Formal Reasoning in Distributed Algorithms

TU Wien, Freihaus, Green area, 5^{th} floor, Seminarraum 104

Day 1 (July 23)

Joseph Halpern : Invited talk — **9:00 – 10:15**

Beyond Nash Equilibrium: Solution Concepts for the 21st Century

Abstract. Nash equilibrium is the most commonly-used notion of equilibrium in game theory. However, it suffers from numerous problems. Some are well known in the game theory community; for example, the Nash equilibrium of repeated prisoner's dilemma is neither normatively nor descriptively reasonable. However, new problems arise when considering Nash equilibrium from a computer science perspective: for example, Nash equilibrium is not robust (it does not tolerate "faulty" or "unexpected" behavior), it does not deal with coalitions, it does not take computation cost into account, and it does not deal with cases where players are not aware of all aspects of the game. In this talk, I discuss solution concepts that try to address these shortcomings of Nash equilibrium. This talk represents joint work with various collaborators, including Ittai Abraham, Danny Dolev, Rica Gonen, Rafael Pass, and Leandro Rego. No background in game theory will be presumed.

Coffee break

10:15 – 10:45

Matthias Függer, Thomas Nowak, Ulrich Schmid and Robert Najvirt — **10:45 – 11:15**

Towards binary circuit models that faithfully reflect physical (un)solvability

Abstract. Binary circuit models are high-level abstractions intended to reflect the behavior of digital circuits, while restricting signal values to 0 and 1. Such models play an important role in assessing the correctness and performance characteristics of digital circuit designs: (i) modern circuit design relies on fast digital timing simulation tools and, hence, on accurate binary-valued circuit models that faithfully model signal propagation, even throughout a complex design, and (ii) binary circuit models provide a level of abstraction that is amenable to formal analysis.

Of particular importance is the ability to trace glitches and other short pulses, as their presence/absence may affect a circuit's correctness and its performance characteristics.

We show that that no existing binary-valued circuit model proposed so far, including the two most commonly used pure and inertial delay channels, faithfully captures glitch propagation: For the simple Short-Pulse Filtration (SPF) problem, which is related to a circuit's ability to suppress a single glitch, we show that the quite broad class of bounded single-history channels either contradict the unsolvability of SPF in bounded time or the solvability of SPF in unbounded time in physical circuits.

We then propose a class of binary circuit models that do not suffer from this deficiency: Like bounded single-history channels, our involution channels involve delays that may depend on the time of the previous output transition. Their characteristic property are delay functions which are based on involutions, i.e., functions that form their own inverse.

We prove that, in sharp contrast to what is possible with bounded single-history channels, SPF cannot be solved in bounded time whereas it is easy to provide an unbounded SPF implementation. It hence follows that binary-valued circuit models based on involution channels allow to solve SPF precisely when this is possible in physical circuits.

This renders them a promising candidate, both, for simulation and the formal analysis of circuits.

(Part of the results presented in this talk were published at ASYNC'13.)

11:15 – 11:45

Benjamin Bisping, Paul-David Brodmann, Tim Jungnickel, **Christina Rickmann**, Henning Seidler, Anke Stüber, Arno Wilhelm-Weidner, Kirstin Peters and Uwe Nestmann

Mechanical Verification of a Constructive Proof for FLP

Abstract. We present a formalization of Vlzer's paper "A constructive proof for FLP" using the interactive theorem prover Isabelle/HOL. We focus on the main differences between our proof and Vlzer's and summarize necessary design decisions in our formal approach.

11:45 – 12:15

Noran Azmy

Having SPASS with Pastry and TLA+

Abstract. Peer-to-peer protocols are becoming more and more popular for modern internet applications. While such protocols typically come with certain correctness and performance guarantees, verification attempts using formal methods invariably discover inconsistencies. We are interested in using the SPASS theorem prover for the verification of peer-to-peer protocols, that are modeled in the specification language TLA+. In addition to the specification language, TLA+ comes with its own verification tools: an interactive proof written in the TLA+ proof language consists of steps, or proof obligations, that are processed by TLA+'s own proof manager, and passed to one of several back-end provers such as Zenon or Isabelle/TLA. In 2013, Tianxiang Lu already made the first steps in the case of the protocol Pastry, where the author's attempt at formal verification reveals that the full protocol is incorrect with respect to a safety property which he calls correct delivery. His final proof of correctness is for a restricted version of the protocol and seriously lacks automation, due to the inability of current back-end provers to tackle proof obligations from this class of problems, which typically contain a mixture of uninterpreted functions, modular integer arithmetic, and set theory with the cardinality operator.

