DAGSTUHL
REPORTS

Volume 13, Issue 3, March 2023

Dagstuhl Reports, Vol. 13, Issue 3 ISSN 2192-5283

ISSN 2192-5283

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at
https://www.dagstuhl.de/dagpub/2192-5283

Publication date
October, 2023

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at `https://dnb.d-nb.de`.

Digital Object Identifier: 10.4230/DagRep.13.3.i

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:

- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

This basic framework can be extended by suitable contributions that are related to the program of the seminar, e. g. summaries from panel discussions or open problem sessions.

Contact
Schloss Dagstuhl – Leibniz-Zentrum für Informatik
Dagstuhl Reports, Editorial Office
Oktavie-Allee, 66687 Wadern, Germany
reports@dagstuhl.de
https://www.dagstuhl.de/dagrep

Report from Dagstuhl Seminar 23101

Foundations of WebAssembly

Karthikeyan Bhargavan[*1], **Jonathan Protzenko**[*2], **Andreas Rossberg**[*3], and **Deian Stefan**[*4]

1 **INRIA – Paris, FR.** `karthikeyan.bhargavan@inria.fr`
2 **Microsoft – Redmond, US.** `jonathan.protzenko@gmail.com`
3 **München, DE.** `rossberg@mpi-sws.org`
4 **University of California – San Diego, US.** `deian@cs.ucsd.edu`

Abstract

WebAssembly (Wasm) is a new portable code format with a formal semantics whose popularity has been growing fast, as a platform for new application domains, as a target for compilers and languages, and as a subject of research into its semantics, its performance, and its use in building verified and secure systems. This Dagstuhl Seminar brought together leading academics and industry representatives currently involved in the design, implementation and formal study of Wasm, to exchange ideas around topics such as formal methods for, verified compilation to, and verified implementation of Wasm.

Seminar March 5–10, 2023 – https://www.dagstuhl.de/23101

2012 ACM Subject Classification Software and its engineering → Formal language definitions; Software and its engineering → Virtual machines; Theory of computation → Semantics and reasoning

Keywords and phrases Compilation, Formal methods, Programming languages, Verification, Virtual machines, WebAssembly

Digital Object Identifier 10.4230/DagRep.13.3.1

1 Executive Summary

Andreas Rossberg

WebAssembly – commonly known as Wasm – is a modern, portable code format and execution environment with a formal semantics that enforces safety and isolation. Though initially designed to run native, high-performance applications in Web browsers, Wasm is now used in many other applications domains – from CDNs to serverless, IoT, library sandboxing, and smart contracts. Wasm is one of the rare cases where practitioners are collaborating with the semantics and programming languages research community. This was exemplified by the initial design of Wasm itself, a collaboration with academia that culminated in a PLDI paper. The popularity of Wasm has since been growing exponentially as a platform for new application domains, as a target for compilers and languages, and as a subject of active scientific research – from its future semantics to its performance, and its use in building verified and secure systems.

This Dagstuhl Seminar brought together leading academics and industry representatives currently involved in the design, implementation and formal study of Wasm. It was a forum to exchange ideas that set new directions for WebAssembly research. The main focus was around three topics:

[*] Editor / Organizer

Foundations of WebAssembly, *Dagstuhl Reports*, Vol. 13, Issue 3, pp. 1–16
Editors: Karthikeyan Bhargavan, Jonathan Protzenko, Andreas Rossberg, and Deian Stefan
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Formal methods for Wasm revolves around formalizing, reasoning and proving properties about Wasm itself. There are many WebAssembly extensions (e.g., bulk memory operations and vector instructions) which can benefit from formal semantics. Since Wasm is not a standalone language, there also is need to develop formal methods to reason about its interaction with the operating system, the execution of JITed Wasm code, etc. Finally, logics are needed that will allow us to formally capture interesting properties beyond what current work handles.

Verified Compilation to Wasm focusses on Wasm as a target of verified compilation toolchains. Wasm is positioned as a viable candidate for verified and secure compilation and we established that the clean design of Wasm offers greater simplicity when it comes to verifying a compilation toolchain – in particular, simpler and shorter proofs of compiler correctness and security.

Verified Compilation of Wasm studies the compilation of WebAssembly to native code, i.e., how to securely and correctly compile WebAssembly code to machine code. Wasm is growing rapidly, and is used on the Web and beyond (e.g., embedded systems, edge computing, IoT, and even OS kernels), and across different platforms and toolchains.

One particularly noteworthy result of the seminar was the birth of a new project that resulted in a collaboration between various participants of the seminar: to create a domain-specific language (DSL) for authoring the official Wasm specification. This project will enable creating a single source of truth for generating both the formalism and the alternative prose description in the standard, as well as transformations to representations in various theorem provers or executable reference interpreters that process the Wasm semantics for formal methods.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 RichWasm: Bringing Shared Memory Interoperability to WebAssembly

Amal Ahmed (Northeastern University – Boston, US)

Joint work of Amal Ahmed, Michael Fitzgibbons, Zoe Paraskevopoulou, Michelle Thalakottur, Noble Mushtak, Jose Sulaiman Manzur

Though Wasm provides a safe, sandboxed environment for programs to run in, it lacks the facilities to enable safe, shared-memory interoperability between Wasm modules, a feature that we believe is essential for a low-level language in a multi-language world. I'll present RichWasm, a higher-level version of WebAssembly with an enriched capability-based type system to support fine-grained type-safe shared-memory interoperability. RichWasm is rich enough to serve as a typed compilation target for both typed garbage-collected languages and languages with an ownership-based type system and manually managed memory. RichWasm takes inspiration from earlier work on languages with linear capability types to support safe strong updates, and adds analogous unrestricted capability types for garbage-collected locations, allowing a module to provide fine-grained memory access to another module, regardless of memory-management strategy. RichWasm types are not intended to be made part of core Wasm; instead we compile RichWasm to core Wasm, allowing for use in existing environments. We have formalized RichWasm in Coq and are currently proving its safety via progress and preservation.

3.2 Wasocaml: compiling OCaml to WebAssembly

Léo Andrès (University Paris-Saclay – Orsay, FR)

Joint work of Léo Andrès, Pierre Chambart

OCaml is a rich programming language. It is comprised of a lot of advanced functional and imperative features while allowing low-level manipulations. Our talk will begin with a description of the value representation technique as well as the memory layout used by the OCaml runtime. We will then examine the distinctions between various intermediate representations in the OCaml compiler, and then justify the selection of Flambda as a source language. Additionally, we will present our translation process from Flambda to wasm-gc, with a particular focus on the encoding of small scalars, heap-allocated blocks and functions closures. To top it off, we will provide a comparative analysis of our compiler against the alternatives, based on informative benchmarks.

3.3 WebAssembly Diversification for Malware Evasion

Javier Cabrera Arteaga (KTH Royal Institute of Technology – Stockholm, SE)

Joint work of Javier Cabrera Arteaga, Martin Monperrus, Benoit Baudry
Main reference Javier Cabrera-Arteaga, Martin Monperrus, Tim Toady, Benoit Baudry: "WebAssembly Diversification for Malware Evasion", CoRR, Vol. abs/2212.08427, 2022.
URL https://doi.org//10.48550/arXiv.2212.08427

WebAssembly is an important binary format that has become an integral part of the modern web. This technology offers a faster alternative to JavaScript in web browsers, but it has also been utilized for cryptojacking since its inception. To counter this threat, considerable efforts have been made to develop defenses that can detect WebAssembly malware. However, these defenses have not taken into account the possibility that attackers may use complex evasion techniques. We explore how to evade detection by WebAssembly cryptojacking detectors. We propose a technique that uses wasm-mutate, a fuzzing tailored tool of wasmtime, to create variants of the original code that can evade the detectors and demystify the previous assumption. To evaluate our technique, we used VirusTotal. Our results demonstrate that our approach swiftly generates WebAssembly cryptojacking variants that evade detection, while the generated WebAssembly binaries show only minimal performance overhead. Our experiments also provide valuable insights into which WebAssembly code transformations are best suited for evading malware detection. This knowledge can be used to improve the state of the art in WebAssembly malware detection, which will benefit the wider community. Although our technique exposes weaknesses in detection mechanisms, it also serves as a valuable tool for testing other systems using WebAssembly as an input, e.g. compilers, validators and verification tools.

3.4 From Dynamic to Static Symbolic Execution for WebAssembly

José Fragoso Santos (INESC-ID – Lisbon, PT)

Joint work of José Fragoso Santos, Filipe Marques, Nuno Santos, Pedro Adão

We present WASP, a configurable symbolic execution engine for analysing Wasm modules. WASP works directly on Wasm code and is built on top of the official Wasm reference interpreter. One key advantage of WASP compared to other symbolic execution engines is that it is highly configurable, supporting various flavours of symbolic execution and exploration strategies of the program's state space.

Using WASP, we created WASP-C, a new symbolic execution framework for testing C programs. WASP-C was used to symbolically test a generic data-structure library for C and the Amazon Encryption SDK, demonstrating that it can find new bugs and generate high-coverage testing inputs for real-world C code. WASP-C was further tested against the Test-Comp 2022/2023 benchmarks, obtaining results comparable to well-established symbolic execution and testing tools for C.

3.5 WasmCert-Coq: A Mechanised Specification of WebAssembly

Philippa Gardner (Imperial College London, GB)

Milner pioneered formal language specification, proving hand-written correctness results about type safety and module instantiation. His work led to many formal then mechanised specifications including the large Coq-mechanisation of JavaScript, JSCert [1], developed at Imperial and Inria. Most of these large mechanised specifications were developed long after the language standards had been essentially settled. The challenge now is to establish mechanised language specification within the language standardisation process.

The W3C WebAssembly (Wasm) language is the first programming language to have a formal standard as envisaged by Milner. Inspired by JSCert, Gardner and Watt developed the mechanised specification of the Wasm 1.0 standard, WasmCert-Coq and WasmCert-Isabelle [2]: crucially, Watt fixed errors in the specification and type-safety result before the Wasm draft publication [3], adapting ideas from JSCert; correctness of module instantiation was proved in WasmCert.

In Conrad's Dagstuhl talk, he will present WasmRef-Isabelle [4], an efficient certified reference interpreter for Wasm 1.0 supported by the ByteCode Alliance. In this talk, I will present WasmCert-Coq and explore how to define a certified reference interpreter, WasmRef-Coq, in such a way that the definitions and correctness proofs might have a better chance to keep up with the evolving standard in future.

References

1 Martin Bodin, Arthur Charguèraud, Daniele Filaretti, Philippa Gardner, Sergio Maffeis, Daiva Naudžiūnienė, Alan Schmitt and Gareth Smith. *A Trusted Mechanised JavaScript Specification.* Principles of Programming Languages (POPL), 2014

2 Watt, Rao, Pichon-Pharabod, Bodin and Gardner. *Two Mechanisations of WebAssembly 1.0.* Formal Methods (FM), 2021

3 Conrad Watt. *Mechanising and Verifying the WebAssembly Specification.* Certified Programs and Proofs (CPP), 2018

4 Watt, Trela, Lammich, Märki. *WasmRef-Isabelle: a Verified Monadic Interpreter and Industrial Fuzzing Oracle for WebAssembly.* Programming Language Design and Implementation (PLDI), 2023

3.6 Iris-Wasm, a mechanized separation logic for WebAssembly

Aïna Linn Georges (Aarhus University, DK)

Joint work of Xiaojia Rao, Aïna Linn Georges, Maxime Legoupil, Conrad Watt, Jean Pichon-Pharabod, Philippa Gardner, Lars Birkedal
Main reference Xiaojia Rao, Aïna Linn Georges, Maxime Legoupil, Conrad Watt, Jean Pichon-Pharabod, Philippa Gardner, Lars Birkedal: "Iris-Wasm: Robust and Modular Verification of WebAssembly Programs", Proc. ACM Program. Lang., Vol. 7(PLDI), Association for Computing Machinery, 2023.
URL https://doi.org//10.1145/3591265

Iris-Wasm is a mechanized separation logic that we have developed and used to practically verify both individual Wasm programs and properties of Wasm itself. In this talk, we will explore how Iris enables the specification and verification of individual modules separately, which can then be combined modularly to reason about complex programs. Additionally, we will demonstrate how Iris enables the verification of functional correctness in WebAssembly programs, even when they interact with unknown or adversarial code, demonstrating the promise of WebAssembly's module isolation. Iris is a rich and expressive higher-order separation logic that provides a powerful toolset for program verification. By successfully instantiating the full language standard into Iris, we can, going forward, leverage its numerous applications, such as exploring weak memory, robust safety, capabilities, effects, secure compilation, garbage collection, and more. Thus, we open up many exciting prospects for the verification of WebAssembly programs, and create an expressive foundational support for the WebAssembly ecosystem. In this talk, I will present a high level overview of Iris-Wasm, demonstrating the kind of verification it enables and providing a practical demonstration of what a proof in Iris-Wasm looks like.

3.7 Flexible and Secure Hardware-Assisted Wasm with HFI

Shravan Narayan (University of California – San Diego, US), Evan Johnson (University of California – San Diego, US), and Deian Stefan (University of California – San Diego, US)

Joint work of Shravan Narayan, Tal Garfinkel, Mohammadkazem Taram, Joey Rudek, Daniel Moghimi, Evan Johnson, Chris Fallin, Anjo Vahldiek-Oberwagner, Michael LeMay, Ravi Sahita, Dean Tullsen, Deian Stefan
Main reference Shravan Narayan, Tal Garfinkel, Mohammadkazem Taram, Joey Rudek, Daniel Moghimi, Evan Johnson, Chris Fallin, Anjo Vahldiek-Oberwagner, Michael LeMay, Ravi Sahita, Dean M. Tullsen, Deian Stefan: "Going beyond the Limits of SFI: Flexible and Secure Hardware-Assisted In-Process Isolation with HFI", in Proc. of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3, ASPLOS 2023, Vancouver, BC, Canada, March 25-29, 2023, pp. 266–281, ACM, 2023.
URL https://doi.org//10.1145/3582016.3582023

In this talk, I will introduce Hardware-assisted Fault Isolation (HFI), a simple extension to existing processors to support secure, flexible, and efficient in-process isolation. HFI addresses the limitations of ex- isting software-based isolation (SFI) systems including: runtime overheads, limited scalability, vulnerability to Spectre attacks, and limited compatibility with existing code. HFI can seamlessly in- tegrate with current SFI systems (e.g., WebAssembly), or directly sandbox unmodified native binaries. To ease adoption, HFI relies only on incremental changes to the data and control path of existing high-performance processors. I will also cover our evaluation of HFI for x86-64 using the gem5 simulator and compiler-based emulation on a mix of real and synthetic workloads

3.8 Let's Go Coroutine

Luna Phipps-Costin (Northeastern University – Boston, US) and Daniel Hillerström (Huawei Technologies – Zürich, CH)

Joint work of Arjun Guha, Daan Leijen, Sam Lindley, Matija Pretnar, Andreas Rossberg, KC Sivaramakrishnan

Non-local control flow features provide the ability to suspend the current execution context and later resume it. Many industrial-strength programming languages feature a wealth of non-local control flow features such as async/await, coroutines, generators/iterators, effect handlers, call/cc, and so forth. For some programming languages non-local control flow is central to their identity, meaning that they rely on non-local control flow for efficiency, e.g. to support massively scalable concurrency. Currently, WebAssembly lacks support for implementing such features directly and efficiently without a circuitous global transformation of source programs on the producer side. During this talk we will introduce WasmFX, an extension of Wasm with effect handlers for handling non-local control-flow in a structured manner. We will demonstrate WasmFX by example by compiling and running some coroutine programs live (uh-oh). We will also discuss the implementation of WasmFX in wasmtime, and the future directions that we are looking to explore.

3.9 Wasm 2.0, 2.1 and beyond

Andreas Rossberg (München, DE)

Since the release of Wasm 1.0, many formal proposals for language extensions have been and are still being developed. A first batch has been included as part of Wasm 2.0, another one is soon expected with Wasm 2.1. I gave an overview of the extensions already adopted, those nearing completion, and those still under active development, and briefly touched on the wider implications they might have on Wasm's semantics. I also explained the proposal process itself and its requirements, for those who are not following the Wasm CG closely.

3.10 How to design, document, and implement programming languages

Sukyoung Ryu (KAIST – Daejeon, KR)

Since 2015, the JavaScript language has rapidly evolved with a yearly release cadence and open development process. However, it results in the gap between the language specification written in English and tools, such as parsers, interpreters, and static analyzers, which makes language designers and tool developers suffer from manually filling the gap. JISET and its extensions lessen the burden by automatically extracting a mechanized specification from the language specification in prose.

We introduce several tools in the JISET family and show how they fill the gap between the language specification and tools. We then discuss how to apply this technique to WebAssembly.

3.11 Wanilla: Sound Automated Horn-clause-based Noninterference Analysis for WebAssembly

Markus Scherer (TU Wien, AT)

Joint work of Jeppe Fredsgaard Blaabjerg, Markus Scherer, Magdalena Solitro, Alexander Sjösten, Matteo Maffei

Noninterference is an important information flow property that can be formalized as 2-safety-property. In this talk we will explore how to leverage horn-clause-based abstractions to overapproximate it as a reachability property. Having done so, we can use solvers for SMT's Constrained-Horn-Clause fragment to assess noninterference in WebAssembly, a compilation target widely used in the real world. Our approach aims to be sound and automated at the same time which forces us to carefully balance runtime performance and precision.

3.12 That's a Tough Call! Studying the Challenges of Call Graph Construction for WebAssembly

Michelle Thalakottur (Northeastern University – Boston, US), Daniel Lehmann (Universität Stuttgart, DE & Google – München, DE), Michael Pradel (Universität Stuttgart, DE), and Frank Tip

Call graphs are at the core of many inter-procedural static analysis and optimization techniques. However, WebAssembly poses some unique challenges for static call graph construction. Currently, these challenges are neither well understood, nor is it clear to what extent existing techniques address them. We systematically study WebAssembly specific challenges for static call graph construction and identify and classify 12 challenges. We then measure their prevalence in real-world binaries. We also study the soundness and precision of four existing static analyses. Our findings include that, surprisingly, all of the existing techniques are unsound, without this being documented anywhere. We envision our work to provide guidance for improving static call graph construction for WebAssembly.

3.13 WebAssembly as the Basis of All Things?

Ben L. Titzer (Carnegie Mellon University – Pittsburgh, US)

WebAssembly is a low-level, portable bytecode offering a compilation target with near-native performance. Now a standard feature of all web browsers, Wasm has started to expand to many other applications such as edge computing, distributed cryptographic digital contracts, networking stacks, and more. As Wasm gains features through the standardization process, it becomes a more attractive target for new kinds of languages. In this talk I will fast-forward to look at a whole new set of language runtime system designs that are made possible with some new features that can be added to Wasm. In particular, can we build a *really* fast language implementation without having to write a new JIT compiler? A new code format (with validator)? A new garbage collector? Can we do this without ever having to look at assembly language? I hope so! Let's look at what I've discovered and what I think that means.

3.14 Verifying Instruction Selection in a Wasm-to-native Compiler

Alexa VanHattum (Cornell University – Ithaca, US)

Joint work of Alexa VanHattum, Monica Pardeshi, Chris Fallin, Adrian Sampson, Fraser Brown

For ahead-of-time or just-in-time compilation, Wasm's sandboxing guarantees rely on the correctness of the generated native assembly. Subtle wrong-code bugs in native instruction selection can introduce security flaws. In this talk, I'll present our efforts toward automated verification for instruction lowering rules within Cranelift, a production code generator for Wasmtime. I'll discuss our approach to modeling the Cranelift intermediate representation and ARM aarch64 backend, challenges with generalizing over types, and several case studies of faults analyzed by our tool.

