composition of workflow nets with interfaces

We equip van der Aalst' workflow nets with interfaces, along which the nets can be composed in order to communicate asynchronously during execution. We state non-trivial, liberal conditions that guarantee the composition of two sound such nets be again a sound workflow net.

Uwe Nestmann (TU Berlin)

Discussion Session: Formal Verification of Distributed Algorithms

Over the years, we have worked on various distributed algorithms, trying out a number of quite diffferent techniques, also at varying levels of formality, in some cases even using proof checkers like the Isabelle/HOL system. The goal of the session would be to stimulate a lively exchange of experiences and also to point out challenges in this domain.

Daniel Stan (Universität des Saarlandes)

Syntactic Partial Order Compression for Probabilistic Reachability

The state space explosion problem is among the largest impediments to the performance of any model checker. Modelling languages for compositional systems contribute to this problem by placing each instruction of an instruction sequence onto a dedicated transition, giving concurrent processes opportunities to interleave after every instruction. Users wishing to avoid the excessive number of interleavings caused by this default can choose to explicitly declare instruction sequences as atomic, which however requires careful considerations regarding the impact this might have on the model as well as on the properties that are to be checked. We instead propose a preprocessing technique that automatically identifies instruction sequences that can safely be considered atomic. This is done in the context of concurrent variable-decorated Markov Decision Processes. Our approach is compatible with any off-the-shelf probabilistic model checker. We prove that our transformation preserves maximal reachability probabilities and present case studies to illustrate its usefulness.

Philipp Meyer (TU München)

Complexity of analyzing workflow nets with resources, costs and time

Workflow nets are a class of Petri nets for representation and analysis of business process. Transitions represent tasks and can additionally be annotated with quantitative information such as the resources, the cost and the time required to execute this task. This leads to several analysis questions: what is the minimum number of resources required to execute the workflow as fast as possible? Given enough resources, what is the expected cost or the expected time of the workflow net?

We present some recents results on the complexity of computing these numbers. As workflow nets can express both concurrency and choice, the type of choice and combination with concurrency often leads to different complexities. We mainly look at the the simplest class of free-choice nets, where choices can always be freely resolved and are not in conflict with concurrency. Here, basic questions for workflows such as soundness or reachability in sound nets can be decided in polynomial time.

We show that determining the minimal number of resources needed to always execute the net as fast as possible is NP-hard, even for nets without choice but deterministic timining information, or for free-choice nets with unknown timing information. In the latter case, we present an algorithm for computing this number through an approximation method and a model checker. Determining the expected cost can be done in polynomial time for free-choice nets with probabilistic choice, and is PSPACE-complete for general nets. In contrast to this, determining the expected time is #P-hard for free-choice nets, even if all times are 0 ore 1 and all probabilities are 1 or 0.5. Therefore a polynomial-time algorithm is unlikely to exist. However, we present an exponential-time algorithm for computing the expected time. We evaluate computing the number of resources, expected cost and time on as set of 642 sound free-choice workflow nets from a popular benchmark suite. Here, we can always compute these numbers within milliseconds.

Stefan Jaax (TU München)

Expressive Power of Broadcast Protocols

interacting in pairs. It has been shown that their computational power is rather limited: They can only compute the predicates expressible in Presburger arithmetic. Broadcast protocols are extensions of population protocols where individual agents can broadcast signals to all other agents. Broadcast protocols are more expressive than standard population protocols, e.g. they can compute whether a given number is a power of two, which is not a Presburger predicate and thus not computable by a population protocol. - But how expressive are broadcast protocols exactly? In this talk we show that broadcast protocols compute precisely the predicates in NSPACE(n), that is, all predicates computable by a Turing machine in nondeterministic linear space.