Our ultimate goal is to create a SPASS back end for TLA+ that is better capable of solving this kind of proof obligations. This includes (1) the development of an efficient and effective translation from the strongly untyped, higher order TLA+ to a typed, first-order input language for SPASS, and (2) incorporating theory reasoning, typed reasoning and other necessary techniques into SPASS itself.

In this paper, we give the first insights from running the current version of SPASS on the proof obligations from Lu's proof using a prototype, untyped translation. We also devise a modification to the current translation that achieves an impressive improvement in the way SPASS deals with the particularly problematic CHOOSE operator of TLA+.

12:15 – 12:45

Ana Sokolova

Concurrent Data Structures: Semantics and (Quantitative) Relaxation

Abstract. Concurrent data structures are often considered bottlenecks in terms of performance in scalability. Their semantics usually consists of two parts: a sequential specification (specifying correct sequential behavior) and a consistency condition (specifying valid concurrent histories).

There is a trade-off between performance and correctness in implementing concurrent data structures. Better performance may be achieved at the expense of relaxing correctness, by redefining the semantics of data structures.

We have addressed a redefinition of data structure sequential specification and presented a systematic and formal framework for obtaining new data structures by quantitatively relaxing existing ones. In contrast to other existing work, our relaxations are semantic (quantitative relaxation in terms of data-structure states). As an instantiation of our framework, we presented two simple yet generic relaxation schemes, called out-of-order and stuttering relaxation, along with several ways of computing distances. We showed that the out-of-order relaxation, when further instantiated to stacks, queues, and priority queues, amounts to tolerating bounded out-of-order behavior, which cannot be captured by a purely syntactic relaxation (distance in terms of sequence manipulation, e.g. edit distance). This work was done together with Thomas Henzinger, Christoph Kirsch, Hannes Payer, and Ali Sezgin and published at POPL 2013. Part of my talk will focus on it.

I will also briefly overview different implementations of relaxed concurrent data structures, that indeed show good performance and linear scalability. These results were obtained together with Andreas Haas, Thomas Henzinger, Christoph Kirsch, Michael Lippautz, Hannes Payer, and Ali Sezgin and published partially in the above mentioned work, partially in a paper at CF 2013.

Furthermore, I will mention ongoing work on: (1) Different view of relaxing the semantics of concurrent data structures by both relaxing the sequential specification (replacing sequentiality with a preorder) and relaxing the consistency condition (the consistency predicate). This is part of research done together with Mike Dodds and Ali Sezgin; and (2) Exhaustive testing for comparing the semantics of strict and relaxed concurrent data structure implementations in practice. We are pursuing this research direction together with all the above mentioned people (except for Mike Dodds) as well as Andreas Holzer and Helmut Veith.

Lunch

13:00 – 14:30

Ahmed Bouajjani : Invited talk

14:30 – 15:45

On Checking Correctness of Concurrent Data Structures

Abstract. We address the issue of checking the correctness of implementations of libraries of concurrent/distributed data structures. We present results concerning the verification of linearizability in the context of shared-memory concurrent data structures, and eventual consistency in the context of replicated, distributed data structures.

This talk is based on joint work with Michael Emmi, Constantin Enea, and Jad Hamza.