3.15 MSWasm: Soundly Enforcing Memory-Safe Execution of Unsafe Code

Marco Vassena (Utrecht University, NL)

Joint work of Alexandra E. Michael, Anitha Gollamudi, Jay Bosamiya, Evan Johnson, Aidan Denlinger, Craig Disselkoen, Conrad Watt, Bryan Parno, Marco Patrignani, Marco Vassena, Deian Stefan
Main reference Alexandra E. Michael, Anitha Gollamudi, Jay Bosamiya, Evan Johnson, Aidan Denlinger, Craig Disselkoen, Conrad Watt, Bryan Parno, Marco Patrignani, Marco Vassena, Deian Stefan: "MSWasm: Soundly Enforcing Memory-Safe Execution of Unsafe Code", Proc. ACM Program. Lang., Vol. 7(POPL), pp. 425–454, 2023.
URL https://doi.org//10.1145/3571208

Most programs compiled to WebAssembly (Wasm) today are written in unsafe languages like C and C++. Unfortunately, memory-unsafe C code remains unsafe when compiled to Wasm—and attackers can exploit buffer overflows and use-after-frees in Wasm almost as easily as they can on native platforms. This talk presents Memory-Safe WebAssembly (MSWasm), an extension of Wasm with language-level memory-safety abstractions to precisely address this problem. In the talk, we will discuss the design of MSWasm and show how compilers can leverage these abstractions to automatically eliminate memory vulnerabilities from unsafe code. We have developed a C-to-MSWasm compiler on top of Clang and two compilers of MSWasm to native code, which support different enforcement mechanisms, and thus allow developers to make security-performance trade-offs according to their needs. More importantly, MSWasm's design makes it easy to swap between enforcement mechanisms; as fast (especially hardware-based) enforcement techniques become available, MSWasm will be able to take advantage of these advances almost for free.

3.16 The Path to Components

Luke Wagner (Fastly – San Francisco, US)

This talk described the motivation for starting work on a new set of standards focused on portably and securely executing WebAssembly outside the browser, viz., WASI and the Component Model. Next, the talk covered the motivation for not simply adopting the well-established design of POSIX by identifying 4 areas where POSIX has performance and composability problems, viz., around linking, the passing of high-level values between processes, the handling of external resources and the basis for concurrency. Lastly, the talk gave a quick preview of how the tooling could work in practice and be shared across a heterogeneous ecosystem of platforms executing WebAssembly.

3.17 Usefully Mechanising All of WebAssembly

Conrad Watt (University of Cambridge, GB)

This talk will describe WasmCert-Isabelle, an Isabelle/HOL mechanisation of the WebAssembly specification; and recent work on developing WasmRef-Isabelle, a practically-useful reference implementation of WebAssembly that is verified with respect to this mechanisation. We describe our successes in driving WasmRef-Isabelle's adoption as a fuzzing oracle for the widely-used Wasmtime implementation of WebAssembly, as well as the challenges we will face in keeping our work up to date with the ever-evolving WebAssembly standard.

4 Working groups

4.1 A DSL for writing the WebAssembly Specification

Andreas Rossberg (München, DE), Joachim Breitner (Freiburg, DE), Pierre Chambart (Société OCamlPro SAS – Paris, FR), Philippa Gardner (Imperial College London, GB), Sam Lindley (University of Edinburgh, GB), Matija Pretnar (University of Ljubljana, SI), Xlaojia Rao, Sukyoung Ryu (KAIST – Daejeon, KR), Luke Wagner (Fastly – San Francisco, US), Conrad Watt (University of Cambridge, GB), and Dongjun Youn (KAIST – Daejeon, KR)

Motivation

To standardise a WebAssembly feature, the following specification artefacts must be produced:

- a formal specification of the feature in LaTeX
- a prose description of the feature
- a reference implementation in OCaml

This process is onerous and several important upcoming WebAssembly features such as Threads and Exception Handling have not yet been standardised purely because they do not meet these requirements, despite widespread industrial support, implementation, and use. In addition, inconsistencies between these definitions can lead to divergences in implementations[1].

Moreover, academic mechanisations of WebAssembly such as WasmCert-Isabelle[2] and WasmCert-Coq[3] must be updated with each new feature if they wish to remain correspondant to the current version of WebAssembly. Even when this effort is undertaken, academics working with other popular theorem provers such as Lean and Agda are not be able to make use of these models and would need to write their own from scratch in order to carry out mechanisation-related research on WebAssembly.

We propose to develop a machine readable domain-specific language (DSL) that will function as a unified source of truth for the WebAssembly specification, and for which we can define a number of separate *backends* to generate not only the above specification artefacts, but also mechanisations in all major theorem provers. This will significantly improve the productivity of WebAssembly's industrial standards body, and allow academics to access a feature-complete mechanisation of WebAssembly no matter their preferred theorem prover.

Why not use an existing DSL?

There are a number of existing tools for writing language definitions [1, 2, 3], with backends that could generate LaTeX, parts of a reference interpreter, and stubs for proof assistants. We have considered using them for the Wasm specification, but ultimately decided against them – we believe that we will be able to generate higher-quality artefacts by building our DSL to intrinsically make use of domain-specific knowledge about the WebAssembly specification.

As an immediate example, it would likely be impossible to generate a prose description acceptable to WebAssembly's standards body from a generic IR. In addition, our initial work suggests that our generated interpreter will be significantly more efficient if we are able to make use of certain knowledge regarding the restricted structure of WebAssembly's evaluation contexts.

Moreover, the primary audience of the DSL is the existing specification authors, so we intend for the DSL to closely reflect the current style of the hand-written specification – for example the pervasive use of sequences and corresponding iterators. For the same reason, we have also quickly abandoned a prototype of an embedded DSL in OCaml[4].

Proposed solution

The solution proposed by Andreas Rossberg and quickly adopted by the whole group is SpecTec[5]. In it, the specification is written in a text format similar to the existing specification. The language consists of few generic concepts:

[1] https://github.com/WebAssembly/threads/issues/195

[2] https://github.com/WasmCert/WasmCert-Isabelle

[3] https://github.com/WasmCert/WasmCert-Coq

[4] https://github.com/matijapretnar/wasm-spec-dsl

[5] https://github.com/Wasm-DSL/spectec

- *Syntax definitions*, describing the grammar of the input language or auxiliary constructs. These are essentially type definitions for the object language. For example:

  ```
  syntax valtype = | I32 | I64 | F32 | F64
  syntax functype = valtype* -> valtype*
  syntax instr = | NOP | BLOCK instr* | IF instr* ELSE instr*
  syntax context = { FUNC functype*, LABEL (valtype*)* }
  syntax config = state; instr*
  ```

- *Variable declarations*, ascribing the syntactic class (i.e., type) that meta variables used in rules range over. For example:

  ```
  var t : valtype
  var ft : functype
  var ‘C : context
  ```

  (Also, every type name is implicitly usable as a variable of the respective type.)

- *Relation declarations*, defining the shape of judgement forms, such as typing or reduction relations. These are essentially type declarations for the meta language. For example:

  ```
  relation Instr_ok: context |- instr : functype
  relation Step: config ~> config
  ```

- *Rule definitions*, expressing the individual rules defining relations. For example:

  ```
  rule Instr_ok/nop:
    ‘C |- NOP : epsilon -> epsilon

  rule Instr_ok/if:
    ‘C |- IF instr_1* ELSE instr_2* : t_1* -> t_2
    -- InstrSeq_ok: ‘C, LABEL t_2* |- instr_1* : t_1* -> t_2*
    -- InstrSeq_ok: ‘C, LABEL t_2* |- instr_2* : t_1* -> t_2*

  rule Step/nop:
    z; NOP ~> z; epsilon

  rule Step/if-true:
    z; (I32.CONST c) (IF instr_1* ELSE instr_2*) ~> z; (BLOCK instr_1*)
    -- if c =/= 0
  rule Step/if-false:
    z; (I32.CONST c) (IF instr_1* ELSE instr_2*) ~> z; (BLOCK instr_2*)
    -- if c = 0
  ```

  Every rule is named, so that it can be referenced. Each premise is introduced by a dash and includes the name of the relation it is referencing, easing checking and processing.

- *Auxiliary Functions*, allowing to abstract complex conditions into separate definitions. For example:

  ```
  def $size(numtype) : nat
  def $size(I32) = 32
  def $size(I64) = 64
  def $size(F32) = 32
  def $size(F64) = 64
  ```

- *Hint annotations* that are uninterpreted by default, but may offer occasional extra guidance for different backends (eg. LaTeX macros to be used).

The implementation defines two AST representations: an *external language* (EL), which is close to the written specification and suitable for backends generating LaTeX, and an *internal language* (IL), suitable for backends generating programs. Elaboration from EL into IL infers additional information and makes it explicit in the representation:

- resolve notational overloading and mixfix applications,
- resolve overloading of variant constructors and annotate them with their type,
- insert injections from variant subtypes into supertypes,
- insert injections from singletons into options/lists,
- insert binders and types for local variables in rules and functions,
- mark recursion groups and group definitions with rules, ordering everything by dependency.

Progress since the seminar

Since the seminar, there has been significant progress on the implementation. In addition to the basic infrastructure and a LaTeX backend, there is a prototype backend generating prose English, as well as ones generating Agda, Coq and Lean code, and a number of passes that further elaborate the IL. Particular effort is being put into *animating* the operational semantics, i.e., transforming the rules into algorithmic steps that then can be used for generating both the prose part of the specification and a reference interpreter.

References

1 Martin Bodin, Philippa Gardner, Thomas P. Jensen, and Alan Schmitt. Skeletal semantics and their interpretations. *Proc. ACM Program. Lang.*, 3(POPL):44:1–44:31, 2019.

2 Dominic P. Mulligan, Scott Owens, Kathryn E. Gray, Tom Ridge, and Peter Sewell. Lem: reusable engineering of real-world semantics. In *ICFP*, pages 175–188. ACM, 2014.

3 Peter Sewell, Francesco Zappa Nardelli, Scott Owens, Gilles Peskine, Thomas Ridge, Susmit Sarkar, and Rok Strnisa. Ott: Effective tool support for the working semanticist. *J. Funct. Program.*, 20(1):71–122, 2010.

Participants

- Amal Ahmed
Northeastern University – Boston, US
- Léo Andrès
University Paris-Saclay – Orsay, FR
- Javier Cabrera Arteaga
KTH Royal Institute of Technology – Stockholm, SE
- Karthikeyan Bhargavan
INRIA – Paris, FR
- Joachim Breitner
Freiburg, DE
- Pierre Chambart
Société OCamlPro SAS – Paris, FR
- Martin Fink
TU München – Garching, DE
- Philippa Gardner
Imperial College – London, UK
- Aïna Linn Georges
Aarhus University, DK
- Arjun Guha
Northeastern University – Boston, US
- Reiner Hähnle
TU Darmstadt, DE
- Daniel Hillerström
Huawei Technologies – Zürich, CH
- Evan Johnson
University of California – San Diego, US
- Daniel Lehmann
Google – München, DE
- Sam Lindley
University of Edinburgh, UK
- Tyler McMullen
Fastly – San Francisco, US
- Lucy Menon
Northeastern University – Boston, US
- Shravan Narayan
University of California – San Diego, US
- Luna Phipps-Costin
Northeastern University – Boston, US
- Jean Pichon-Pharabod
Aarhus University, DK
- Michael Pradel
Universität Stuttgart, DE
- Matija Pretnar
University of Ljubljana, SI
- Jonathan Protzenko
Microsoft – Redmond, US
- Andreas Rossberg
München, DE
- José Fragoso Santos
INESC-ID – Lisbon, PT
- Claudio Russo
Dfinity – Cambridge, UK
- Sukyoung Ryu
KAIST – Daejeon, KR
- Markus Scherer
TU Wien, AT
- Sabine Schmaltz
Tarides – Saarbrücken, DE
- Till Schneidereit
Fermyon – Heidelberg, DE
- KC Sivaramakrishnan
Indian Institute of Technology – Madras, IN
- Deian Stefan
University of California – San Diego, US
- Michelle Thalakottur
Northeastern University – Boston, US
- David Thien
University of California – San Diego, US
- Ben Titzer
Carnegie Mellon University – Pittsburgh, US
- Alexa VanHattum
Cornell University – Ithaca, US
- Marco Vassena
Utrecht University, NL
- Luke Wagner
Fastly – San Francisco, US
- Conrad Watt
University of Cambridge, UK
- Dongjun Youn
KAIST – Daejeon, KR

Report from Dagstuhl Seminar 23111

Computational Complexity of Discrete Problems

Anna Gál*[1], Meena Mahajan*[2], Rahul Santhanam*[3], Till Tantau*[4], and Manaswi Paraashar†[5]

1 **University of Texas – Austin, US.** panni@cs.utexas.edu
2 **The Institute of Mathematical Sciences – Chennai, IN.** meena@imsc.res.in
3 **University of Oxford, GB.** rahul.santhanam@cs.ox.ac.uk
4 **Universität zu Lübeck, DE.** tantau@tcs.uni-luebeck.de
5 **Aarhus University, DK.** manaswi.isi@gmail.com

Abstract

This report documents the program and activities of Dagstuhl Seminar 23111 “Computational Complexity of Discrete Problems”, which was held in-person in March 2023 (the previous instance of the seminar series had been held online in March 2021). Following a description of the seminar’s objectives and its overall organization, this report lists the different major talks given during the seminar in alphabetical order of speakers, followed by the abstracts of the talks, including the main references and relevant sources where applicable. The return to an in-person setting allowed an intense atmosphere of active research and interaction throughout the five day seminar.

Seminar March 12–17, 2023 – https://www.dagstuhl.de/23111
2012 ACM Subject Classification Theory of computation → Probabilistic computation; Theory of computation → Streaming models; Theory of computation → Circuit complexity; Theory of computation → Pseudorandomness and derandomization
Keywords and phrases circuit complexity, communication complexity, computational complexity, lower bounds, randomness
Digital Object Identifier 10.4230/DagRep.13.3.17

1 Executive Summary

Anna Gál (University of Texas, Austin, US)
Meena Mahajan (The Institute of Mathematical Sciences, Chennai, IN)
Rahul Santhanam (University of Oxford, GB)
Till Tantau (Universität zu Lübeck, DE)

Computational complexity studies the amount of resources (such as time, space, randomness, parallelism, or communication) that are necessary to solve computational problems in various models of computation. Finding efficient algorithms for solving computational tasks is crucial in many practical applications. Despite a long line of research, for many discrete problems that arise in practice it is not known if they can be solved efficiently – in particular, in (randomized) polynomial time. While efficient algorithms clearly have obvious applications, knowing that a problem can*not* be solved efficiently can *also* have high practical impact. For example, lower bounds on the amount of resources needed to solve specific problems can be used to construct good pseudorandom generators to derandomize probabilistic algorithms. Similarly, the security of our currently used crypto-systems hinges on the *assumption* that

* Editor / Organizer
† Editorial Assistant / Collector

certain discrete problems – like factoring – are *hard* to solve; and we would very much like to *prove* that more efficient algorithms cannot exist for factoring. Proving lower bounds is a challenging task since one needs to argue against all possible algorithms. In the last few decades, lower bound methods have been developed for various restricted or special models of computation. These results often involve the use of sophisticated mathematical techniques and despite a lot of effort we still do not have – somewhat frustratingly – strong enough techniques to establish for instance superlinear lower bounds for specific problems in general computational models, such as the model of Boolean circuits. In this Dagstuhl Seminar, which is the most recent incarnation of a seminar series that stretches back many years, we brought together leading experts and talented junior researchers to discuss the most exciting recent developments in different areas of computational complexity of discrete problems – both regarding recent *results,* but also regarding *open problems.* In both cases, a particular focus was on lower bounds and on whether and, if so, how ideas and methods from one theory area can yield insights in another theory area.

To enable and encourage discussions between the researchers present in Dagstuhl, time was allotted to three different formats: The presentation and discussion of current research results and methods, the presentation and discussion of open problems and conjectures, and on-site collaborative theory research. Each day of the seminar started with a morning session dedicated to survey talks, talks sharing research results, and talks introducing new techniques (the titles and abstracts of most of these talks appear later in this report). The afternoons were dedicated to collaborative research in various forms: On Tuesday, research was done in break-out sessions in which smaller groups of participants explored different open research questions in depth. The topics covered were the following (in alphabetical order of chairs):

1. *Tree Codes,* chaired by Gil Cohen.
2. *Lifting Dichotomies,* chaired by Yuval Filmus.
3. *Finding Tarski Fixed-Points,* chaired by Kristoffer Hansen.
4. *Hitting Sets Versus Orthogonal Vectors (a.k.a. k*SAT *Versus* $\forall\exists k$SAT*),* chaired by Marvin Kühnemann.
5. *Simple Versions of #*SAT*,* chaired by Till Tantau.

Two other formats were intended to intrigue participants in research questions beyond their own speciality (and succeeded in doing so). Firstly, there were open problem sessions, where Gil Cohen, Yuval Filmus, Mika Göös, Rohit Gurjar, Alexander Kulikov, Jakob Nordström, Rüdiger Reischuk, Robert Robere, and Ben Lee Volk presented different research questions. Secondly, at various points during the seminar, there were short “talks to talk about”, which introduced exciting recent topics, results, or problems and which gave people intriguing topics to discuss over lunch or dinner. Talks to talk about were given by Sourav Chakraborty, Manaswi Parashar, and Till Tantau.

A final important objective of the seminar was to foster collaborations not only between researchers working on different topics, but also between junior and senior researchers. Towards this aim, talks of more junior researchers were scheduled (as far as possible) on the first day, giving them early exposure and allowing other participants to talk to them about the presented research results during the whole week. Naturally, both junior and senior participants had ample opportunity to socialize, be it during the traditional Wednesday afternoon hike or the wine-and-cheese party on Thursday.

The organizers, Anna Gál, Meena Mahajan, Rahul Santhanam, and Till Tantau, thank all participants for the many contributions they made. We would also like to especially thank the Dagstuhl staff, who were – as usual – extremely friendly, helpful, and professional regarding all organizational matters surrounding the seminar. Finally, we express our great gratitude to Manaswi Paraashar for his invaluable help assembling and preparing this report.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

Open problems

3 Overview of Talks

3.1 Models of CDCL solving for quantified Boolean formulas

Olaf Beyersdorff (Friedrich-Schiller-Universität Jena, DE)

Joint work of Benjamin Böhm, Tomas Peitl, Olaf Beyersdorff
Main reference Benjamin Böhm, Tomás Peitl, Olaf Beyersdorff: "QCDCL with Cube Learning or Pure Literal Elimination – What is Best?", in Proc. of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022, pp. 1781–1787, ijcai.org, 2022.
URL https://doi.org//10.24963/ijcai.2022/248

This talk explained the relations between solvers based on the conflict-driven clause learning (CDCL) paradigm for quantified Boolean formulas (QBF) and QBF resolution systems. Particular emphasis was placed on how to theoretically model CDCL algorithms for QBF and investigate the proof-theoretic strength of different QCDCL solving approaches.

3.2 Distinct Elements in Streams: An Algorithm for the Text Book

Sourav Chakraborty (Indian Statistical Institute – Kolkata, IN)

Joint work of Sourav Chakraborty, Kuldeep Meel, N.V. Vinodchandran
Main reference Sourav Chakraborty, N. V. Vinodchandran, Kuldeep S. Meel: "Distinct Elements in Streams: An Algorithm for the (Text) Book", CoRR, Vol. abs/2301.10191, 2023.
URL https://doi.org//10.48550/arXiv.2301.10191

Given a data stream $D = a_1, a_2, ..., a_m$ of m elements where each $a_i \in [n]$, the Distinct Elements problem is to estimate the number of distinct elements that appear in the stream.

Distinct Elements has been a subject of theoretical and empirical investigations over the past four decades resulting in space optimal algorithms for it. All the current state-of-the-art algorithms are, however, beyond the reach of an undergraduate textbook owing to their reliance on the usage of notions such as pairwise independence and universal hash functions. We present a simple, intuitive, sampling-based space-efficient algorithm whose description and the proof are accessible to undergraduates with the knowledge of basic probability theory.

3.3 Testing correctness of samplers using property testing: from theory to practice and back again

Sourav Chakraborty (Indian Statistical Institute – Kolkata, IN)

Joint work of Sourav Chakraborty, Kuldeep Meel, Yash Pote, Rishiraj Bhattacharyya, Priyanka Golia, Mate Soos

How can one test the correctness of a program that is supposed to output an element from a large universe according to a certain distribution? These kind of programs are heavily used in real life but are rarely tested for correctness.

This problem can be framed as a problem in property testing. Property testing is a subject that deals with these challenges. It tries to design sub-linear algorithms for testing various properties of inputs. The key lies in the way the data is accessed by the algorithm.

One of the central problems in property testing and many other related subjects is testing if a distribution has a certain property – say whether a distribution on a finite set is uniform. The conventional way of accessing the distributions is by drawing samples according to the distributions. Unfortunately, in this setting the number of samples that are necessary for testing properties of distribution (for most natural properties) is polynomial in the size of support of the distribution. Thus when the support is relatively big the algorithms become impractical in real life applications.

We define a new way of accessing the distribution using "conditional-sampling oracle". This oracle can be used to design much faster algorithms for testing properties of distribution and thus makes the algorithm useful in practical scenarios.

We show that the conditional oracle can be implemented in many real life problems and we have been able to show the usefulness of this model and our algorithms in practical purposes and in other areas of research – like testing of probabilistic verification. This model also throws a number of interesting theoretical questions.

The talk will be based on the following works:

References

1. Eldar Fischer, Arie MAtsliah and Yonatan Goldhrish: On the Power of Conditional Samples in Distribution Testing, (SICOMP 2016)
2. Rishiraj Bhattacharyya: Property Testing of Joint Distributions using Conditional Samples, (ToCT 2018)
3. Kuldeep Meel: On Testing of Uniform Samplers, (AAAI2019)
4. Kuldeep Meel and Yash Pote: On Testing of Samplers, (NeuRIPS 2020)
5. Kuldeep Meel, Priyanka Golia and Mate Soos: Designing Samplers is Easy: The Boon of Testers, (FMCAD22)
6. Kuldeep Meel, Priyanka Golia and Mate Soos: On Quantitative Testing of Samplers, (CP22)
7. Ansuman Banerjee, Shayak Chakraborty, Sayantan Sen, Uddalok Sarkar and Kuldeep Meel: Testing of Horn Samplers, (AISTAT 2023)

3.4 Graph Colouring Is Hard on Average for Polynomial Calculus

Susanna de Rezende (Lund University, SE), Jakob Nordström (University of Copenhagen, DK & Lund University, SE)

Joint work of Jonas Conneryd, Susanna de Rezende, Shuo Pang, Jakob Nordström, Kilian Risse

We prove that polynomial calculus and hence also Nullstellensatz requires linear degree to refute that sparse random regular graphs, as well as sparse Erdös-Rényi random graphs, are 3-colourable. Using the known relation between size and degree for polynomial calculus proofs, this implies strongly exponential lower bounds on proof size.

3.5 The HITTING proof system

Yuval Filmus (Technion – Haifa, IL)

Joint work of Yuval Filmus, Edward A. Hirsch, Artur Riazanov, Alexander Smal, Marc Vinyals
Main reference Yuval Filmus, Edward A. Hirsch, Artur Riazanov, Alexander Smal, Marc Vinyals: "Proving Unsatisfiability with Hitting Formulas", CoRR, Vol. abs/2302.06241, 2023.
URL https://doi.org//10.48550/arXiv.2302.06241

A tree-like Resolution refutation of a CNF is a decision tree that solves the falsified clause search problem. We can think of the leaves of the decision tree as a partition of the space of truth assignments into "monochromatic" subcubes, in the sense that each subcube can be associated with a single refuted clause. A HITTING refutation of a CNF is any partition of the space of truth assignments into monochromatic subcubes.

We explore the relation between HITTING and other proof systems. By construction, HITTING p-simulates tree-like Resolution, and in contrast, tree-like Resolution qp-simulates HITTING, and there is a qp-separation between the two systems. Resolution can be exponentially more powerful than HITTING, but we conjecture that it does not p-simulate HITTING. Using the Raz-Shpilka PIT, we show that Extended Resolution p-simulates HITTING, though this is probably an overkill.

3.6 Top-Down Lower Bounds for Depth-Four Circuits

Mika Göös (EPFL Lausanne, CH)

Joint work of Mika Göös, Artur Riazanov, Anastasia Sofronova, Dmitry Sokolov

We present a top-down lower-bound method for depth-4 boolean circuits. In particular, we give a new proof of the well-known result that the parity function requires depth-4 circuits of size exponential in $n^{1/3}$. Our proof is an application of robust sunflowers and block unpredictability.

3.7 Capturing one-way functions via NP-hardness of meta-complexity

Shuichi Hirahara (National Institute of Informatics – Tokyo, JP)

Main reference Shuichi Hirahara: "Capturing One-Way Functions via NP-Hardness of Meta-Complexity", in Proc. of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, p. 1027–1038, Association for Computing Machinery, 2023.
URL https://doi.org//10.1145/3564246.3585130

We present the first characterization of a one-way function by worst-case hardness assumptions: A one-way function exists iff NP is hard in the worst case and "distributional Kolmogorov complexity" is NP-hard under randomized reductions. Here, the t-time bounded distributional Kolmogorov complexity of a string x given a distribution D is defined to be the length of a shortest t-time program that outputs x given as input y drawn from the distribution D with high probability. The characterization suggests that the recent approaches of using meta-complexity to exclude Heuristica and Pessiland are both sufficient and necessary.

3.8 Unprovability of strong complexity lower bounds in bounded arithmetic

Igor Carboni Oliveira (University of Warwick – Coventry, GB)

Joint work of Igor Carboni Oliveira, Jiatu Li
Main reference Jiatu Li, Igor Carboni Oliveira: "Unprovability of strong complexity lower bounds in bounded arithmetic", Electron. Colloquium Comput. Complex., Vol. TR23-022, 2023.
URL https://eccc.weizmann.ac.il/report/2023/022

While there has been progress in establishing the unprovability of complexity statements in lower fragments of bounded arithmetic, understanding the limits of Jeřábek's theory APC_1 (2007) and of higher levels of Buss's hierarchy S_2^i (1986) has been a more elusive task. Even in the more restricted setting of Cook's theory PV_1 (1975), known results often rely on a less natural formalization that encodes a complexity statement using a collection of sentences instead of a single sentence. This is done to reduce the quantifier complexity of the resulting sentences so that standard witnessing results can be invoked.

In this work, we establish unprovability results for stronger theories and for sentences of higher quantifier complexity. In particular, we unconditionally show that APC_1 cannot prove strong complexity lower bounds separating the third level of the polynomial hierarchy. In more detail, the lower bound sentence refers to the non-uniform setting ($\exists\forall\exists$ Circuits vs. $\forall\exists\forall$ Circuits) and to a mild average-case lower bound for polynomial size circuits against sub-exponential size circuits.

Our argument employs a convenient game-theoretic witnessing result that can be applied to sentences of arbitrary quantifier complexity. We combine it with extensions of a technique introduced by Krajíček (2011) that was recently employed by Pich and Santhanam (2021) to establish the unprovability of lower bounds in PV_1 and in a fragment of APC_1.

3.9 On small-depth Frege proofs for PHP

Johan Håstad (KTH Royal Institute of Technology – Stockholm, SE)

We study Frege proofs for the one-to-one graph Pigeon Hole Principle defined on the $n \times n$ grid where n is odd. We are interested in the case where each formula in the proof is a depth d formula in the basis given by $\wedge$, $\vee$, and $\neg$. We prove that in this situation the proof needs to be of size exponential in $n^{\Omega(1/d)}$. If we restrict the size of each line in the proof to be of size M then the number of lines needed is exponential in $n/(\log M)^{O(d)}$. The main technical component of the proofs is to design a new family of random restrictions and to prove the appropriate switching lemmas.

3.10 The Elliptic Curve Fast Fourier Transform (ECFFT)

Swastik Kopparty (University of Toronto, CA)

Joint work of Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, David Levit
Main reference Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, David Levit: "Elliptic Curve Fast Fourier Transform (ECFFT) Part I: Low-degree Extension in Time $O(n \log n)$ over all Finite Fields", in Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pp. 700–737, SIAM, 2023.
URL https://doi.org//10.1137/1.9781611977554.ch30
Main reference Eli Ben-Sasson, Dan Carmon, Swastik Kopparty, David Levit: "Scalable and Transparent Proofs over All Large Fields, via Elliptic Curves – (ECFFT Part II)", in Proc. of the Theory of Cryptography – 20th International Conference, TCC 2022, Chicago, IL, USA, November 7-10, 2022, Proceedings, Part I, Lecture Notes in Computer Science, Vol. 13747, pp. 467–496, Springer, 2022.
URL https://doi.org//10.1007/978-3-031-22318-1_17

This is based on the papers ECFFT I (Fast algorithms for polynomials over all fields) and ECFFT II (Scalable and Transparent proofs over all large fields), both joint work with Eli Ben-Sasson, Dan Carmon, and David Levit

I will talk about a variant (the ECFFT) of the FFT which is based on elliptic-curve groups in place of multiplicative groups. While the classical FFT over finite fields is directly applicable only when the size of the multiplicative group of the field is special, the ECFFT turns out to be directly applicable over all finite fields (because all finite fields have *some* elliptic curve group whose size is special).

We then use the ECFFT in place of the FFT for applications in fast polynomial algorithms and interactive property testing.

3.11 Locally consistent decomposition of strings with applications to edit distance sketching

Michal Koucký (Charles University – Prague, CZ)

Joint work of Sudatta Bhattacharya, Michal Koucký
Main reference Sudatta Bhattacharya, Michal Koucký: "Streaming k-edit approximate pattern matching via string decomposition", CoRR, Vol. abs/2305.00615, 2023.
URL https://doi.org//10.48550/arXiv.2305.00615

We present a new locally consistent decomposition of strings. Each string x is decomposed into blocks that can be described by grammars of size $\widetilde{O}(k)$ (using some amount of randomness). If we take two strings x and y of edit distance at most k then their block decomposition uses the same number of grammars and the i-th grammar of x is the same as the i-th grammar of y except for at most k indexes i. The edit distance of x and y equals to the sum of edit distances of pairs of blocks where x and y differ. Our decomposition can be used to design a sketch of size $\widetilde{O}(k^2)$ for edit distance, and also a rolling sketch for edit distance of size $\widetilde{O}(k^2)$. The rolling sketch allows to update the sketched string by appending a symbol or removing a symbol from the beginning of the string.

3.12 Polynomial formulations as a barrier for reduction-based hardness proofs

Alexander S. Kulikov (JetBrains Research – Paphos, CY)

Joint work of Tatiana Belova, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Denil Sharipov
Main reference Tatiana Belova, Alexander Golovnev, Alexander S. Kulikov, Ivan Mihajlin, Denil Sharipov: "Polynomial formulations as a barrier for reduction-based hardness proofs", in Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pp. 3245–3281, SIAM, 2023.
URL https://doi.org//10.1137/1.9781611977554.ch124

The Strong Exponential Time Hypothesis (SETH) asserts that for every $\varepsilon > 0$ there exists k such that k-SAT requires time $(2-\varepsilon)^n$. The field of fine-grained complexity has leveraged SETH to prove quite tight conditional lower bounds for dozens of problems in various domains and complexity classes, including Edit Distance, Graph Diameter, Hitting Set, Independent Set, and Orthogonal Vectors. Yet, it has been repeatedly asked in the literature whether SETH-hardness results can be proven for other fundamental problems such as Hamiltonian Path, Independent Set, Chromatic Number, MAX-k-SAT, and Set Cover.

In this paper, we show that fine-grained reductions implying even λ^n-hardness of these problems from SETH for *any* $\lambda > 1$, would imply new circuit lower bounds: super-linear lower bounds for Boolean series-parallel circuits or polynomial lower bounds for arithmetic circuits (each of which is a four-decade open question).

We also extend this barrier result to the class of parameterized problems. Namely, for every $\lambda > 1$, we conditionally rule out fine-grained reductions implying SETH-based lower bounds of λ^k for a number of problems parameterized by the solution size k.

Our main technical tool is a new concept called polynomial formulations. In particular, we show that many problems can be represented by relatively succinct low-degree polynomials, and that any problem with such a representation cannot be proven SETH-hard (without proving new circuit lower bounds).

3.13 Colourful TFNP and Propositional Proofs

Robert Robere (McGill University – Montréal, CA)

Joint work of Ben Davis (McGill University), Robert Robere (McGill University)
Main reference Ben Davis and Robert Robere. "Colourful TFNP and Propositional Proofs". To appear at CCC 2023.
URL https://eccc.weizmann.ac.il/report/2023/068/

Recent work in proof complexity has shown that studying many of the major proof systems studied in practice is, in a sense, completely equivalent to studying black-box versions of syntactically-defined subclasses of TFNP. Many weak proof systems, such as Resolution, Sherali-Adams, and Nullstellensatz are now known to admit characterizations of this type, and these new characterizations have been used to obtain new results in both proof complexity and the study of TFNP.

In this talk, we outline recent work in which we have characterized stronger proof systems – including $Res(k)$ and higher-depth analogues of Sherali-Adams – inside of TFNP by using the so-called "coloured" generalization of standard TFNP classes. This talk is based on joint work with Ben Davis.

3.14 Simple, deterministic, and fast (but weak) approximation for Edit Distance and Dyck Edit Distance

Michael E. Saks (Rutgers University – Piscataway, US)

Joint work of Michal Koucký, Michael E. Saks
Main reference Michal Koucký, Michael E. Saks: "Simple, deterministic, fast (but weak) approximations to edit distance and Dyck edit distance", in Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pp. 5203–5219, SIAM, 2023.
URL https://doi.org//10.1137/1.9781611977554.ch188

The edit distance between two strings, equal to the minimum number of operations (insertions, deletions or substitutions) needed to transform one to the other, is a standard measure of similarity of strings. The classic dynamic programming algorithm for edit distance requires time quadratic in n to compute the edit distance between two strings of length n, and there is evidence (via the strong exponential time hypothesis) that it may be impossible to improve substantially on this time complexity.

Recently, there has been considerable progress in developing approximation algorithms for edit distance (and the more general problem of Dyck edit distance) that are fast and have constant, or near constant approximation factors. These algorithms run in near linear time, but are logically complex, and the constants in both the running time and the approximation factor are huge, making the algorithms impractical.

In this work, we seek algorithms with weaker but still useful approximation guarantees that are practical: simple, fast and space efficient. We introduce a class of algorithms called single pass algorithms. In such an algorithm we maintain a single pointer within each string, starting at the left. In each step, if the current symbols match we advance both pointers, otherwise we have a mismatch and choose one of the pointers to advance. Such an algorithm is specified by its advancement rule, which determines which pointer to advance. We consider particularly simple (possibly randomized) advancement rules where at each mismatch step the pointer advanced depends only on the number of mismatches seen so far and the randomness of the algorithm. It is easy to see that the total number of mismatches is always an upper bound on edit distance. Saha (2014) showed that the simple randomized rule (on mismatch advance a pointer at random) when run on two strings of edit distance d returns (with high probability) an upper bound of $O(d^2)$.

In this work we (1) present a deterministic single pass algorithm that achieves similar performance and (2) prove that no algorithm (even randomized) in this class can give a better approximation factor.

For the Dyck edit distance problem, Saha gave a complicated randomized reduction from Dyck edit distance to standard edit distance at a cost of a $O(\log d)$ factor where d is the Dyck edit distance. I will present a simple deterministic reduction with a similar (slightly better) approximation guarantee.

3.15 HDX Condensers

Amnon Ta-Shma (Tel Aviv University, IL)

Joint work of Itay Cohen, Ron Roth, Amnon Ta-Shma

More than twenty years ago, Capalbo, Reingold, Vadhan, and Wigderson gave the first (and up-to-date only) explicit construction of a bipartite expander with almost full combinatorial expansion. The construction incorporates zig-zag ideas and extractor technology and is rather complicated. We give an alternative construction that builds upon recent constructions of hyper-regular, high-dimensional expanders. The new construction is, in our opinion, simple and elegant.

Beyond demonstrating a new, surprising, and intriguing, application of high-dimensional expanders, the construction employs totally new ideas which we hope may lead to progress on the still remaining open problems in the area.

3.16 Cutting Planes Width and the Complexity of Graph Isomorphism Refutations

Jacobo Torán (Universität Ulm, DE)

Joint work of Jacobo Torán, Florian Wörz

The width complexity measure plays a central role in Resolution and other propositional proof systems like Polynomial Calculus (under the name of degree). The study of width lower bounds is the most extended method for proving size lower bounds, and it is known that for these systems, proofs with small width also imply the existence of proofs with small size. Not much has been studied, however, about the width parameter in the Cutting Planes (CP) proof system, a measure that was introduced by Dantchev and Martin in 2011 under the name of CP cutwidth.

In this talk, we consider the width complexity of CP refutations of graph isomorphism formulas. For a pair of non-isomorphic graphs G and H, we show a direct connection between the Weisfeiler–Leman differentiation number $\mathsf{WL}(G, H)$ of the graphs and the width of a CP refutation for the corresponding isomorphism formula $Iso(G, H)$. In particular, we show that if $\mathsf{WL}(G, H) \leq k$, then there is a CP refutation of $Iso(G, H)$ with width k, and if $\mathsf{WL}(G, H) > k$, then there are no CP refutations of $Iso(G, H)$ with width $k - 2$. Similar results are known for other proof systems, like Resolution, Sherali–Adams, or Polynomial Calculus. We also show polynomial-size CP refutations from our width bound for isomorphism formulas for graphs with constant WL.

3.17 Extractors for Algebraic Sources

Ben Lee Volk (Reichman University – Herzliya, IL)

Joint work of Zeyu Guo, Ben Lee Volk, Akhil Jalan, David Zuckerman
Main reference Zeyu Guo, Ben Lee Volk, Akhil Jalan, David Zuckerman: "Extractors for Images of Varieties", CoRR, Vol. abs/2211.14497, 2022.
URL https://doi.org//10.48550/arXiv.2211.14497

Randomness extractors are tools for converting "low quality" randomness into "high quality" randomness. In addition to being useful in the areas of pseudorandomness and derandomization, these objects are also connected to various fundamental notions in complexity theory and mathematics in general. In this talk we'll consider the randomness extraction problem from distributions with algebraic structure. We'll survey the different types of algebraic sources and constructions, and talk about a recent construction of extractors for polynomial images of varieties

4 Working groups

4.1 Lifting dichotomy theorems

Amit Chakrabarti (Dartmouth College – Hanover, US), Susanna de Rezende (Lund University, SE), Yuval Filmus (Technion – Haifa, IL), Mika Göös (EPFL Lausanne, CH), Johan Hastad (KTH Royal Institute of Technology – Stockholm, SE), Robert Robere (McGill University – Montréal, CA), and Avishay Tal (University of California – Berkeley, US)

Whenever there is a lifting theorem that works with constant size gadgets, there is hope to understand *all* gadgets. As a simple example, consider lifting decision tree depth to decision tree size. Using a simulation-type argument or a random restriction, it is not hard to check that $\log_2 \mathrm{DTsize}(f \circ \oplus_2) \geq \mathrm{DTdepth}(f)$. In fact, this works for any gadget g as long as g does not have certificates of size 1. If g does have a certificate of size 1 then up to negating inputs and outputs, g is either a (possibly degenerate) OR, or it projects to $g_0 = x \vee (y \wedge z)$. In the former case, depth does not lift to size (take f to be a large OR). In the latter case, we can lower bound $\log_2 \mathrm{DTsize}(f \circ g)$ by both the certificate complexity of f and (using a result of Sherstov) the degree of f; in particular, $\log_2 \mathrm{DTsize}(f \circ g) = \Omega(\mathrm{DTdepth}(f)^{1/2})$. We do not know whether the square root loss is necessary.

5 Open problems

5.1 Sampling modular distributions locally

Yuval Filmus (Technion – Haifa, IL)

Joint work of Yuval Filmus, Itai Leigh, Artur Riazanov, Dmitry Sokolov

Emanuele Viola initiated the study of the complexity of distributions. Given an infinite supply of iid unbiased random bits, which distributions can we sample in low complexity? Let us focus on locally samplable distributions. These are distributions such that for each $\epsilon > 0$ there is $d = d(\epsilon)$ and a d-local sampler (meaning that every output bit depends on at most d input bits) whose output is within variation distance ϵ of the target distribution.

The uniform distribution of all even parity strings is famously samplable with no error and locality 2. What about the uniform distribution over all strings whose Hamming weight is a multiple of m? We conjecture that for $m > 2$, this distribution cannot be sampled locally.