Coffee break

16:00 – 16:30

Cezara Dragoi, Thomas Henzinger, Helmut Veith, Josef Widder and Damien Zufferey

16:30 – 17:00

A Logic-based Framework for Verifying Consensus Algorithms

Abstract. Fault-tolerant distributed algorithms play an important role in ensuring the reliability of many software applications. In this paper we consider distributed algorithms whose computations are organized in rounds. To verify the correctness of such algorithms, we reason about (i) properties (such as invariants) of the state, (ii) the transitions controlled by the algorithm, and (iii) the communication graph. We introduce a logic that addresses these points, and contains set comprehensions with cardinality constraints, function symbols to describe the local states of each process, and a limited form of quantier alternation to express the verication conditions. We show its use in automating the verication of consensus algorithms. In particular, we give a semi-decision procedure for the unsatisability problem of the logic and identify a decidable fragment. We successfully applied our framework to verify the correctness of a variety of consensus algorithms tolerant to both benign faults (message loss, process crashes) and value faults (message corruption).

17:00 – 17:30 Francesco Alberti, **Silvio Ghilardi**, and Natasha Sharygina

Monotonic Abstraction Techniques: from Parametric to Software Model Checking

Abstract. Monotonic abstraction is a technique introduced in model checking parameterized distributed systems in order to cope with transitions containing global conditions within guards. The technique has been re-interpreted in a declarative setting in previous papers of ours and applied to the verification of fault tolerant systems under the so-called stopping failures model. The declarative reinterpretation consists in logical techniques (quantifier relativizations and, especially, quantifier instantiations) making sense in a broader context. In fact, we recently showed that such techniques can over-approximate array accelerations, so that they can be employed as a meaningful (and practically effective) component of CEGAR loops in software model checking too.

17:30 – 18:00 Giorgio Delzanno and **Michele Tatarek**

Model Checking Distributed Consensus Algorithms

Abstract. We present formal models of distributed consensus algorithms like Paxos and Raft in the executable specification language Promela extended with a new type of guards, called counting guards, needed to implement transitions that depend on majority voting. Our formalization exploits abstractions that follow from reduction theorems (w.r.t. the number of processes) extracted from specific case-studies. We exploit reductions to apply the model checker Spin to automatically validate finite instances of the model and to extract preconditions on the size of quorums used in the election phases of the protocol. We discuss verification results obtained via different optimizations we obtained by a careful design of the Promela specification.

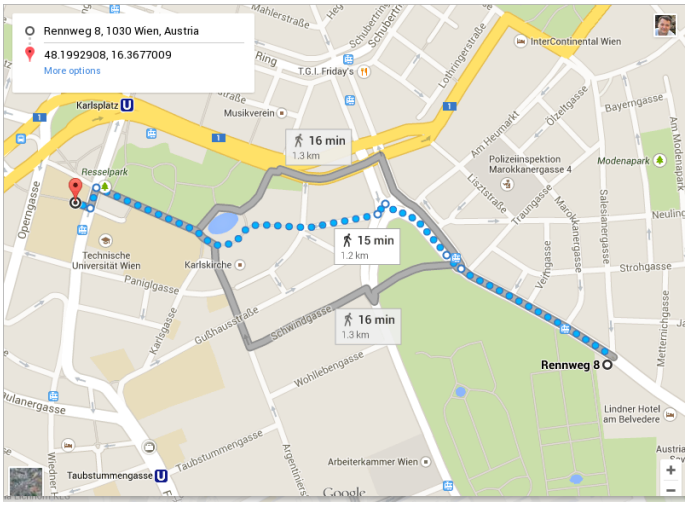
Workshop Dinner at Salm Bräu Brewery

19:00 – ∞

Address: 1030, Rennweg 8
URL: `www.salmbraeu.com`

How to get there:

- Either take tram 71. Hop on "*Wien Kärntner Ring/Oper*" and hop off at "*Unteres Belvedere*".
- Or walk for 15-20 min. from *Freihaus*, where the workshop takes place (check with the route below).