Participants

- Shyan Akmal
MIT – Cambridge, US
- Max Bannach
Universität zu Lübeck, DE
- Olaf Beyersdorff
Friedrich-Schiller-Universität Jena, DE
- Harry Buhrman
CWI – Amsterdam, NL
- Igor Carboni Oliveira
University of Warwick – Coventry, GB
- Gaia Carenini
ENS – Paris, FR
- Amit Chakrabarti
Dartmouth College – Hanover, US
- Sourav Chakraborty
Indian Statistical Institute – Kolkata, IN
- Gil Cohen
Tel Aviv University, IL
- Susanna de Rezende
Lund University, SE
- Yuval Filmus
Technion – Haifa, IL
- Anna Gál
University of Texas – Austin, US
- Mika Göös
EPFL Lausanne, CH
- Rohit Gurjar
Indian Institute of Technology – Mumbai, IN
- Kristoffer Arnsfelt Hansen
Aarhus University, DK
- Johan Hastad
KTH Royal Institute of Technology – Stockholm, SE
- Shuichi Hirahara
National Institute of Informatics – Tokyo, JP
- Rahul Ilango
MIT – Cambridge, US
- Swastik Kopparty
University of Toronto, CA
- Michal Koucký
Charles University – Prague, CZ
- Marvin Künnemann
RPTU – Kaiserslautern, DE
- Alexander S. Kulikov
JetBrains Research – Paphos, CY
- Sophie Laplante
Université Paris Cité, FR
- Zhenjian Lu
University of Oxford, GB
- Meena Mahajan
The Institute of Mathematical Sciences – Chennai, IN
- Jakob Nordström
University of Copenhagen, DK & Lund University, SE
- Manaswi Parashar
Aarhus University, DK
- Rüdiger Reischuk
Universität zu Lübeck, DE
- Robert Robere
McGill University – Montréal, CA
- Michael E. Saks
Rutgers University – Piscataway, US
- Rahul Santhanam
University of Oxford, GB
- Melanie Schmidt
Heinrich-Heine-Universität Düsseldorf, DE
- Amnon Ta-Shma
Tel Aviv University, IL
- Avishay Tal
University of California – Berkeley, US
- Till Tantau
Universität zu Lübeck, DE
- Thomas Thierauf
Hochschule Aalen, DE
- Jacobo Torán
Universität Ulm, DE
- Quinten Tupker
CWI – Amsterdam, NL
- Ben Lee Volk
Reichman University – Herzliya, IL

Report from Dagstuhl Seminar 23112

Unifying Formal Methods for Trustworthy Distributed Systems

Swen Jacobs*[1], Kenneth McMillan*[2], Roopsha Samanta*[3], and Ilya Sergey*[4]

1 **CISPA – Saarbrücken, DE.** jacobs@cispa.de
2 **University of Texas – Austin, US.** kenmcm@cs.utexas.edu
3 **Purdue University – West Lafayette, US.** roopsha@purdue.edu
4 **National University of Singapore, SG.** ilya@nus.edu.sg

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23112 "Unifying Formal Methods for Trustworthy Distributed Systems".

Distributed systems are challenging to develop and reason about. Unsurprisingly, there have been many efforts in formally specifying, modeling, and verifying distributed systems. A bird's eye view of this vast body of work reveals two primary sensibilities. The first is that of semi-automated or interactive deductive verification targeting structured programs and implementations, and focusing on simplifying the user's task of providing inductive invariants. The second is that of fully-automated model checking, targeting more abstract models of distributed systems, and focusing on extending the boundaries of decidability for the parameterized model checking problem. Regrettably, solution frameworks and results in deductive verification and parameterized model checking have largely evolved in isolation while targeting the same overall goal.

This seminar aimed at enabling conversations and solutions cutting across the deductive verification and model checking communities, leveraging the complementary strengths of these approaches. In particular, we explored layered and compositional approaches for modeling and verification of industrial-scale distributed systems that lend themselves well to separation of verification tasks, and thereby the use of diverse proof methodologies.

Seminar March 12–15, 2023 – https://www.dagstuhl.de/23112

2012 ACM Subject Classification Computing methodologies → Distributed algorithms; Theory of computation → Concurrency; Theory of computation → Program verification

Keywords and phrases Deductive Verification, Distributed Algorithms, Formal Verification, Model Checking

Digital Object Identifier 10.4230/DagRep.13.3.32

1 Executive Summary

Swen Jacobs (CISPA – Saarbrücken, DE)
Kenneth McMillan (University of Texas – Austin, US)
Roopsha Samanta (Purdue University – West Lafayette, US)
Ilya Sergey (National University of Singapore, SG)

Dagstuhl Seminar 23112 Unifying Formal Methods for Trustworthy Distributed Systems took place on March 12–15, 2023 and had 25 participants: 9 female and 16 male, 22 from academia and 4 from industry, representing 9 different countries.

* Editor / Organizer

Unifying Formal Methods for Trustworthy Distributed Systems, *Dagstuhl Reports*, Vol. 13, Issue 3, pp. 32–48
Editors: Swen Jacobs, Kenneth McMillan, Roopsha Samanta, and Ilya Sergey
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This was a short seminar spanning 2.5 days and included four one-hour keynotes, 16 regular and short (lightning) talks, as well as two two-hour whole-seminar plenary discussions. The keynote talks were given by

1. Peter Müller (ETH Zurich) on Verified Secure Routing
2. Ken McMillan (UT Austin) on Techniques for Decidable Verification
3. Swen Jacobs (CISPA) on Parameterized Model Checking and Synthesis
4. Murdoch Jamie Gabbay (Heriot-Watt University) on Semitopologies for Heterogeneous Consensus.

The abstracts of all talks appear in this seminar report, except for one of the keynotes and two impromptu talks for which we only give the titles here:

- "Taming Unbounded Distributed Systems with Modular, Bounded Verification" by Roopsha Samanta (Purdue University), and
- "Pushing Formal Methods Tools to Industry" by Mike Dodds (Galois).

The two plenary discussions that have taken place during the seminar were focusing on the topics of (1) performing comparative studies amongst different approaches for validating distributed systems and (2) grand challenges that call for joint efforts across different approaches and schools of thought in this area.

The outcome of the first discussion was an informal proposal on a "Distributed System Verification Competition" – a community effort in the spirit of the famous "VerifyThis" competition in software verification, which would offer, on a regular basis, a selection of micro-benchmarks and semi-artificial challenges in verification, validation, and bug-finding in distributed system, focusing on different aspects of safety, liveness and providing a landscape to showcase the recent advances in interactive or automated verification.

The second panel has concluded with several ideas of a large-scale verification/validation effort in distributed systems. The most viable option was suggested based on the topic of the first keynote talk on Verified Secure Routing, which is currently only partially achieved by a combination of two specific technologies and leaves a lot of room to improvement, both in terms of specification of the properties of interest (e.g., liveness) as well as for exploring possibilities for automating proofs as well as complementing sound verification methods with testing and dynamic analyses.

Given the short nature of this seminar, the social component of its program was limited to a dinner in local restaurant "Zum Schloßberg", during which possibilities for collaboration have been discussed between the participants. As one outcome of this social interaction, possible internship opportunities in system verification were offered by one of the industry participants, with one of the junior participants currently considering taking them for the Summer 2024.

The seminar has generated several ideas for follow-up meetings. In particular, the following areas will likely benefit from more focused discussions and exchanges: (a) testing and dynamic validation of distributed systems; (b) addressing the challenge of so-called "latent proof" (ignored abstraction gap) in automated verification, and (c) programming-language based techniques for implementing large-scale systems with a support for formal reasoning and verification.

2 Table of Contents

Executive Summary

Overview of Talks

Panel discussions

3 Overview of Talks

3.1 Session types, time, timeout

Laura Bocchi (University of Kent – Canterbury, GB)

In this talk I give an introduction on binary session types, outline their relation with other formalisms, in particular communicating finite state machines, and with verification problems. I then discuss the links between session types and programming languages, and some of their usage scenarios that include static typing, run-time monitoring, and API generation. Finally, I present the extension of session types with time constraints and timeouts, discussing recent and ongoing work, as well as open problems.

3.2 Commutativity Quotients of Concurrent or Distributed Algorithms

Constantin Enea (Ecole Polytechnique – Palaiseau, FR)

Joint work of Constantin Enea, Parisa Fathololumi, Eric Koskinen
Main reference Constantin Enea, Parisa Fathololumi, Eric Koskinen: "The Commutativity Quotients of Concurrent Objects", CoRR, Vol. abs/2301.05740, 2023.
URL https://doi.org//10.48550/arXiv.2301.05740

Concurrent or distributed algorithms form the foundation of many modern automated services. Reasoning about the fine-grained complexities (interleavings, invariants, etc.) of these algorithms, however, is notoriously difficult. Formal proof methodologies for arguing about their correctness are still somewhat disconnected from the intuitive correctness arguments. Intuitions are often about a few canonical executions, possibly with few threads, whereas formal proofs would often use generic but complex arguments about arbitrary interleavings over unboundedly many threads. As a way to bring formal proofs closer to intuitive arguments, we introduce a new methodology for characterizing the interleavings of concurrent or distributed algorithms, based on their commutativity quotient. This quotient represents every interleaving up to reordering of commutative steps and, when chosen carefully, admits simple abstractions in the form of regular or context-free languages that enable simple proofs of correctness.

3.3 Checking Qualitative Liveness Properties of Replicated Systems with Stochastic Scheduling

Javier Esparza (TU München, DE)

Joint work of Michael Blondin, Javier Esparza, Martin Helfrich, Antonín Kucera, Philipp J. Meyer
Main reference Michael Blondin, Javier Esparza, Martin Helfrich, Antonín Kucera, Philipp J. Meyer: "Checking Qualitative Liveness Properties of Replicated Systems with Stochastic Scheduling", in Proc. of the Computer Aided Verification – 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 12225, pp. 372–397, Springer, 2020.
URL https://doi.org//10.1007/978-3-030-53291-8_20

We present a sound and complete method for the verification of qualitative liveness properties of replicated systems under stochastic scheduling. These are systems consisting of a finite-state program, executed by an unknown number of indistinguishable agents, where the next agent to make a move is determined by the result of a random experiment. We show that if a property of such a system holds, then there is always a witness in the shape of a Presburger stage graph: a finite graph whose nodes are Presburger-definable sets of configurations. Due to the high complexity of the verification problem (Ackermann-complete), we introduce an incomplete procedure for the construction of Presburger stage graphs, and implement it on top of an SMT solver. The procedure makes extensive use of the theory of well-quasi-orders, and of the structural theory of Petri nets and vector addition systems. We apply our results to a set of benchmarks, in particular to a large collection of population protocols, a model of distributed computation extensively studied by the distributed computing community.

3.4 The semitopology of permissionless consensus

Murdoch Jamie Gabbay (Heriot-Watt University – Edinburgh, GB) and Giuliano Losa (Stellar Development Foundation – San Francisco, US)

Main reference Murdoch Gabbay, Giuliano Losa: "Semitopology: a new topological model of heterogeneous consensus", CoRR, Vol. abs/2303.09287, 2023.
URL https://doi.org//10.48550/arXiv.2303.09287

A distributed system is *permissionless* when participants can join and leave the network without permission from a central authority. Many modern distributed systems are naturally permissionless, in the sense that a central permissioning authority would defeat their design purpose: this includes blockchains, filesharing protocols, some voting systems, and more. Due to their permissionless nature, such systems are also heterogeneous: participants may only have a partial view of the system, and they may also have different goals and beliefs. The traditional notion of consensus, i.e. system-wide agreement, may therefore not be adequate.

This is a mathematical challenge; how should we understand what permissionless consensus means? And how can we use this understanding to build mathematical models to help us engineer simple, robust, effective, and secure practical systems?

We study a new definition of permissionless consensus, based on *semitopology* – like topology, but without the restriction that intersections of opens be open. Semitopologies have a rich theory which is related to topology, but with a distinct character and mathematics. We introduce novel well-behavedness conditions, including an anti-Hausdorff property and a new notion of '*topen set*, and we show how these structures relate to consensus. We give a restriction of semitopologies to *witness semitopologies*, which are an algorithmically tractable subclass corresponding to Horn clause theories, having particularly good mathematical properties.

3.5 Parameterized Model Checking (and Synthesis)

Swen Jacobs (CISPA – Saarbrücken, DE)

Joint work of Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, Josef Widder, Nouraldin Jaber, Christopher Wagner, Milind Kulkarni, Roopsha Samanta

In this talk, I first gave a short overview of existing results in the area of parameterized model checking, including an introduction of basic techniques for obtaining decidability results, and more recent results that build on and extend these techniques [1]. In the second half of the talk, I presented our own recent work in the area. Here, I introduced the computational model of global synchronization protocols (GSPs), and described how we obtained decidability and cutoff results for its parameterized model checking problem [2, 3]. Finally, I showed how these results not only enable verification, but also synthesis, which furthermore can relieve the designer of the system from fitting the system into the decidable fragment, instead letting the synthesis algorithm take care of that [4].

References

1 Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, Josef Widder. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory, Morgan & Claypool Publishers 2015

2 Nouraldin Jaber, Swen Jacobs, Christopher Wagner, Milind Kulkarni, Roopsha Samanta. *Parameterized Verification of Systems with Global Synchronization and Guards*. CAV (1) 2020: 299-323

3 Nouraldin Jaber, Christopher Wagner, Swen Jacobs, Milind Kulkarni, Roopsha Samanta. *QuickSilver: modeling and parameterized verification for distributed agreement-based systems*. Proc. ACM Program. Lang. 5(OOPSLA): 1-31 (2021)

4 Nouraldin Jaber, Christopher Wagner, Swen Jacobs, Milind Kulkarni, Roopsha Samanta. *Synthesis of Distributed Agreement-Based Systems with Efficiently-Decidable Verification*. TACAS (2) 2023: 289-308

3.6 Verifying Indistinguishability of Privacy-Preserving Protocols

Gowtham Kaki (University of Colorado – Boulder, US)

Joint work of Kirby Linvill, Gowtham Kaki, Eric Wustrow
Main reference Kirby Linvill, Gowtham Kaki, Eric Wustrow: "Verifying Indistinguishability of Privacy-Preserving Protocols". Under submission.

Internet users rely on the protocols they use to protect their private information including their identity and the websites they visit. Formal verification of these protocols can detect subtle bugs that compromise these protections at design time, but is a challenging task as it involves probabilistic reasoning about random sampling, cryptographic primitives, and concurrent execution. Existing approaches either reason about symbolic models of the protocols that sacrifice precision for automation, or reason about more precise models that are harder to automate and require cryptographic expertise. In this talk I describe a novel approach to verifying privacy-preserving protocols that is more precise than symbolic models yet more accessible than computational models. Our approach permits direct-style proofs of privacy, as opposed to indirect game-based proofs in computational models, by formalizing privacy as indistinguishability of possible network traces induced by a protocol. We ease automation by leveraging insights from the distributed systems verification community to create sound synchronous models of concurrent protocols. Our verification framework is implemented in F* as a library we call Waldo. I talk about two large case studies of using Waldo to verify indistinguishability; one on the Encrypted Client Hello (ECH) extension of the TLS protocol and another on a Private Information Retrieval (PIR) protocol. I describe subtle flaws we uncovered in the TLS ECH specification that were missed by other efforts.

3.7 Improving usability of TLA+ tools for blockchain engineers

Igor Konnov (Informal Systems – Wien, AT)

Joint work of Igor Konnov, Shon Feder, Jure Kukovec, Gabriela Moreira, Thomas Pani

In this talk, I gave a brief introduction into the Cosmos ecosystem and a summary of results on formal specification & model checking of blockchain protocols conducted at Informal Systems. We further discussed the benefits and practical challenges of applying Temporal Logic of Actions (TLA+) and the Apalache model checker in the blockchain industry. The talk concluded with an introduction of Quint, the new syntax for the logic of TLA+. We introduced a new specification development cycle, which accommodates the needs of the protocol designers, blockchain engineers, and verification engineers.

More details about Quint may be found at the project webpage: `https://github.com/informalsystems/quint/`.

3.8 Random testing of Byzantine fault tolerant algorithms

Burcu Kulahcioglu Ozkan (TU Delft, NL)

Joint work of Levin N. Winter, Florena Buse, Daan de Graaf, Klaus von Gleissenthall, Burcu Kulahcioglu Ozkan
Main reference Levin N. Winter, Florena Buse, Daan de Graaf, Klaus von Gleissenthall, Burcu Kulahcioglu Ozkan: "Randomized Testing of Byzantine Fault Tolerant Algorithms", Proc. ACM Program. Lang., Vol. 7(OOPSLA1), pp. 757–788, 2023.
URL https://doi.org//10.1145/3586053

Byzantine fault-tolerant algorithms promise agreement on a correct value, even if a subset of processes can deviate from the algorithm arbitrarily. While these algorithms provide strong guarantees in theory, protocol bugs and implementation mistakes may cause them to violate fault tolerance in practice.

This talk discusses the challenges of testing Byzantine fault-tolerant systems and introduces ByzzFuzz, a method for automatically finding errors in implementations of Byzantine fault-tolerant algorithms through randomized testing. ByzzFuzz detects fault-tolerance bugs by injecting randomly generated network and process faults into their executions. To navigate the space of possible process faults, ByzzFuzz introduces small-scope message mutations which mutate the contents of the protocol messages by applying small changes to the original message either in value (e.g., by incrementing the round number) or in time (e.g., by repeating a proposal value from a previous message). The evaluation of ByzzFuzz on the implementations of popular blockchains show that small-scope mutations, combined with insights from the testing and fuzzing literature, are effective at uncovering protocol logic and implementation bugs in real-world fault-tolerant systems.

3.9 Verified Causal Broadcast with Liquid Haskell

Lindsey Kuper (University of California – Santa Cruz, US)

Joint work of Patrick Redmond, Gan Shen, Niki Vazou, Lindsey Kuper

Protocols to ensure that messages are delivered in causal order are a ubiquitous building block of distributed systems. For instance, distributed data storage systems can use causally ordered message delivery to ensure causal consistency, and CRDTs can rely on the existence of an underlying causally-ordered messaging layer to simplify their implementation. A causal delivery protocol ensures that when a message is delivered to a process, any causally preceding messages sent to the same process have already been delivered to it. While causal delivery protocols are widely used, verification of their correctness is less common, much less machine-checked proofs about executable implementations.

We implemented a standard causal broadcast protocol in Haskell and used the Liquid Haskell solver-aided verification system to express and mechanically prove that messages will never be delivered to a process in an order that violates causality. We express this property using refinement types and prove that it holds of our implementation, taking advantage of Liquid Haskell's underlying SMT solver to automate parts of the proof and using its manual theorem-proving features for the rest. We then put our verified causal broadcast implementation to work as the foundation of a distributed key-value store.

3.10 Potential-based semantics for causally consistent shared memory

Ori Lahav (Tel Aviv University, IL)

Joint work of Ori Lahav, Udi Boker
Main reference Ori Lahav, Udi Boker: "What's Decidable About Causally Consistent Shared Memory?", ACM Trans. Program. Lang. Syst., Vol. 44(2), pp. 8:1–8:55, 2022.
URL https://doi.org//10.1145/3505273

While causal consistency is one of the most fundamental consistency models weaker than sequential consistency, the decidability of safety verification for (finite-state) concurrent programs running under causally consistent shared memories is still unclear. In this paper, we establish the decidability of this problem for two standard and well-studied variants of causal consistency. To do so, for each variant, we develop an equivalent "lossy" operational semantics, whose states track possible futures, rather than more standard semantics that record the history of the execution. We show that these semantics constitute well-structured transition systems, thus enabling decidable verification. Based on a key observation, which we call the "shared-memory causality principle", the two novel semantics may also be of independent use in the investigation of weakly consistent models and their verification. Interestingly, our results are in contrast to the undecidability of this problem under the Release/Acquire fragment of the C/C++11 memory model, which forms another variant of causally consistent memory that, in terms of allowed outcomes, lies strictly between the two models studied here. Nevertheless, we show that all these three variants coincide for write/write-race-free programs, which implies the decidability of verification for such programs under Release/Acquire.

References

1 Ori Lahav. *Verification under causally consistent shared memory.* ACM SIGLOG News 6:2, April 2019, pages 43–56. `https://doi.org/10.1145/3326938.3326942`

2 Ori Lahav and Udi Boker. *Decidable verification under a causally consistent shared memory.* In *PLDI*, ACM, 2020. `https://doi.org/10.1145/3385412.3385966`

3 Lahav, O., Boker, U.: What's Decidable About Causally Consistent Shared Memory? ACM Trans. Program. Lang. Syst. **44**(2), 8:1–8:55 (2022), `https://doi.org/10.1145/3505273`

3.11 Parameterized Verification of Randomized Consensus Algorithms

Marijana Lazic (TU München, DE)

Joint work of Nathalie Bertrand, Igor Konnov, Josef Widder

In this talk I showed the extension of threshold automata for modeling randomized consensus algorithms that perform an unbounded number of asynchronous rounds. Moreover, I presented techniques for parameterized verification of the three randomized consensus properties: agreement, validity and almost sure termination.

For non-probabilistic properties, I showed that it is necessary and sufficient to verify these properties under round-rigid schedules, that is, schedules where processes enter round r only after all processes finished round $r - 1$.

For almost-sure termination, I proceed in 2 steps. First, I analyze these algorithms under round-rigid adversaries, that is, fair adversaries that only generate round-rigid schedules. This allows us to do compositional and inductive reasoning that reduces verification of the asynchronous multi-round algorithms to model checking of a one-round threshold automaton.