Day 2 (July 24)

Alexander Shvartsman : Invited talk — **9:00 – 10:15**

Specifying, Reasoning About, Optimizing, and Implementing Atomic Data Services for Distributed Systems

Abstract. Consistent shareable data services supporting atomic (linearizable) objects provide convenient building blocks for dynamic distributed systems. In general it is notoriously challenging to combine provable correctness guarantees with efficiency in distributed systems subject to perturbations in the computing medium. We overview our work on specification and implementation of distributed system services. Then we focus on a general framework for dynamic consistent data services that can be tailored to yield implementations for various target network settings and that incorporates on-the-fly reconfiguration that only modestly interferes with on-going operations. Here the goal is to guarantee safety (atomicity) for arbitrary patterns of asynchrony, crashes, and message loss, while enabling practical implementations. We describe examples of specification, rigorous reasoning about correctness, provable optimizations, and methodical implementations of consistent data services in distributed systems.

Coffee break

10:15 – 10:45

Tobias Gawron-Deutsch — **10:45 – 11:15**

Verification and Validation Challenges in Smart Grids from the Industrial Perspective

Abstract. We discuss several characteristics of smart grids, and show that at several crucial parts, we rely on correct and efficient operation of (distributed) computerized control systems. We discuss requirements of these systems, the necessary protocols, and present possible new application domains for protocol design, automated validation, and automated verification from an industrial perspective.

Mike Dodds, Andreas Haas and Christoph Kirsch — **11:15 – 11:45**

A Fast, Correct Time-Stamped Stack

Abstract. Efficient and scalable concurrent data-structures are key to high-performance multicore systems. Algorithms such as stacks and queues are widely used to handle synchronisation and distribute work between cores. We present the TS stack, a new concurrent stack implementation which allows elements to remain unordered in the stack if they were pushed concurrently. We use a new proof technique to show the correctness of the TS stack according to linearizability as the common linearization point technique cannot be applied.

11:45 – 12:15 Victor Altukhov, Eugene Chemeritskiy, **Vladislav Podymov**, and Vladimir Zakharov

Models and techniques for verification of Software Defined Networks

Abstract. Software-defined networks (SDNs) is a new type computer networks where data planes and control planes are separated from each other and a centralized controller manages a distributed set of switches. A set of commands for packet forwarding and flow-table updating was defined in the form of a protocol known as OpenFlow. SDNs can both simplify existing network applications and serve as a platform for developing new ones. The main advantage of this network architecture is that programmers are able to control the behaviour of the whole network by configuring appropriately the packet forwarding rules installed on the switches. Nevertheless, correct and safe management of SDNs is not an easy task. Every time the current load of flow tables should satisfy certain requirements: some packets have to reach their destination, although some other packets have to be dropped, certain switches are forbidden for some packets, whereas some other switches have to be obligatorily traversed. These and some other requirements constitute a Packet Forwarding Policies (PFPs). One of the aims of network engineering is to provide such a loading of switches with forwarding rules as to guarantee compliance with the PFP. The solution is to develop a toolset which could be able 1) to check correctness of a separate application operating on the controller w.r.t. a specified forwarding policy, 2) to check consistency of forwarding policies implemented by various applications, and 3) to monitor and check correctness and safety of the entire SDN. In an attempt to produce such a toolset we developed 1) a combined (relational and automata based) formal model which captures the most essential features of SDN behavior, 2) a multi-level formal language for specification of SDN forwarding policies which uses transitive closure operator to specify reachability properties of packet forwarding relation and temporal operators to specify behaviour of SDN as a whole, and 3) a BDD-based model-checking techniques for verification of SDN models against PFP specifications.

Discussion

12:15 – 12:45

Lunch

13:00 – 14:30

14:30 – 15:00 Cezara Dragoi, Josef Widder, and **Damien Zufferey**

Round model for distributed algorithms: from verification to implementation

Abstract. Fault-tolerant distributed algorithms are central to ensuring reliability and availability of many on-line services that we use on a daily basis. However, they are difficult to build correctly. On top of the usual implementation challenges, one has to consider non-determinism in the scheduling of the participants and faults such as machine failures, and dropped messages. Using round models as a layer of abstraction simplify the design of such systems. Rounds are communication-closed and have a deterministic schedule. From a theoretical perspective, those models are justified by partial synchrony assumptions and simulation relations that map non-determinism to faults. From a practical perspective, many asynchronous algorithms are structured around logical rounds. For instance, in the Paxos algorithm, a replica becomes a leader in the first round and proposes its value during the second round. However, rounds are an abstraction and real systems are asynchronous. Therefore, such models are popular in theoretical publications, but less so in system ones. This talk will describe ongoing efforts in the automated verification of round-based algorithms and the synthesis of asynchronous implementations from round-based descriptions.