We apply this framework and automatically verify the following classic algorithms: Ben-Or's and Bracha's seminal consensus algorithms for crashes and Byzantine faults, 2-set agreement for crash faults, and RS-Bosco for the Byzantine case.

Second, I focus on weak adversaries, that express the property that the adversary (scheduler), which has to decide which messages to deliver to which process, has no means of inferring the outcome of random choices, and the content of the messages. I introduced a model for randomized distributed algorithms that allows us to formalize the notion of weak adversaries. I show that for verification purposes, the class of weak adversaries can be restricted to round-rigid adversaries. This new reduction theorem paves the way to the parameterized verification of randomized distributed algorithms under the more realistic weak adversaries.

References

1 Nathalie Bertrand, Marijana Lazic, Josef Widder. *A Reduction Theorem for Randomized Distributed Algorithms Under Weak Adversaries.* VMCAI 2021: 219-239

2 Nathalie Bertrand, Igor Konnov, Marijana Lazic, Josef Widder. *Verification of Randomized Consensus Algorithms Under Round-Rigid Adversaries.* CONCUR 2019: 33:1-33:15

3.12 A simple proof of the FLP impossibility result

Giuliano Losa (Stellar Development Foundation – San Francisco, US)

Joint work of Giuliano Losa, Eli Gafni
Main reference Eli Gafni, Giuliano Losa: "Time is not a Healer, but it Sure Makes Hindsight 20:20", CoRR, Vol. abs/2305.02295, 2023.
URL https://doi.org//10.48550/arXiv.2305.02295

We present a remarkably simple proof of the famous FLP impossibility result. We first observe that solving consensus in an asynchronous system where one process may fail implies solving consensus in the synchronous model of Santoro and Widmayer. Then, we build on insights from Volzer to obtain an almost trivial impossibility proof in the synchronous model.

3.13 Random Testing of Distributed Systems

Rupak Majumdar (MPI-SWS – Kaiserslautern, DE)

This talk was an overview of the main ideas behind the state-of-the-art techniques for effective and efficient fuzz-testing of realistic distributed systems. I have outlined the basic theory facts that explain why well-adopted "black-box" testing tools, such as Jepsen, are surprisingly effective in discovering interesting bugs in distributed systems. I have also described approaches that can be used to improve the algorithms that navigated through the very large space of possible distributed interactions by exploiting the ideas of partial synchrony and round-based formulation of distributed protocols. These algorithms define a sample space based on the underlying partial orderings of events in the distributed system and sample efficiently from that space.

The talk described work that appeared in the following papers and dissertation:

References

1 Rupak Majumdar and Filip Niksic. Why is random testing effective for partition tolerance bugs? Proc. ACM Program. Lang. 2(POPL): 46:1-46:24 (2018)

2 Burcu Kulahcioglu Ozkan, Rupak Majumdar, Filip Niksic, Mitra Tabaei Befrouei, Georg Weissenbacher. Randomized testing of distributed systems with probabilistic guarantees. Proc. ACM Program. Lang. 2(OOPSLA): 160:1-160:28 (2018)

3 Filip Niksic. Combinatorial Constructions for Effective Testing. Kaiserslautern University of Technology, Germany, 2019

4 Cezara Dragoi, Constantin Enea, Burcu Kulahcioglu Ozkan, Rupak Majumdar, Filip Niksic: Testing consensus implementations using communication closure. Proc. ACM Program. Lang. 4(OOPSLA): 210:1-210:29 (2020)

3.14 Verified Secure Routing

Peter Müller (ETH Zürich, CH)

SCION is a new Internet architecture that addresses many of the security vulnerabilities of today's Internet. Its clean-slate design provides, among other properties, route control, failure isolation, and multi-path communication. The verifiedSCION project is an effort to formally verify the correctness and security of SCION. It aims to provide strong guarantees for the entire architecture, from the protocol design to its concrete implementation. The project uses stepwise refinement to prove that the protocol withstands increasingly strong attackers. The refinement proofs assume that all network components such as routers satisfy their specifications. This property is then verified separately using deductive program verification in separation logic. This talk will give an overview of the verifiedSCION project and explain, in particular, how we verify code-level properties such as memory safety, I/O behavior, and information flow security.

3.15 Interactive Synthesis of Distributed Protocols

Kedar Namjoshi (Nokia Bell Labs – Murray Hill, US)

Joint work of Kedar Namjoshi, Senthil Rajasekaran

It is difficult to verify distributed protocols: one must prove that all configurations satisfy a global property, which is in general an undecidable question. Could one synthesize such protocols instead? That is undecidable, too; but we suggest using a process of successive refinement on specifications, with the goal of obtaining a specification that is localized to a generic process and a generic neighborhood, which is simpler to synthesize. A protocol designer suggests the sequence of specifications, with automated help in establishing refinements.

3.16 Reasoning about Byzantine Accountability

Ilya Sergey (National University of Singapore, SG) and George Pîrlea (National University of Singapore, SG)

Modern Byzantine distributed consensus protocols can achieve agreement in the presence of a bounded number of faulty nodes trying to corrupt the network, yet they fail to identify or disincentivise Byzantine behaviour by malicious nodes. Accountable Byzantine Consensus (ABC) is a protocol transformation that, when combined with any Byzantine consensus protocol, guarantees both consensus and accountability.

I this talk, I presented the key ideas of Byzantine accountability, its semantic model, as well some preliminary results on formalising and verifying accountable consensus protocols.

3.17 Finding Infinite Counter Models in Deductive Verification

Sharon Shoham Buchbinder (Tel Aviv University, IL)

Joint work of Sharon Shoham Buchbinder, Neta Elad, Oded Padon

First-order logic, and quantifiers in particular, are widely used in deductive verification of programs and systems. Quantifiers are essential for describing systems with unbounded domains, but prove difficult for automated solvers. Significant effort has been dedicated to finding quantifier instantiations that establish unsatisfiability of quantified formulas, thus ensuring validity of a system's verification conditions. However, in many cases the formulas are satisfiable—this is often the case in intermediate steps of the verification process, e.g., when an invariant is not yet inductive. For such cases, existing tools are limited to finding finite models. Yet, some quantified formulas are satisfiable but only have infinite models, which current solvers are unable to find. Such infinite counter-models are especially typical when first-order logic is used to approximate the natural numbers, the integers, or other inductive definitions, which is common in deductive verification.

In this work, we tackle the problem of finding such infinite models, specifically, finite representations thereof that can be presented to the user of a deductive verification tool. These models give insight into the verification failure, and allow the user to identify and fix bugs in the modeling of the system and its properties. Our approach consists of three parts. First, we introduce templates as a way to represent certain infinite models, and show that formulas can be efficiently model checked against them. Second, we identify a new decidable fragment of first-order logic that extends and subsumes EPR, where satisfiable formulas always have a model representable by a template of a bounded size. Finally, we describe an effective decision procedure to symbolically explore this (usually vast) search space of templates.

We evaluate our approach on examples from a variety of domains: distributed consensus protocols, linked lists, and axiomatic arithmetic. Our implementation quickly finds infinite counter-models that demonstrate the source of verification failures in a simple way, even in cases beyond the decidable fragment, while state-of-the-art SMT solvers and theorem provers such as Z3, cvc5, and Vampire diverge or return "unknown".

3.18 Deadlock-free asynchronous message reordering in Rust with multiparty session types

Nobuko Yoshida (University of Oxford, GB)

Joint work of Zak Cutner, Nobuko Yoshida, Martin Vassor
Main reference Zak Cutner, Nobuko Yoshida, Martin Vassor: "Deadlock-free asynchronous message reordering in rust with multiparty session types", in Proc. of the PPoPP '22: 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, Seoul, Republic of Korea, April 2 – 6, 2022, pp. 246–261, ACM, 2022.
URL https://doi.org//10.1145/3503221.3508404

Rust is a modern systems language focussed on performance and reliability. Complementing Rust's promise to provide "fearless concurrency," developers frequently exploit asynchronous message passing. Unfortunately, sending and receiving messages in an arbitrary order to maximise computation-communication overlap (a popular optimisation in message-passing applications) opens up a Pandora's box of subtle concurrency bugs. To guarantee deadlock-freedom by construction, we present Rumpsteak: a new Rust framework based on multiparty session types. Previous session type implementations in Rust are either built upon synchronous and blocking communication and/or are limited to two-party interactions. Crucially, none support the arbitrary ordering of messages for efficiency. Rumpsteak instead targets asynchronous async/await code. Its unique ability is allowing developers to arbitrarily order send/receive messages whilst preserving deadlock-freedom. For this, Rumpsteak incorporates two recent advanced session type theories: (1) k-multiparty compatibility, which globally verifies the safety of a set of participants, and (2) asynchronous multiparty session subtyping, which locally verifies optimisations in the context of a single participant. Specifically, we propose a novel algorithm for asynchronous subtyping that is both sound and decidable. We first talk about Rumpsteak and show the new algorithm. We then talk about our evaluation against other Rust implementations and asynchronous verification tools. We conclude the talk with a demonstration of Rumpsteak.

3.19 A (not very simple) Protocol whose Mechanized Proof is ????

Lenore D. Zuck (University of Illinois – Chicago, US)

In JACM 41(6) Afek et al described a protocol [1], originally conceived by Wang and Zuck [2], then simplified by Afek, whose goal is to transmit an infinite sequence of messages, from a finite alphabet, over bi-directional channels that can reorder and delete messages. While the impossibility of transmitting such a sequence over channels that can also duplicate messages was well known at the time, it was conjectured that reordering and deleting channels suffice to render the problem impossible. The protocol served to refute this conjecture. The original description of the protocol was "flat," and Afek's suggestion to embed it into the "probe" mechanism transformed it into a layer protocol. The top layer implements a FIFO transmission over a lossy channel, for which the well-studied and verified Alternating Bit Protocol suffices. The bottom layer implements a lossy channel over a bi-directional channel that can loss and reorder messages. This protocol, as well as others using on the "probe" mechanism, was never mechanically verified. The talk introduces the protocols and describes the challenges in using existing tools to formally verify probe-based protocols.

References

1 Yehuda Afek, Hagit Attiya, Alan D. Fekete, Michael J. Fischer, Nancy A. Lynch, Yishay Mansour, Da-Wei Wang, Lenore D. Zuck. *Reliable Communication Over Unreliable Channels*. J. ACM 41(6): 1267-1297, 1994

2 Da-Wei Wang, Lenore D. Zuck. *Tight Bounds for the Sequence Transmission Problem*. PODC 1989: 73-83

4 Panel discussions

4.1 A Competition for Distributed Systems Verification?

Swen Jacobs (CISPA – Saarbrücken, DE), Kenneth McMillan (University of Texas – Austin, US), Roopsha Samanta (Purdue University – West Lafayette, US), and Ilya Sergey (National University of Singapore, SG)

Competitions have a long-standing tradition in the fields of verification and automated reasoning. They serve to unify, energize and provide guidance to the research community by establishing a universal format in which verification problems are stated, collecting a library of interesting and challenging benchmark problems, and providing an independent and unbiased platform for the comparison of verification tools. Competitions for different flavours of verification and automated reasoning have been very successful in achieving these goals [4, 1, 2, 3, 5], and were able to draw positive attention to their respective fields in the process.

While competitions have been successful in areas where verification tasks can be fully automated, this seems to be an overly ambitious goal for distributed verification in general: if we consider the verification of realistic distributed algorithms or systems, then the problem is arguably more difficult than in any of the areas where competitions of push-button tools exist. While a restricted scope of the competition would allow us to make it amenable to fully automatic tools, this would make it uninteresting for a large part of the community.

Thus, we concluded that for distributed systems verification an interactive competition would be more suitable, in the style of the VerifyThis competition [6], where a verification team tries to solve verification problems interactively with their tool of choice. This could either be a stand-alone solution, or be separated into multiple tracks, some of which are limited in scope and only only allow fully automated tools. Finally, a third choice would be to try to achieve the benefits of a competition without actually hosting one, i.e., trying to establish a standard format for problems and collecting a library of challenging problems that are offered to the research community, but without a dedicated comparison of tools at specific fixed times.

References

1 Gianpiero Cabodi, Carmelo Loiacono, Marco Palena, Paolo Pasini, Denis Patti, Stefano Quer, Danilo Vendraminetto, Armin Biere, Keijo Heljanko. *Hardware Model Checking Competition 2014: An Analysis and Comparison of Solvers and Benchmarks.* J. Satisf. Boolean Model. Comput. 9(1): 135-172 (2014)

2 Dirk Beyer. *Competition on Software Verification – (SV-COMP).* TACAS 2012: 504-524

3 Clark W. Barrett, Morgan Deters, Leonardo Mendonça de Moura, Albert Oliveras, Aaron Stump. *6 Years of SMT-COMP.* J. Autom. Reason. 50(3): 243-277 (2013)

4 Geoff Sutcliffe. *The CADE ATP System Competition – CASC.* AI Mag. 37(2): 99-101 (2016)

5 Swen Jacobs, Roderick Bloem, Romain Brenguier, Rüdiger Ehlers, Timotheus Hell, Robert Könighofer, Guillermo A. Pérez, Jean-François Raskin, Leonid Ryzhyk, Ocan Sankur, Martina Seidl, Leander Tentrup, Adam Walker. *The first reactive synthesis competition (SYNTCOMP 2014).* Int. J. Softw. Tools Technol. Transf. 19(3): 367-390 (2017)

6 Gidon Ernst, Marieke Huisman, Wojciech Mostowski, Mattias Ulbrich. *VerifyThis – Verification Competition with a Human Factor.* TACAS (3) 2019: 176-195

4.2 Grand Challenges for Distributed Systems Verification

Swen Jacobs (CISPA – Saarbrücken, DE), Kenneth McMillan (University of Texas – Austin, US), Roopsha Samanta (Purdue University – West Lafayette, US), and Ilya Sergey (National University of Singapore, SG)

Grand challenges in science are considered as beneficial in energizing the scientific community and focusing its efforts on meaningful goals. According to their name, they should be sufficiently challenging such that they cannot be completely solved by any research group in a single project, but require a long-term effort and collaboration between different research groups and communities. Solving them should have a major positive impact, not only on the scientific community, but also on society as a whole.

For grand challenges in the area of distributed systems verification, we discussed several ideas. However, we concluded that the effort it would take to design a project that is sufficiently large-scale and challenging, while at the same time allowing a large part of the distributed systems verification community to participate without major obstacles, would go well beyond what could be discussed in this short time frame.

Instead, we found that a particularly promising idea is to take "Verified Secure Routing" (as presented in the talk by Peter Müller) as a grand challenge. The reasons are that this includes challenging aspects and sub-problems for many different research directions, ranging from low-level protocol design over path exploration to information-flow properties. Moreover, the verification tasks are sufficiently difficult to be suitable (at different levels of abstraction) for different flavors of verification, from mechanized interactive proofs to partially or fully automated proof techniques. Finally, a lot of the groundwork in defining the problem and many of its sub-problems has already been done in the large-scale project "verified SCION" at ETH. This project concentrates on mechanized proofs with tight interaction of the protocol and system engineers, and leaves open many details, as well as aspects of automating the verification. This results in a low entry bar for even small research groups to contribute to this grand challenge.

Participants

- Laura Bocchi
University of Kent –
Canterbury, GB
- Ahmed Bouajjani
Université Paris Cité, FR
- Andreea Costea
National University of
Singapore, SG
- Mike Dodds
Galois – Portland, US
- Constantin Enea
Ecole Polytechnique –
Palaiseau, FR
- Javier Esparza
TU München, DE
- Murdoch Jamie Gabbay
Heriot-Watt University –
Edinburgh, GB
- Swen Jacobs
CISPA – Saarbrücken, DE
- Gowtham Kaki
University of Colorado –
Boulder, US
- Igor Konnov
Informal Systems – Wien, AT
- Burcu Kulahcioglu Ozkan
TU Delft, NL
- Lindsey Kuper
University of California –
Santa Cruz, US
- Ori Lahav
Tel Aviv University, IL
- Marijana Lazic
TU München, DE
- Giuliano Losa
Stellar Development Foundation –
San Francisco, US
- Rupak Majumdar
MPI-SWS – Kaiserslautern, DE
- Kenneth McMillan
University of Texas – Austin, US
- Peter Müller
ETH Zürich, CH
- Kedar Namjoshi
Nokia Bell Labs –
Murray Hill, US
- George Pîrlea
National University of
Singapore, SG
- Roopsha Samanta
Purdue University – West
Lafayette, US
- Ilya Sergey
National University of
Singapore, SG
- Sharon Shoham Buchbinder
Tel Aviv University, IL
- Nobuko Yoshida
University of Oxford, GB
- Lenore D. Zuck
University of Illinois –
Chicago, US

Report from Dagstuhl Seminar 23121

Pattern Avoidance, Statistical Mechanics and Computational Complexity

David Bevan[*1], **Miklós Bóna**[*2], **and István Miklós**[*3]

1 **University of Strathclyde – Glasgow, GB.** david.bevan@strath.ac.uk
2 **University of Florida – Gainesville, US.** bona@ufl.edu
3 **ELKH – Budapest, HU.** miklos.istvan.74@gmail.com

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23121 “Pattern Avoidance, Statistical Mechanics and Computational Complexity”.

Seminar March 19–24, 2023 – https://www.dagstuhl.de/23121

2012 ACM Subject Classification Information systems → Data structures; Theory of computation → Formal languages and automata theory; Mathematics of computing → Combinatoric problems; Theory of computation → Algebraic complexity theory; Computing methodologies → Modeling and simulation

Keywords and phrases algorithms, counting and sampling, modeling, permutation patterns, statistical physics.

Digital Object Identifier 10.4230/DagRep.13.3.49

1 Executive Summary

Miklós Bóna (University of Florida – Gainesville, US)
David Bevan (University of Strathclyde – Glasgow, GB)
István Miklós (ELKH – Budapest, HU)

The Dagstuhl Seminar took place from March 19, 2023 to March 24, 2023. It had 36 participants, who were researchers in theoretical computer science, combinatorics, and statistical mechanics. The aftermath of COVID made the planning of the seminar more challenging than usual; for instance, researchers from China, Australia and New Zealand were still extremely reluctant to travel. However, in the end we succeeded in bringing together a geographically diverse group of researchers. The participants came from 12 countries, from Canada, the Czech Republic, Denmark, Finland, France, Germany, Hungary, Iceland, Italy, Turkey, the United Kingdom, and the United States. The seminar featured 21 talks, four of which were hour-long talks, and two open problem sessions.

Several collaborative projects have been started. For example, David Bevan started a collaboration with Mathilde Bouvel on the scaling limit of permutation classes avoiding a pattern with extremal first or last point. Jessica Striker, Mathilde Bouvel and Rebecca Smith started working on the open questions in Striker’s talk on six-vertex configurations. Colin Defant, Rebecca Smith, Miklós Bóna and Justin Troyka discussed new observations on pattern avoiding permutations whose squares are also pattern avoiding. Jessica Striker and

[*] Editor / Organizer

Sergi Elizalde started working together on promotion in cylindric tableaux. Natasha Blitvić and Sergi Elizalde explored why counting permutations avoiding a pattern seems to yield moment sequences, and how to interpret that phenomenon.

The open problem sessions were especially successful. In earlier seminars, we held one open problem session. However, this time there was so much interest in presenting open problems that we decided to hold two open problem sessions. Six families of open questions were presented at each of them.

Numerous participants expressed their pleasure with the seminar and its sequence of talks. The prevailing view was that while the participants came from three different fields, they were all open to the other two fields, and therefore, they all learned about results that they would not have learned otherwise. Therefore, we have all the reasons to believe that the seminar was a success, and we would like to repeat something similar sometime in the future. We have even discussed specific plans for a possible future seminar.

2 Table of Contents

Executive Summary

Overview of Talks

Open problems

3 Overview of Talks

3.1 Saturation for permutation matrices

Benjamin Aram Berendsohn (FU Berlin, DE)

Main reference Benjamin Aram Berendsohn: "An exact characterization of saturation for permutation matrices". Combinatorial Theory, 3(1), 2023.
URL https://doi.org/10.5070/C63160430

A 0-1 matrix M contains a 0-1 matrix *pattern* P if we can obtain P from M by deleting rows and/or columns and turning arbitrary 1-entries into 0s. The saturation function $\mathrm{sat}(P,n)$ for a 0-1 matrix pattern P indicates the minimum number of 1s in an $n \times n$ 0-1 matrix that does not contain P, but where changing any 0-entry into a 1-entry creates an occurrence of P.

Saturation for 0-1 matrices was introduced by Brualdi and Cao [arXiv 2020]. Fulek and Keszegh [SIAM J. Discret. Math. 2021] started a systematic study, and showed that each pattern has a saturation function that is either bounded or linear. They found large classes of patterns with linear saturation function, but only a single pattern with bounded saturation function.

Subsequently, Geneson [Electron. J. Comb. 2021] showed that almost all *permutation matrices* have bounded saturation functions. In this talk, we outline how to complete the classification of permutation matrices using a construction based on oscillations in indecomposable permutation matrices.

3.2 Mesh patterns in random permutations

David Bevan (University of Strathclyde – Glasgow, GB)

Joint work of David Bevan, Jason Smith

We say that the *likelihood* of a mesh pattern is the asymptotic probability that a random permutation contains an occurrence of the pattern. In this talk we investigate the likelihood of a variety of patterns, determining their values for every vincular pattern. For bivincular patterns, the Small Anchors Theorem distinguishes between those patterns whose likelihood equals zero, those whose likelihood is positive but less than 1, and those whose likelihood equals 1. We also determine the (rational) likelihood of any bivincular pattern formed of what we call *anchored trees*. Other bivincular patterns, such as the small ascent and small descent, have irrational likelihoods, whose values can be established using the Chen–Stein method.

3.3 Combinatorial moment sequences and permutation patterns

Natasha Blitvic (Queen Mary University of London, GB)

Joint work of Natasha Blitvic, Mohammed Slim Kammoun, Einar Steingrimsson
Main reference Natasha Blitvic, Einar Steingrimsson: "Permutations, Moments, Measures" in Transactions of the American Mathematical Society, Vol. 374, Number 8, August 2021, pp. 5473–5509.
URL https://doi.org/10.1090/tran/8330

Take your favorite integer sequence. Is it a sequence of moments of some positive Borel measure on the real line? The necessary and sufficient condition for a real sequence $(a_n)_{n\geq 0}$ to be a moment sequence was determined by Hamburger over a hundred years ago: namely, the Hankel matrices $(a_{i+j})_{0\leq i,j\leq n}$ must all be positive semidefinite. In practice, when dealing with combinatorial sequences, positivity can be difficult to evaluate. In fact, starting with a given sequence, it can be surprisingly tricky to predict (or intuit) whether it will turn out to be a moment sequence. For example, take a_n to count the permutations on n letters avoiding the consecutive permutation pattern 123 and b_n the permutations avoiding the consecutive pattern 132. Only one of these is a moment sequence – can you predict which one?

In [1], we present a combinatorial framework, in the form of a fourteen-parameter continued fraction, that turns a hard question (that of establishing positivity) into a much easier one (that of decomposing combinatorial statistics into certain elementary building blocks). It builds on the rich tradition of work on combinatorial interpretations of continued fractions (see references in [1]). Applied to permutation patterns, the framework allows us to classify all (vincular) permutation patterns of length three according to whether they give rise to moment sequences.

While several teams of researchers in this area believe that avoiders of classical permutation patterns uniformly give moment sequences (see the Open Problem presented by this author at the same conference), consecutive permutation patterns, as per the example given above, are significantly worse behaved. Nevertheless, in [2], we identify the "correct lens" through which consecutive permutation patterns appear to enjoy analogous positivity properties. This positivity result is still conjectural but, thanks to enumerative results in [2], is strongly supported by numerical evidence.

References

1 N. Blitvić and E. Steingrímsson, *Permutations, Moments, Measures*, Transactions of the American Mathematical Society, Vol. 374, Number 8, August 2021, pp. 5473–5509.
2 N. Blitvić, S. M. Kammoun, E. Steingrímsson,*A new perspective on positivity in (consecutive) permutation patterns*, Proceedings of the FPSAC 2023, July 17-21, Davis CA.

3.4 Non-uniform permutations biased according to their records

Mathilde Bouvel (LORIA – Nancy, FR)

Joint work of Nicolas Auger, Mathilde Bouvel, Cyril Nicaud, Carine Pivoteau

In this talk, we study a non-uniform distribution on permutations (of any given size), where the probability of a permutation is proportional to θ^{rec} where rec denotes the number of records (a.k.a. left-to-right maxima). The motivation for defining this model of non-uniform random permutations is the analysis of algorithms. Indeed, when analyzing algorithms working on arrays of numbers (modeled by permutations), the uniform distribution on the set of possible inputs is usually assumed. However, the actual data on which these algorithms are used is rarely uniform, and often displays a bias towards "sortedness". Our model has this bias towards "sortedness" while remaining tractable from the point of view of the analysis of algorithms. Our results on this model are of three types. First, we exhibit several efficient random samplers of permutations under this distribution. Second, we analyze the behavior of some classical permutation statistics, some of which with applications to the analysis of algorithms. Finally, we describe the "typical shape" of permutations in our model, by means of their (deterministic) permuton limit.

3.5 On the problem of Hertzsprung and similar problems

Anders Claesson (University of Iceland – Reykjavik, IS)

Main reference Anders Claesson: "From Hertzsprung's problem to pattern-rewriting systems", Algebraic Combinatorics, 5(6):1257–1277, 2022.
URL https://doi.org/10.5802/alco.202

Drawing on a problem posed by Hertzsprung in 1887 (sometimes called the n-kings problem), we say that a permutation w contains the Hertzsprung pattern u if there is factor of w that differ only by a constant from u in the sense that there is a factor $w(d+1)w(d+2)\dots w(d+k)$ such that $w(d+1) - u(1) = \cdots = w(d+k) - u(k)$. Using a combination of the Goulden-Jackson cluster method and the transfer-matrix method we determine the joint distribution of occurrences of any set of Hertzsprung patterns, thus substantially generalizing earlier results by Jackson et al. on the distribution of ascending and descending runs in permutations. We apply our results to the problem of counting permutations up to pattern-replacement equivalences, and using pattern-rewriting systems – a new formalism similar to the much studied string-rewriting systems – we solve a couple of open problems raised by Linton et al. in 2012.

3.6 Permutations and Exclusion Processes

Sylvie Corteel (Université Paris Cité, FR)

We will review results on the combinatorics of the exclusion process which is a classical model in statistical physics. We will explain why permutations and the pattern 31-2 appear when we study the exclusion process with open boundaries. We will then propose a series of open problems on the combinatorics of more general processes.

3.7 On complex roots of the independence polynomial

Péter Csikvári (Alfréd Rényi Institute of Mathematics – Budapest, HU)

Joint work of Ferenc Bencs, Péter Csikvári, Piyush Srivastava, Jan Vondrák
Main reference Ferenc Bencs, Péter Csikvári, Piyush Srivastava, Jan Vondrák: "On complex roots of the independence polynomial", in Proc. of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023, pp. 675–699, SIAM, 2023.
URL https://doi.org//10.1137/1.9781611977554.ch29

The independence polynomial of a graph is the generating polynomial of all its independent sets. Formally, given a graph G, its independence polynomial $Z_G(\lambda)$ is given by $\sum_I \lambda^{|I|}$, where the sum is over all independent sets I of G. The independence polynomial has been an important object of study in both combinatorics, statistical physics and computer science. In particular, the algorithmic problem of estimating $Z_G(\lambda)$ for a fixed positive λ on an input graph G is a natural generalization of the problem of counting independent sets, and its study has led to some of the most striking connections between computational complexity and the theory of phase transitions. More surprisingly, the independence polynomial for negative and complex values of λ also turns out to be related to problems in statistical physics and combinatorics. In particular, the locations of the complex roots of the independence polynomial of bounded degree graphs turn out to be very closely related to the Lovász local lemma, and also to the questions in the computational complexity of counting. In this talk we give new geometric criteria for establishing zero-free regions as well as for carrying out semi-rigorous numerical explorations. We then provide several examples of the (rigorous) use of these criteria, by establishing new zero-free regions. Joint work with Ferenc Bencs, Piyush Srivastava and Jan Vondrák.

3.8 The complexity of computing immanants

Radu Curticapean (IT University of Copenhagen, DK)

Main reference Radu Curticapean: "A full complexity dichotomy for immanant families", CoRR, Vol. abs/2102.04340, 2021.
URL https://arxiv.org/abs/2102.04340

Immanants are matrix functions that generalize determinants and permanents. Given an irreducible character X_λ of S_n for some partition λ of n, the immanant associated with λ is a sum-product over permutations π in S_n, much like the determinant, but with $X_\lambda(\pi)$ playing the role of $sgn(\pi)$.

Hartmann showed in 1985 that immanants can be evaluated in polynomial time for sign-ish characters. More precisely, for a partition λ of n with s parts, let $b(\lambda) := n - s$ count the boxes to the right of the first column in the Young diagram of λ. The immanant associated with λ can be evaluated in $n^O(b(\lambda))$ time.

Since this initial result, complementing hardness results have been obtained for several families of immanants derived from partitions with unbounded $b(\lambda)$. This includes permanents, immanants associated with hook characters, and other classes. In this talk, we complete the picture of hard immanant families: Under a standard assumption from parameterized complexity, we rule out polynomial-time algorithms for well-behaved immanant families with unbounded $b(\lambda)$. For immanant families in which $b(\lambda)$ even grows polynomially, we establish hardness for #P and VNP.

3.9 Three Topics in Pattern Avoidance

Colin Defant (MIT – Cambridge, US)

Joint work of Colin Defant, Amanda Burcroff, Noah Kravitz, Ashwin Sah

This talk will survey three topics related to pattern avoidance, each motivated by an interesting question:

1. Given a class C of objects and a set P of patterns from C, what is the smallest size of an object in C that contains all of the patterns in P? We will focus on recent developments in which C is either a class of words/permutations or a class of trees. The discussion of words/permutations is mostly taken from the survey article [9] as well as the more recent articles [6, 11]. The discussion of trees is taken from my article [8], which was written with Noah Kravitz and Ashwin Sah.
2. How does permutation pattern avoidance interact with the group structure of the symmetric group? The first part of this topic will concern pattern-avoiding permutations with particular cycle types; the second part will concern permutations whose powers are required to avoid a pattern. This portion of the talk will draw from the articles [1, 3, 10, 2, 4, 5].
3. If your socks come out of the laundry all mixed up, how should you sort them? We will discuss a novel foot-sorting algorithm that uses feet to attempt to sort a sock ordering; one can view this algorithm as an analogue of Knuth's stack-sorting algorithm for set partitions. This part of the talk is based on my article [7], which was written with Noah Kravitz.

References

1. K. Archer and S. Elizalde, Cyclic permutations realized by signed shifts. *J. Comb.*, **5** (2014), 1–30.
2. K. Archer and C. Graves, Pattern-restricted permutations composed of 3-cycles. *Discrete Math.*, **345** (2022).
3. M. Bóna and M. Cory, Cyclic permutations avoiding pairs of patterns of length three. *Discrete Math. Theor. Comput. Sci.*, **21** (2019).
4. M. Bóna and R. Smith, Pattern avoidance in permutations and their squares. *Discrete Math.*, **342** (2019), 3194– 3200.
5. A. Burcroff and C. Defant, Pattern-avoiding permutation powers. *Discrete Math.*, **343** (2020).
6. Z. Chroman, M. Kwan, and M Singhal, Lower bounds for superpatterns and universal sequences. *J. Combin. Theory Ser. A*, **182** (2021).
7. C. Defant and N. Kravitz, Foot-sorting for socks. arXiv:2211.02021.
8. C. Defant, N. Kravitz, and A. Sah, Supertrees. *Electron. J. Combin.*, **27** (2020).
9. M. Engen and V. Vatter, Containing all permutations. *Amer. Math. Monthly*, **128** (2021), 4–24.
10. B. Huang, An upper bound on the number of $(132, 213)$-avoiding cyclic permutations. *Discrete Math.*, **342** (2019), 1762–1771.
11. Z. Hunter, An asymptotically tight lower bound for superpatterns with small alphabets. To appear in *Comb. Theory.*

3.10 Walks in simplices, cylindric tableaux, and asymmetric exclusion processes

Sergi Elizalde (Dartmouth College – Hanover, US)

Main reference Sergi Elizalde: "Walks in Simplices, Cylindric Tableaux, and Asymmetric Exclusion Processes", Séminaire Lotharingien de Combinatoire, 86B.8 (2022), 12 pp.
URL https://www.mat.univie.ac.at/ slc/wpapers/FPSAC2022/8.html

We describe bijections between three classes of combinatorial objects that have appeared in different contexts: lattice walks in simplicial regions as introduced by Mortimer–Prellberg (in a previous Dagstuhl Seminar), standard cylindric tableaux as introduced by Gessel–Krattenthaler and Postnikov, and sequences of states in the totally asymmetric simple exclusion process. Our perspective gives new insights into these objects, providing a vehicle to translate enumerative results and certain symmetries from one setting to another. As an example, we use a cylindric analogue of the Robinson–Schensted correspondence to give an alternative bijective proof of a recent result of Courtiel, Elvey Price and Marcovici relating forward and backward walks in simplices.

3.11 Fighting fish and pattern avoiding permutations

Luca Ferrari (University of Firenze, IT)

Joint work of Lapo Cioni, Luca Ferrari, Corentin Henriet

Fighting fish are combinatorial objects recently introduced by Duchi, Guerrini, Rinaldi and Schaeffer. They are polyomino-like objects which can branch out of the plane into independent substructures. The main motivation for considering such structures lies in their remarkable probabilistic properties. In particular, it is known that the average area of fighting fish having semiperimeter n is of order $n^{5/4}$, which is a rather non-standard behaviour. The previously mentioned authors have discovered a lot of interesting combinatorics related to fighting fish. In particular, they have shown that the number of fighting fish of semiperimeter $n+1$ is given by $\frac{2}{(n+1)(2n+1)}\binom{3n}{n}$, which is the same as the number of (West)-two-stack sortable permutations of size n. This result spurred much research to better understand the relationships between these objects (and others also counted by the same sequence). Wenjie Fang was able to describe a bijection between two-stack-sortable permutations and fighting fish which accounts for the above enumerative result (and also preserves a lot of other statistics). However, his bijection is recursive, and no direct bijection is known yet. In my talk I will present a general construction which maps any permutation to a certain labelled tree, which in turn encodes a unique fighting fish. Such a construction is not bijective in general, but it becomes a bijection when restricted to two-stack-sortable permutations, thus defining the (almost) direct bijection that was missing. As a matter of fact, this construction is equivalent to Fang's one (in other words, it is a more direct version of the recursive procedure of Fang). Our hope would then be to use our construction to understand what the parameter area (on fighting fish) is on permutations, but we have not been successful yet. We have however some partial results, such as a description of those permutations (of fixed size) whose associated fighting fish has minimum area.

The sequence enumerating fighting fish with respect to semiperimeter also counts another class of permutations, namely those avoiding the two vincular patterns 3-1-4-2 and 2-41-3. Such permutations have been investigated by Claesson, Kitaev and Steingrimsson, in particular they provide a bijection with so-called $\beta(1,0)$-trees. These are trees whose recursive structure encodes somehow more naturally the recursive structures of fighting fish. This leads us to think that this class of permutations could be more useful to have a better combinatorial understanding of the area of fighting fish.

3.12 Length-4 Pattern Avoidance in Inversion Sequences

Carina Letong Hong (University of Oxford, GB)

Joint work of Carina Letong Hong, Rupert Li
Main reference Letong Hong, Rupert Li: “Length-Four Pattern Avoidance in Inversion Sequences”, Electron. J. Comb., Vol. 29(4), 2022.
URL https://doi.org//10.37236/10948

Pattern avoidance for permutations is a robust and well-established branch of enumerative combinatorics since the systematic study of Simeon and Schmidt in 1985. This talk summarizes recent progress on pattern avoidance in inversion sequences, including Mansour and Shattuck (2015), Corteel, Martinez, Savage, and Weselcouch (2016), Mansour and Yildirim (2022), and Testar (2022). Their work completely enumerated length-3 pattern avoidance except the case 120.

Recently, we completely classified all eight length-4 Wilf equivalences: $1011 = 1101 = 1110$, $2110 = 2101 = 2011$, $0221 = 0212$, $0312 = 0321$, $1102 = 1012$, $2201 = 2210$, $2301 = 2310$, $3201 = 3210 = 3012$. This talk mentions key lemmas that help establish the proofs that can be generalized to avoidance of longer patterns. Furthermore, there are four length-4 patterns whose enumeration appears in the OEIS by computer experiments: in addition to 0012 conjectured by Lin and Ma and proved by Chern, we resolve the cases 0000 and 0111 by proving the formulas for general 00...0 and 01...1 cases, giving bijections to bounded-degree label-increasing trees, and give a conjecture for 0021 which is then subsequently solved by Chern, Fu, and Lin and Mansour in 2022.

3.13 What makes permutation patterns hard to match?

Vít Jelínek (Charles University – Prague, CZ)

Joint work of Vít Jelínek, Michal Opler, Jakub Pekárek
Main reference Michal Opler: “Structural and Algorithmic Properties of Permutation Classes”, PhD thesis, Faculty of Mathematics and Physics, Charles University in Prague, 2022.
URL http://hdl.handle.net/20.500.11956/179844

Permutation Pattern Matching (or PPM) is a fundamental decision problem in the study of permutations. Its input is a pair permutations P (the “pattern”) and T (the “text”), and the goal is to determine whether P is contained in T. While PPM is NP-hard on general inputs, it often becomes tractable when P or T are restricted to a proper hereditary permutation class.

In my talk, I will present some results and conjectures that attempt to describe the boundary between tractable and hard cases of such restrictions of PPM and other related problems. Specifically, I will focus on the relationships between these three topics: computational complexity of PPM restricted to a given permutation class; structural properties of a permutation class, such as the presence of large grid-like substructures; and the growth of width parameters, like tree-width or grid-width, within a given permutation class.

3.14 Pattern avoidance: algorithmic connections

László Kozma (FU Berlin, DE)

Main reference Benjamin Aram Berendsohn, László Kozma, Dániel Marx: "Finding and Counting Permutations via CSPs", Algorithmica, Vol. 83(8), pp. 2552–2577, 2021.
URL https://doi.org//10.1007/s00453-021-00812-z

Already from the beginnings of the field, pattern-avoidance has been studied in tandem with algorithmic applications. For instance, Knuth's study of permutation-patterns was motivated by connections to stack-sorting. In this talk I will survey results from the last few years on two related lines of work: (1) the study of algorithms, in various models of computation, for detecting, counting, or enumerating patterns, and (2) the study of pattern-avoidance as a source of algorithmic "easiness": how pattern-avoidance of the input affects the complexity of seemingly unrelated algorithmic tasks.

3.15 Sorting Genomes by Prefix Double-Cut-and-Joins

Anthony Labarre (Gustave Eiffel University – Marne-la-Vallée, FR)

Joint work of Guillaume Fertin, Géraldine Jean, Anthony Labarre
Main reference Guillaume Fertin, Géraldine Jean, Anthony Labarre: "Sorting Genomes by Prefix Double-Cut-and-Joins", in Proc. of the String Processing and Information Retrieval – 29th International Symposium, SPIRE 2022, Concepción, Chile, November 8-10, 2022, Proceedings, Lecture Notes in Computer Science, Vol. 13617, pp. 178–190, Springer, 2022.
URL https://doi.org//10.1007/978-3-031-20643-6_13

A double cut-and-join (DCJ for short) is an operation that replaces two edges u, v and w, x in a graph with either u, x, v, w or u, w, v, x. This operation is of interest in a biological context, as it generalises several other well-studied mutations that are known to happen in genomes, e.g. (possibly signed) reversals or (block-)transpositions.

I consider two different graph models for representing genomes – namely, paths and perfect matchings – and study DCJs under the "prefix restriction", which forces one of the cut edges to contain the first element of the genome. The talk focuses on sorting problems using these operations, which is a popular approach to reconstructing evolutionary scenarios between species, but also has applications in the field of interconnection network design, from which the prefix restriction originates. I will present some recent results on sorting genomes using variants of prefix DCJs using both models. Namely:

- new lower bounds on sorting genomes using prefix DCJs or prefix reversals;
- a polynomial-time algorithm for sorting signed genomes by prefix DCJs; and
- a 3/2-approximation algorithm for sorting unsigned genomes by prefix DCJs.

The latter algorithm is the first polynomial-time approximation algorithm with a ratio smaller than 2 for a prefix sorting problem not known to be in P.