Laure Millet, Maria Gradinariu Potop-Butucaru, Nathalie Sznajder and Sebastien Tixeuil

15:00 – 15:30

On the Synthesis of Mobile Robots Algorithms: the Case of Ring Gathering

Abstract. Recent advances in Distributed Computing highlights models and algorithms for autonomous swarms of mobile robots that self-organize and cooperate to solve global objectives. The overwhelming majority of works so far considers handmade algorithm and correctness proofs.

This paper is the first to propose a formal framework to automatically design distributed algorithms that are dedicated to autonomous mobile robots evolving in a discrete space. As a case study, we consider the problem of gathering all robots at a particular location, not known beforehand. Our contribution is threefold. First, we propose an encoding of the gathering problem as a reachability game. Then, we automatically generate an optimal distributed algorithm for three robots evolving on a fixed size uniform ring. Finally, we prove by induction that the generated algorithm is also correct for any ring size except when an impossibility result holds (that is, when the number of robots divides the ring size).

Sasha Rubin

15:30 – 16:00

Parameterised Verification of Robot Protocols: An Automata Theoretic Approach

Abstract. I propose a framework for the automatic verification of autonomous mobile agents (robot protocols), moving on unknown finite graphs. The framework reduces the parameterised verification problem (i.e., does a given set of robots solve a given task on all graphs from a given infinite set of graphs) to questions in automata theory and monadic second order logic. I will provide some preliminary results, discuss the limitations of this approach, and suggest future research directions.

Coffee break

16:00 – 16:30

Joel Rybicki

16:30 – 17:00

Synthesizing self-stabilizing fault-tolerant algorithms (extended abstract)

Abstract. In this talk, I will present our recent work on the synthesis of self-stabilizing, Byzantine fault-tolerant distributed algorithms for the synchronous counting problem.

We have examined two types of SAT solver-based techniques for synthesis. In the first approach, we have encoded the search problems into CNF SAT instances. This allows us to use highly-optimized, off-the-shelf SAT solvers. The second approach is inspired by CEGAR-style bounded model checking techniques. We have developed a search algorithm based on incremental SAT solving using assumptions on top of MiniSAT.

Both techniques have yielded novel algorithms for the synchronous counting problem. During the talk, I will present these results, discuss the trade-offs between the techniques for synthesis, and highlight open problems for future research.

This is joint work with Danny Dolev, Keijo Heljanko, Matti Järvisalo, Janne H. Korhonen, Christoph Lenzen, Ulrich Schmid, Jukka Suomela and Siert Wieringa.

17:00 – 17:30

Swen Jacobs

Parameterized Synthesis

Abstract. We present our work on the synthesis of finite-state component implementations for concurrent systems with a parametric number of components. We show how the parameterized synthesis problem can be reduced to a synthesis problem for a fixed number of components, using cutoff results from parameterized model checking. To make the approach practical for synthesis, we both extend the existing cutoff results, and optimize the synthesis method for the class of systems and specifications under consideration. With this approach, we have recently solved a parameterized version of a well-known industrial case study for the first time: we synthesize a component implementation for a (simplified) AMBA bus controller, such that copies of this component can be composed to obtain correct controllers for an arbitrary number of clients.

17:30 – 18:00

Nicolas Braud-Santoni and Swen Jacobs

Synthesis for Resilient Distributed Systems

Abstract. The implementation of distributed algorithms is error-prone, and manually constructed programs may not be optimal with respect to desired properties like size or stabilisation time. We propose to use computational methods to synthesise distributed algorithms from formal specifications, with guaranteed correctness and optimality. We present novel synthesis approaches for two general classes of specifications, and show that they are complete for realisable specifications and guaranteed to terminate for given bounds on certain parameters of the desired implementation. The resulting system components are correct by construction for distributed systems of arbitrary size, and have strong failure-resilience properties.