3.16 Computational complexity of counting and sampling

István Miklós (ELKH – Budapest, HU)

Joint work of Carina Letong Hong, István Miklós
Main reference Letong Hong, István Miklós: "A Markov chain on the solution space of edge colorings of bipartite graphs", Discret. Appl. Math., Vol. 332, pp. 7–22, 2023.
URL https://doi.org//10.1016/j.dam.2023.01.029

Whenever we can ask if a certain mathematical object with prescribed properties exists, we can also ask how many such objects exist and how to generate a random one. We can also talk about the computational complexity of these counting and sampling problems, that is, how the running time of computer programs solving these problems increases with the input size. It turns out that counting and sampling are equally hard for a large class of computational problems. Equal hardness also frequently holds for estimating weighted sums and sampling from a distribution proportional to these weights. For example, in statistical physics, these problems are sampling from the Boltzmann distribution and estimating the partition function. In this talk, we give an overview of sampling and counting complexity, then we will focus on the most powerful approach to sampling, the Markov chain Monte Carlo method.

3.17 Scaling Limits of Some Restricted Permutations

Erik Slivken (University of North Carolina Wilmington, US)

Joint work of Borga, Jacopo; Duchi, Enrica; Hoffman, Christopher; Rizzolo, Douglas;
Main reference Jacopo Borga, Enrica Duchi, Erik Slivken: "Almost square permutations are typically square", Annales de l'Institut Henri Poincaré, Probabilités et Statistiques, Vol. 57, 2021.
URL https://doi.org//10.1214/20-AIHP1135
Main reference Christopher Hoffman, Douglas Rizzolo, Erik Slivken: "Scaling limits of permutations avoiding long decreasing sequences", CoRR, Vol. abs//1911.04982, 2023.
URL https://arxiv.org/abs/1911.04982

Suppose we take a large permutation that is chosen uniformly at random and conditioned to satisfy some restriction. What does this permutation look like? The answer depends on the choice of restriction and how one decides to scale the permutation. We introduce a few scaling limits that prove useful in answering this type of question for a variety of restrictions, especially in the case of pattern-avoiding permutations. We will explore what various scaling limits say about these objects and some associated statistics (like the number of fixed points of the permutation)

3.18 Two new bijections on six-vertex configurations

Jessica Striker (North Dakota State University – Fargo, US)

Joint work of Daoji Huang, Jessica Striker
Main reference Daoji Huang, Jessica Striker: "A pipe dream perspective on totally symmetric self-complementary plane partitions", CoRR, Vol. abs//2303.10463, 2023, (Preprint).
URL https://arxiv.org/abs/2303.10463

The six-vertex model is an exactly solvable model in statistical mechanics that has been widely studied by both physicists and combinatorialists for its many lovely properties.

In this talk in two parts, we first describe joint work with Daoji Huang [2] giving a bijection between alternating sign matrices (a combinatorial manifestation of six-vertex configurations) and totally symmetric self-complementary plane partitions in the reduced, 1432-avoiding case. Finding such a bijection in full generality has been an open problem for nearly 40 years; it will be interesting to see whether this new sub-bijection may lead to further (positive or negative) results on a full bijection.

We then introduce a symmetrized version of the six-vertex model that adapts nicely from the square lattice to arbitrary 4-regular graphs embedded in a disk. By modifying certain vertex configurations, we transform these to nearly planar bipartite graphs that have regions corresponding to dimer covers of hexagonal lattices. Our underlying motivation and main result is a bijection between equivalence classes of these graphs and 4-row tableaux in such a way that promotion corresponds to rotation, yielding a web basis for SL_4. This is joint work with Christian Gaetz, Oliver Pechenik, Stephan Pfannerer, and Joshua Swanson [1].

References

1 C. Gaetz, O. Pechenik, S. Pfannerer, J. Striker, J. Swanson, *Rotation-invariant web bases from hourglass plabic graphs* (Preprint).

2 D. Huang and J. Striker, *A pipe dream perspective on totally symmetric self-complementary plane partitions*, `https://arxiv.org/abs/2303.10463` (Preprint).

3.19 Extremal theory of vertex- and edge-ordered graphs

Gábor Tardos (Alfréd Rényi Institute of Mathematics – Budapest, HU)

Turán-type extremal graph theory has been generalized in many directions. In this talk I survey the extremal theories of vertex- and edge-ordered graphs. In the vertex-ordered case the vertices of the host graph are linearly ordered and we only forbid a subgraph with a specified vertex order. As in the classical extremal graph theory, we are still looking for the maximal number of edges in a host graph on n vertices avoiding the forbidden subgraph. The analogous extremal theory for edge ordered graphs was introduced recently in a paper of Gerbner, Methuku, Nagy, Pálvölgyi, T., Vizer (2023). In contrast, the vertex ordered theory has a much richer history going back to related extremal matrix problems studied by Füredi and Hajnal in 1992.

Both theories are rich in specific results: the extremal functions of some small forbidden ordered graphs. Some of these results found applications in combinatorial geometry. Other specific forbidden patterns lead to interesting open problems.

Another direction is to find analogues of general results from classical extremal graph theory. The analogues of the Erdős-Stone-Simonovits theorem has been found in both theories: the key is to find the "correct" version of the chromatic number that applies. For vertex-ordered graphs, this is the simple notion of interval chromatic number, for edge-ordered graphs, however, the corresponding notion is surprisingly rich.

In the classical (unordered) theory we have a very simple dichotomy: forests have linear extremal functions, while other graphs have far-from-linear extremal functions. The search for the analogue of this simple observation yielded several nice results, conjectures and open problems about vertex- and edge-ordered graphs.

3.20 Pattern-avoiding affine permutations

Justin Troyka (California State University – Los Angeles, US)

Joint work of Neal Madras, Justin M. Troyka
Main reference Neal Madras, Justin M. Troyka: "Bounded affine permutations I. Pattern avoidance and enumeration", Discret. Math. Theor. Comput. Sci., Vol. 22(2), 2021.
URL https://doi.org//10.46298/dmtcs.6178

An *affine permutation of size* n is a bijection $\pi\colon \mathbb{Z} \to \mathbb{Z}$ satisfying certain properties, including that $\pi(i+n) = \pi(i) + n$ for all i. The affine permutations of size n have been much studied as a Coxeter group, but our perspective is pattern avoidance. To have a finite set to count, we can require also that $|\pi(i) - -i| < n$ for each i; such π we call a *bounded affine permutation*. We give the asymptotic number of bounded affine permutations of size n that avoid $k \dots 1$; in particular, they have the same growth rate as the ordinary permutations avoiding $k \dots 1$. We also show several results about affine permutation classes with the property that every element is a shift of the infinite direct sum of an ordinary permutation, from which we obtain exact enumerations for several affine permutation classes. This talk covers joint work with Neal Madras, published in 2021 in *Discrete Math. Theor. Comput. Sci.* and in *Ann. Comb.* Our work, especially the boundedness condition, is motivated by the fruitful concept of periodic boundary conditions in statistical physics.

References

1 N. Madras and J. M. Troyka. "Bounded affine permutations I. Pattern avoidance and enumeration". *Discrete Math. Theor. Comput. Sci.* **22**(2) (2021): #1.

2 N. Madras and J. Troyka. "Bounded affine permutations II. Avoidance of decreasing patterns". *Ann. Comb.* **25** (2021): 1007–1048.

3.21 Exploring Permutation Classes with TileScope

Henning Ulfarsson (Reykjavik University, IS)

Joint work of Henning Ulfarsson, Michael Albert, Christian Bean, Anders Claesson, Émile Nadeau, Jay Pantone

In the world of combinatorics, there are numerous sets of objects that are in a one-to-one correspondence with sets of permutations possessing specific properties, commonly characterized by pattern avoidance. This talk will delve into the TileScope algorithm and demonstrate its usefulness in comprehending permutation sets avoiding a finite list of patterns. Additionally, we will examine the algorithm's output, including polynomial counting formulas, systems of equations, uniform random generation, and more. Finally, we will highlight the algorithm's ability to discover bijections automatically, examine the atomicity of permutation classes, and preview planned future advancements. Visit `www.permpal.com` to see successful applications of the algorithm in action.

3.22 Generating Tree Method and Pattern Avoiding Inversion Sequences

Gökhan Yildirim (Bilkent University – Ankara, TR)

Joint work of Gökhan Yildirim, Ilias Kotsireas, Toufik Mansour
Main reference Ilias S. Kotsireas, Toufik Mansour, Gökhan Yildirim: "An algorithmic approach based on generating trees for enumerating pattern-avoiding inversion sequences", J. Symb. Comput., Vol. 120, p. 102231, 2024.
URL https://doi.org//10.1016/j.jsc.2023.102231

An inversion sequence of length n is an integer sequence $e = e_1 \cdots e_n$ such that $0 \leq e_i < i$ for each $0 \leq i \leq n$. We use I_n to denote the set of inversion sequences of length n. Any word τ of length k over the alphabet $[k] := \{0, 1, \cdots, k-1\}$ is called a pattern. For a given pattern τ, we use $I_n(\tau)$ to denote the set of all τ-avoiding inversion sequences of length n.

Pattern-avoiding inversion sequences were systematically studied first by Mansour and Shattuck [2] for the patterns of length three with non-repeating letters and by Corteel et al. [1] for repeating and non-repeating letters. Since then, researchers have obtained several interesting results for these combinatorial objects.

We provide an algorithmic approach based on generating trees for enumerating the pattern-avoiding inversion sequences. First, by using this algorithmic approach, we determine the generating trees for many pattern classes such as $I_n(100), I_n(011, 201), I_n(021, 0112), \cdots$. Then we obtain enumerating formulas for them through generating functions and the kernel method.

G. Yıldırım was partially supported by TUBITAK-Ardeb 120F352.

References

1 S. Corteel, M.A. Martinez, C.D. Savage and M. Weselcouch.*Patterns in inversion sequences I.* Discrete Math. Theor. Comput. Sci. 18 (2), 2016.

2 T. Mansour and M. Shattuck. *Pattern avoidance in inversion sequences.* Pure Math. Appl. 25 (2), 157–176, 2015.

4 Open problems

4.1 (More on) Combinatorial moment sequences and permutation patterns

Natasha Blitvic (Queen Mary University of London, GB)

Joint work of Natasha Blitvic, Einar Steingrimsson

Several years ago, a strange idea was hatched by a probabilist and a combinatorialist, drawing on intuition from noncommutative probability and the types of constructions found there. Namely, could the sequences arising from the study of classical permutation patterns be moment sequences of probability measures on the real line? Here was our central conjecture. Given a classical permutation pattern π, let $G_\pi^{(n)}$ be the generating function of the number of *occurrences* of π in permutations on n letters, that is,

$$G_\pi^{(0)}(q) := 1 \qquad \text{and} \qquad G_\pi^{(n)}(q) := \sum_{\sigma \in S_n} q^{\#\mathrm{occ}_\pi(\sigma)}.$$

▶ **Conjecture.** For any permutation pattern π, there is some real interval containing the origin, such that for any q in that interval, $(G_\pi^{(n)}(q))_{n\geq 0}$ is a moment sequence of some probability measure on the real line.

Given we (still) have very little numerical data, let alone very few examples of permutation patterns π for which G_π has a known expression (see the references in [1] for the two known examples of vincular patterns for which G_π is available), this might be seen as a bold conjecture. Remarkably, independently and around the same time but motivated by the numerical aspects, Tony Guttmann and Andrew Elvey Price were conjecturing the above for $q = 0$, that is, focusing on the suspected positivity of the *avoiders* of classical permutation patterns. Moreover, these appeared to give moments of probability measures on the *positive* real line.

For sequences counting avoiders of classical permutation patterns, some explicit results are available, from which positivity can be deduced using a variety of techniques [3], as well as more data thanks to state-of-the-art numerical algorithms developed for this type of enumeration (see [4] and the references therein). However, we are still far from resolving this conjecture, even in the $q = 0$ case. (Interestingly, see [2] for a non-trivial “consecutive pattern analogue” of the conjecture.) To tackle the general version given above, two distinct flavors of combinatorial results would be helpful:

- Explicit distributional results, that is closed or semi-closed expressions for G_π, for some examples of classical permutation patterns.
- Improved algorithms to enumerate the distributions of the number of occurrences of a given classical permutation pattern, in order to test the above conjecture.

References

1 N. Blitvić and E. Steingrímsson, *Permutations, Moments, Measures*, Transactions of the American Mathematical Society, Vol. 374, Number 8, August 2021, pp. 5473–5509.
2 N. Blitvić, S. M. Kammoun, E. Steingrímsson, *A new perspective on positivity in (consecutive) permutation patterns*, Proceedings of FPSAC 2023, July 17-21, Davis CA.
3 A. Bostan, A. Elvey Price, A. J. Guttmann, and J.-M. Maillard. *Stieltjes moment sequences for pattern-avoiding permutations.* Electronic Journal of Combinatorics 27.4 (2020), P4.20.
4 N. Clisby, A. R. Conway, A. J. Guttmann, and Y. Inoue, *Classical Length-5 Pattern-Avoiding Permutations*, Electronic Journal of Combinatorics 29.3 (2022), P3.14.

4.2 Distribution of sets of descent tops and descent bottoms on permutations avoiding patterns of length 4

Alexander Burstein (Howard University – Washington, US)

We conjecture the Destop-Wilf equivalence classes and (Destop, Desbot)-Wilf equivalence classes of permutation patterns of length 4. Specifically, we conjecture that the nontrivial (i.e. non-singleton) Destop-Wilf equivalence classes in S_4 are $\{1243, 3412\}$, $\{1423, 2413\}$, $\{2143, 3421\}$, $\{2314, 3124\}$, $\{2431, 3142, 3241, 4132\}$, and the only nontrivial (Destop, Desbot)-Wilf equivalence class in S_4 is $\{3142, 3241, 4132\}$. This has been verified for avoiders of size up to 10.

The conjectures in this note concern the distribution of some descent-related statistics on some pattern-avoiding permutation classes.

For basic definitions regarding patterns in permutations, we refer to Bevan [3]. We will need the following additional definitions, the first of which refines Wilf-equivalence, and the second defines additional descent-related statistics.

▶ **Definition 1.** Let f be a permutation statistic. We say that patterns σ and τ are *f-Wilf-equivalent* if there is a bijection $\Theta : \mathrm{Av}_n(\sigma) \to \mathrm{Av}_n(\tau)$ for all $n \geq 0$ that preserves the f statistic, i.e. $f = f \circ \Theta$.

▶ **Definition 2.** Given a permutation σ, and a position i such that $\sigma(i) > \sigma(i+1)$, we call i a *descent* of σ, $\sigma(i)$ a *descent top* of σ, and $\sigma(i+1)$ a *descent bottom* of σ. Likewise, if $\sigma(i) < \sigma(i+1)$, then we call i an *ascent* of σ, $\sigma(i)$ an *ascent bottom* of σ, and $\sigma(i+1)$ an *ascent top* of σ.

▶ **Notation 3.** *For a permutation σ, we define the following sets (which can also be thought of as permutation statistics):*

- $\mathrm{Des}(\sigma) = \{i \mid \sigma(i) > \sigma(i+1)\}$*, the* descent set *of σ,*
- $\mathrm{Destop}(\sigma) = \{\sigma(i) \mid i \in \mathrm{Des}(\sigma)\}$*, the* descent top set *of σ,*
- $\mathrm{Desbot}(\sigma) = \{\sigma(i+1) \mid i \in \mathrm{Des}(\sigma)\}$*, the* descent bottom set *of σ.*

We also define the sets $\mathrm{Asc}(\sigma)$*,* $\mathrm{Ascbot}(\sigma)$*,* $\mathrm{Asctop}(\sigma)$ *of ascents, ascent bottoms, and ascent tops of σ similarly.*

It is straightforward to show that patterns 132 and 231 are both Des-Wilf-equivalent and Destop-Wilf equivalent, but not (Des, Destop)-Wilf equivalent. Indeed, define maps $\phi, \psi : \mathrm{Av}(132) \to \mathrm{Av}(231)$ as follows. Given a nonempty permutation $\sigma \in \mathrm{Av}(132)$, we can write $\sigma = 231[\sigma', 1, \sigma'']$ for some $\sigma', \sigma'' \in \mathrm{Av}(132)$. Then let

$$\phi(\sigma) = 132[\phi(\sigma'), 1, \phi(\sigma'')] \quad \text{and} \quad \psi(\sigma) = 132[\psi(\sigma''), 1, \psi(\sigma')] \quad \text{if } \sigma', \sigma'' \neq \emptyset,$$

and $\phi(\sigma) = \psi(\sigma) = \sigma$ if $\sigma' = \emptyset$ or $\sigma'' = \emptyset$ or $\sigma = \emptyset$. Then

$$\mathrm{Des}(\phi(\sigma)) = \mathrm{Des}(\sigma) \quad \text{and} \quad \mathrm{Destop}(\psi(\sigma)) = \mathrm{Destop}(\sigma).$$

West [8], Stankova [7], and Bóna [5] together established the Wilf-equivalence classes of permutation patterns of length 4. Later, some of the Wilf-equivalences were shown to have the same distribution of various permutation statistics. For example, Bloom [4] showed that patterns 3142 and 4132 are Des-Wilf-equivalent.

The principal motivation for our conjectures comes from the following. A *Dumont permutation of the first kind* is a permutation σ of an even length $2n$ such that $\mathrm{Destop}(\sigma) = \{2, 4, 6, \ldots, 2n\}$. Burstein and Jones [6] and Archer and Lauderdale [2] together conjectured Wilf-equivalences of patterns of length 4 on Dumont permutations of the first kind. We generalize this by claiming that those are exactly the nontrivial Destop-Wilf equivalences for patterns of length 4 on all permutations.

▶ **Conjecture 4.** *The non-singleton* Destop*-Wilf equivalence classes for permutation patterns of length 4 are:*

- $1243 \sim 3412$*,*
- $1423 \sim 2413$*,*
- $2143 \sim 3421$*,*
- $2314 \sim 3124$*,*
- $2431 \sim 3142 \sim 3241 \sim 4132$*.*

Each of the remaining 12 patterns of length 4 is in a Destop-Wilf-equivalence class by itself. For some of the above patterns, we have an even stronger conjecture.

▶ **Conjecture 5.** *Patterns* 3142*,* 3241*,* 4132 *are* $(\mathrm{Destop}, \mathrm{Desbot})$*-Wilf equivalent.*

It is easy to see by comparing the descent tops and descent bottoms of the patterns themselves that this is the only nontrivial $(\mathrm{Destop}, \mathrm{Desbot})$-Wilf equivalence class for patterns of length 4.

Conjecture 5 can be restated by partitioning the set of all values in each permutation $\sigma \in S_n$ into four mutually disjoint parts.

- $\mathrm{Desruntop}(\sigma) = \mathrm{Destop}(\sigma) \setminus \mathrm{Desbot}(\sigma)$, the set of descent run tops of σ,
- $\mathrm{Desrunbot}(\sigma) = \mathrm{Desbot}(\sigma) \setminus \mathrm{Destop}(\sigma)$, the set of descent run bottoms of σ,
- $\mathrm{Desrunmid}(\sigma) = \mathrm{Destop}(\sigma) \cap \mathrm{Desbot}(\sigma)$, the set of descent run middles of σ,
- $\mathrm{Ascrunmid}(\sigma) = [n] \setminus (\mathrm{Destop}(\sigma) \cup \mathrm{Desbot}(\sigma))$, the set of ascent run middles of $\sigma^+ = (0, \sigma, \infty)$. This includes ascent run middles of σ together with $\sigma(1)$ if $\sigma(1) < \sigma(2)$ and $\sigma(n)$ if $\sigma(n-1) < \sigma(n)$.

▶ **Conjecture 6.** *Patterns* 3142*,* 3241*,* 4132 *are* $(\mathrm{Desruntop}, \mathrm{Desrunbot}, \mathrm{Desrunmid}, \mathrm{Ascrunmid})$*-Wilf equivalent.*

Both Conjectures 4 and 5 have been verified for avoiders of length $n \leq 10$ with the help of Michael Albert's *PermLab* [1] software.

References

1 M. Albert, *PermLab*, https://www.cs.otago.ac.nz/PermLab.

2 K. Archer, L.-K. Lauderdale, personal communication, 2019.

3 D. Bevan, Permutation patterns: basic definitions and notation, arXiv:1506.06673.

4 J. Bloom, A refinement of Wilf-equivalence for patterns of length 4, *J. Combin. Theory, Ser. A* **124** (2014), 166–177.

5 M. Bóna, Permutations avoiding certain patterns; The case of length 4 and generalizations, *Discrete Math.* **175** (1997), 55–67.

6 A. Burstein, O. Jones, Enumeration of Dumont permutations avoiding certain four-letter patterns, *Discrete Math. and Theor. Comp. Sci.*, **22**:2 (2021), #7.

7 Z. Stankova, Forbidden subsequences, *Discrete Math.* **132** (1994), 291–316.

8 J. West, Permutations with restricted subsequences and stack-sortable permutations, Ph.D. Thesis, MIT, 1990.

4.3 A curious mesh pattern

Anders Claesson (University of Iceland – Reykjavik, IS)

Main reference Anders Claesson: "From Hertzsprung's problem to pattern-rewriting systems", Algebraic Combinatorics, Vol. 5(6), pp. 1257–1277, The Combinatorics Consortium, 2022.
URL https://doi.org//10.5802/alco.202

I would like to bring attention to a particular mesh pattern. It is of length 3 and enumerating the permutations avoiding it is an open problem. Moreover, this pattern shares some features with the Hertzsprung patterns, yet it does not fall into that category.

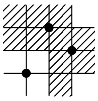
Are there nontrivial mesh-patterns p other than the Hertzsprung patterns for which there is a rational function $R(x)$ such that

$$\sum_{n\geq 0} |\mathcal{S}(p)|x^n = \sum_{m\geq 0} m!R(x)^m\,?$$

It appears that the answer is yes.

▶ **Conjecture 1.** *It holds that*

$$\sum_{n\geq 0} |\mathcal{S}_n(p)|x^n \;=\; \sum_{m\geq 0} m!\left(\frac{x}{1+x^2}\right)^m, \quad \textit{where } p =$$



This conjecture is based on computing the numbers $|\mathcal{S}_n(p)|$ for $n \leq 14$:

$$1, 1, 2, 5, 20, 103, 630, 4475, 36232, 329341, 3320890, 36787889,$$
$$444125628, 5803850515, 81625106990$$

At the time of writing this sequence is not in the OEIS.

4.4 Universal Set Partitions

Colin Defant (MIT – Cambridge, US)

In 2000, Klazar introduced a natural notion of pattern containment/avoidance for set partitions [1, 2]. Let (A, B) be one of the following pairs of phrases:

- (set partition, set partition);
- (set partition, noncrossing partition);
- (set partition, nonnesting partition);
- (noncrossing partition, noncrossing partition);
- (nonnesting partition, nonnesting partition).

For each positive integer k, what is the smallest size of an A that contains all B's of size k as patterns?

References

1 M. Klazar, Counting pattern-free set partitions I: a Generalization of Stirling numbers of the second kind. *European J. Combin.*, **21** (2000), 367–378.

2 M. Klazar, Counting pattern-free set partitions II: noncrossing and other hypergraphs. *Electron. J. Combin.*, **7** (2000).

4.5 Two conjectures on quasi-kernels

Péter L. Erdös (Alfréd Rényi Institute of Mathematics – Budapest, HU)

Let $D = (V, \vec{E})$ be (finite or infinite) directed graph. An independent vertex subset $A \subset V$ is a *quasi-kernel* (also known as *semi-kernel*) iff for each point v there is a path of length at most 2 from some point of A to v. Similarly, the independent vertex subset $B \subset V$ is a *quasi-sink*, iff for each point v there is a path of length at most 2 from v to some point of A.

It is a well-known fact, that every finite (table-tennis) tournament has a (single-point) quasi-kernel (quasi-sink). In 1973 the following nice generalization was proved:

▶ **Theorem 1** (V. Chvátal – L. Lovász [1]). *Every finite directed graph contains a quasi-kernel (quasi-sink).*

In 1976 the following conjecture was stated:

▶ **Conjecture 2** (P.L. Erdős – L. A. Székely). *Assume that in the finite digraph $D = (V, \vec{E})$ for each vertex $v \in V$ the* indegree *$d^-(v) \geq 1$. Then there exists a quasi-kernel A with the property $|A| \leq |V|/2$. For example the disjoint union of oriented C_4's satisfies this with equality.*

Theorem 1 does not hold in case of infinite directed graphs: for example if Z denotes the directed graph of all integers where each edge is directed "upwards", clearly there is nor quasi-kernel neither quasi-sink. However its vertex set can be easily partitioned into two subsets, such that one spanned sub-tournament contains a quasi-kernel, while the other one contains a quasi-sink.

▶ **Conjecture 3** (P.L. Erdős – L. Soukup (2008) [2]). *Every (countable) infinite directed graph D can be partitioned into two vertex classes, such that one spanned subgraph contains a quasi-kernel, while the other one contains a quasi-sink.*

References

1 V. Chvátal – L. Lovász: Every directed graph has a semi-kernel, *Lecture Notes in Math* **411** (1974), 175.

2 P.L. Erdős – L. Soukup: Quasi-kernels and quasi-sinks in infinite graphs, to appear in *Disc. Math* (2008), 1–18.

4.6 Sorting by parallel block reversals

Vít Jelínek (Charles University – Prague, CZ)

Joint work of Vít Jelínek, Michal Opler, Jakub Pekárek

Suppose that we are given a sequence of n numbers, which we want to sort into ascending order by using the following iterative procedure: in each round, we partition the current sequence arbitrarily into disjoint blocks of entries in consecutive positions, not necessarily of the same length, and then in a single step we reverse the order of entries within each block. What is the smallest number of rounds needed to sort any input of length n? Equivalently:

what is the smallest number $K(n)$ such that any permutation of length n can be obtained by composing $K(n)$ layered permutations? A counting argument shows that $\Omega(\log n)$ rounds are sometimes needed, and I can show that $O(\log^2 n)$ rounds suffice, so the problem is to close this gap.

4.7 Füredi-Hajnal limits of permutations

Jan Kyncl (Charles University – Prague, CZ)

Joint work of Josef Cibulka, Jan Kynčl
Main reference Josef Cibulka, Jan Kynčl: "Better upper bounds on the Füredi-Hajnal limits of permutations", CoRR, Vol. abs/1607.07491v3, 2019
URL https://arxiv.org/abs/1607.07491v3

Füredi–Hajnal limits of permutations

A *binary matrix* is a matrix with entries from the set $\{0, 1\}$. We say that a binary matrix A *contains* a binary matrix S if S can be obtained from A by removal of some rows, some columns, and changing some 1-entries to 0-entries. If A does not contain S, we say that A *avoids* S. A *k-permutation matrix* P is a binary $k \times k$ matrix with exactly one 1-entry in every row and one 1-entry in every column.

The Füredi–Hajnal conjecture, proved by Marcus and Tardos, states that for every permutation matrix P, there is a (smallest) constant c_P such that for every $n \in \mathbb{N}$, every $n \times n$ binary matrix A with at least $c_P \cdot n$ 1-entries contains P.

The proof by Marcus and Tardos [3] implies the upper bound $c_P \leq 2k^4\binom{k^2}{k}$ for every k-permutation matrix P. Fox [2] improved the upper bound to $c_P \leq 3k2^{8k}$. Our current best upper bound is $c_P \leq \frac{8}{3}(k+1)^2 \cdot 2^{4k}$ [1].

For the lower bound, Fox [2] gave a randomized construction showing that for every k, there are k-permutation matrices P with $c_P \geq 2^{\Omega(k^{1/2})}$.

For random k-permutation matrices, we can prove that $c_P \leq 2^{O(k^{2/3}\log^{7/3} k/(\log\log k)^{1/3})}$ asymptotically almost surely [1]. In particular, we can prove subexponential upper bounds for so-called *scattered* permutation matrices and a few specific examples of non-scattered permutation matrices. However, there are still many cases where we do not have better than exponential upper bound on c_P.

A *two-diagonal* matrix is a binary matrix whose all 1-entries lie on two parallel diagonals, which may be arbitrarily far apart.

Let P be k-permutation matrix contained in a two-diagonal matrix. Is $c_P \leq 2^{o(k)}$? Is c_P polynomial in k?

References

1 J. Cibulka and J. Kynčl, Better upper bounds on the Füredi–Hajnal limits of permutations (2019), arXiv:1607.07491v3.

2 J. Fox, Stanley–Wilf limits are typically exponential (2013), arXiv:1310.8378v1.

3 A. Marcus and G. Tardos, Excluded permutation matrices and the Stanley–Wilf conjecture, *J. Combin. Theory Ser. A* **107**(1) (2004), 153–160.

4.8 Number of Successful Pressing sequences of black-and-white lines

István Miklós (ELKH – Budapest, HU)

Main reference Eliot Bixby, Toby Flint, István Miklós: "Proving the Pressing Game Conjecture on Linear Graphs", CoRR, Vol. abs/1303.6799, 2013.
URL http://arxiv.org/abs/1303.6799

Let a line graph L be given whose vertices are colored by black and white. Black vertices can be pressed. Pressing a black vertex has an effect that the black vertex is deleted, the neighbors of the black vertex change color and become neighbors (when there are two neighbors). A successful press- ing sequence of L is a permutation of its vertices such that the vertices can be pressed in that order and the result is the empty graph (that is, the last step must be pressing the remaining single black vertex). The open question is the computational complexity of counting the success- ful pressing sequences of a black-and-white line graph. That is, is there a polynomial running time algorithm to compute this number or is the problem #P-complete? It is also an interesting problem to prove rapid mixing of an irreducible Markov chain on successful pressing sequences, see Bixby *et al.* (that paper also tells the motivation of the problem, which is actually a genome rearrangement problem).

References

1 Bixby, E, Flint, T, Miklós, I. . Proving the Pressing Game Conjec- ture on Linear Graphs. Involve, 9(1):41–56. 2016.

4.9 Particle systems for the longest pattern-avoiding subsequences for permutations

Gökhan Yildirim (Bilkent University – Ankara, TR)

The longest increasing subsequence problem for permutations can be rephrased as the longest 21-avoiding subsequence. This leads us to generalize the problem to the longest τ-avoiding subsequence for any given pattern τ [1, 2].

Hammersley's interacting particle process on the unit interval [0,1] corresponds to the longest 21-avoiding subsequence. This particle process is in KPZ universality class. It provides an efficient algorithm to numerically understand the distributional properties of the longest 21-avoiding subsequence problem under the uniform measure(21-case is solved theoretically).

Can we find particle processes corresponding to the longest τ-avoiding subsequence for a given pattern τ? Specifically for patterns of length 3 except 321, which is known [3].

References

1 R. P. Stanley. *Increasing and Decreasing Subsequences and Their Variants.* Proceedings of the International Congress of Mathematicians, Madrid, Spain, 2006, pp. 545–579.

2 M. H. Albert. *On the length of the longest subsequence avoiding an arbitrary pattern in a random permutation.* Random Struct. Algorithms 31 (2007) 227–238

3 A. Atalik, H. S. M. Erol, G. Yıldırım, and M. Yilmaz. *Variations on Hammersley's interacting particle process.* Discrete Math. Lett. 7 (2021) 34–39.

4.10 Linear temporal spanners in temporal cliques

Victor Zamaraev (University of Liverpool, GB)

A *temporal graph* $\mathcal{G}$ is a pair (G, λ), where $G = (V, E)$ is a simple undirected graph and λ is a function that assigns to every edge e of G a finite non-empty set of natural numbers. The temporal graph is *simple* if (1) every edge of G is assigned exactly one time label, i.e., $|\lambda(e)| = 1$ for every $e \in E$; and (2) no two edges get assigned the same time label. Without loss of generality, we will assume that if is simple, then λ is a bijection between E and $\{1, 2, \dots, |E|\}$. For this problem we focus only on simple temporal graphs.

A *temporal* (u, v)*-path* or a *temporal path* from u to v in $\mathcal{G}$ is a path $u = u_0, u_1, \dots, u_\ell = v$ in G such that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_\ell$, where $\lambda_i \in \lambda(u_{i-1}u_i)$.

The temporal graph $\mathcal{G}$ is *temporally connected* if each vertex can reach every other vertex by a temporal path. A temporal graph $\mathcal{G}' = (G', \lambda')$ is a *temporal subgraph* of $\mathcal{G}$ if G' is a subgraph of G and λ' is the restriction of λ to the edges of G'. Furthermore, if $V(G') = V(G)$ and $'$ is temporally connected, then $\mathcal{G}'$ is called a *temporal spanner* of $\mathcal{G}$.

▶ **Problem 1.** *Let $\mathcal{G}$ be an arbitrary simple* temporal clique *on n vertices, i.e. $\mathcal{G} = (K_n, \lambda)$, where K_n is a complete graph on n vertices. Is it true that $\mathcal{G}$ has a temporal spanner with $O(n)$ edges?*

It is known that any temporal clique on n vertices has a temporal spanner with $O(n \log n)$ edges [1]. Whether this bound is order optimal or not is open.

References

1 Arnaud Casteigts, Joseph G. Peters, Jason Schoeters. *Temporal cliques admit sparse spanners.* Journal of Computer and System Sciences, 121, p. 1–17, 2021

Participants

- Benjamin Aram Berendsohn
FU Berlin, DE
- David Bevan
University of Strathclyde – Glasgow, GB
- Natasha Blitvic
Queen Mary University of London, GB
- Miklós Bóna
University of Florida – Gainesville, US
- Mathilde Bouvel
LORIA – Nancy, FR
- Robert Brignall
The Open University – Milton Keynes, GB
- Alexander Burstein
Howard University – Washington, US
- Parinya Chalermsook
Aalto University, FI
- Anders Claesson
University of Iceland – Reykjavik, IS
- Sylvie Corteel
Université Paris Cité, FR
- Péter Csikvári
Alfréd Rényi Institute of Mathematics – Budapest, HU
- Radu Curticapean
IT University of Copenhagen, DK
- Colin Defant
MIT – Cambridge, US
- Sergi Elizalde
Dartmouth College – Hanover, US
- Péter L. Erdös
Alfréd Rényi Institute of Mathematics – Budapest, HU
- Luca Ferrari
University of Firenze, IT
- Sylvie Hamel
University of Montréal, CA
- Carina Letong Hong
University of Oxford, GB
- Vít Jelínek
Charles University – Prague, CZ
- László Kozma
FU Berlin, DE
- Jan Kyncl
Charles University – Prague, CZ
- Anthony Labarre
Gustave Eiffel University – Marne-la-Vallée, FR
- István Miklós
ELKH – Budapest, HU
- Torsten Mütze
University of Warwick – Coventry, GB
- Michal Opler
Czech Technical University – Prague, CZ
- Lara K. Pudwell
Valparaiso University, US
- Erik Slivken
University of North Carolina Wilmington, US
- Rebecca Smith
The College at Brockport, US
- Jessica Striker
North Dakota State University – Fargo, US
- Gábor Tardos
Alfréd Rényi Institute of Mathematics – Budapest, HU
- Bridget Tenner
DePaul Uniersity – Chicago, US
- Justin Troyka
California State University – Los Angeles, US
- Henning Ulfarsson
Reykjavik University, IS
- Gökhan Yildirim
Bilkent University – Ankara, TR
- Sorrachai Yingchareonthawornchai
Aalto University, FI
- Victor Zamaraev
University of Liverpool, GB

Report from Dagstuhl Seminar 23122

Deep Continual Learning

Tinne Tuytelaars*[1], Bing Liu*[2], Vincenzo Lomonaco*[3], Gido van de Ven*[4], and Andrea Cossu†[5]

1 **KU Leuven, BE.** tinne.tuytelaars@esat.kuleuven.be
2 **University of Illinois – Chicago, US.** liub@uic.edu
3 **University of Pisa, IT.** vincenzo.lomonaco@unipi.it
4 **KU Leuven, BE.** gido.vandeven@kuleuven.be
5 **University of Pisa, IT.** andrea.cossu@sns.it

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23122 "Deep Continual Learning". This seminar brought together 26 researchers to discuss open problems and future directions of Continual Learning. The discussion revolved around key properties and the definition of Continual Learning itself, on the way Continual Learning should be evaluated, and on its real-world applications beyond academic research.

Seminar March 19–24, 2023 – https://www.dagstuhl.de/23122
2012 ACM Subject Classification Computing methodologies → Learning settings; Computing methodologies → Neural networks
Keywords and phrases continual learning, incremental learning
Digital Object Identifier 10.4230/DagRep.13.3.74

1 Executive Summary

Bing Liu (University of Illinois – Chicago, US)
Vincenzo Lomonaco (University of Pisa, IT)
Tinne Tuytelaars (KU Leuven, BE)
Gido van de Ven (KU Leuven, BE)

Continual learning, also referred to as lifelong learning, is a sub-field of machine learning that focuses on the challenging problem of incrementally training models for sequentially arriving tasks and/or when data distributions vary over time. Such non-stationarity calls for learning algorithms that can acquire new knowledge over time with minimal forgetting of what they have learned previously, transfer knowledge across tasks, and smoothly adapt to new circumstances as needed. This is in contrast with the traditional setting of machine learning, which typically builds on the premise that all data, both for training and testing, are sampled i.i.d. from a single, stationary data distribution.

Deep learning models in particular are in need of continual learning capabilities. A first reason for this is the strong data-dependence of these models. When trained on a stream of data whose underlying distribution changes over time, deep learning models tend to almost fully adapt to the most recently seen data, thereby "catastrophically" forgetting the skills that have been learned earlier. Second, continual learning capabilities can be especially

* Editor / Organizer
† Editorial Assistant / Collector

beneficial for deep learning models as they can help deal with the very long training time of these models. The current practice in industry is to re-train on a regular basis to add new skills and to prevent the knowledge learned previously from being outdated. Re-training is time inefficient, unsustainable and sub-optimal. Freezing the feature extraction layers is often not an option, as the power of deep learning in many challenging applications, be it in computer vision, natural language processing or audio processing, hinges on the learned representations.

The objective of the seminar was to bring together world-class researchers in the field of deep continual learning, as well as in the related fields of online learning, meta-learning, Bayesian deep learning, robotics and neuroscience, to discuss and to brainstorm, and to set the research agenda for years to come.

During the seminar, participants presented new ideas and recent findings from their research in plenary sessions that triggered many interesting discussions. There were also several tutorials that helped create a shared understanding of similarities and differences between continual learning and other related fields. Specifically, the relation with online learning and streaming learning was discussed in detail. Furthermore, there were several breakout discussion sessions in which open research questions and points of controversy within the continual learning field were discussed. An important outcome of the seminar is the shared feeling that the scope and potential benefit of the research on deep continual learning should be communicated better to computer scientists outside of our subfield. Following up on this, most of the seminar participants are currently collaborating on writing a perspective article to do so.

2 Table of Contents

Executive Summary

Overview of Tutorials

Overview of Talks

Working groups

3 Overview of Tutorials

3.1 Deep Continual Learning

Gido van de Ven (KU Leuven, BE)

Incrementally learning new information from a non-stationary stream of data, referred to as "continual learning", is a key feature of natural intelligence, but an open challenge for deep learning. For example, standard deep neural networks tend to catastrophically forget previous tasks or data distributions when trained on a new one. Enabling these networks to incrementally learn, and retain, information from different contexts has become a topic of intense research. In the first half of this tutorial I introduce the continual learning problem. After covering some key terminology, I discuss three different types of continual learning, each with their own set of challenges: task-incremental, domain-incremental and class-incremental learning. I also cover the distinction between task-based and task-free continual learning. I end this part of the tutorial with a general framework for continual learning unifiying these different aspects. In the second half of the tutorial I review approaches that have been proposed for addressing the continual learning problem. I do this at the level of computational strategies, distinguishing between the following: (1) using context-specific components, (2) parameter regularization, (3) functional regularization, (4) replay, and (5) template-based classification. For each strategy I highlight two representative example methods.

3.2 Neuroscience inspired continual learning

Dhireesha Kudithipudi (University of Texas – San Antonio, US)

Continual learning is commonplace in humans and other mammals, but has proven difficult to achieve in artificial systems. By leveraging findings from neuroscience we can make progress towards designing continual learning AI. In this tutorial, we present the key features desirable in a continual learning system and how brain-inspired mechanisms for regularization, dynamic architectures and replay can be realized in artificial systems. Specific examples of metaplasticity, synaptic consolidation and neurogenesis are delved into closely. A canonical theme in these neuro-inspired approaches is that they can be performed at extreme low energy. We present a case for such framework.

3.3 A Light Introduction to Online Algorithms and Concept Drift

Joao Gama (INESC TEC – Porto, PT)

In this tutorial we present the basic concepts about online learning from data streams. In the first part of the tutorial, we present Hoeffding algorithms for learning decision trees, regression trees, decision and regression rules, bagging, boosting and random forests. The second part covers concept drift topics. We discuss data management, detection methods, adaptation methods and model management methods to deal with non-stationary data. We present few illustrative algorithms for explicitly drift detection. We end the tutorial, presenting open-source software available that implement most of the algorithms we discuss in the tutorial.

4 Overview of Talks

4.1 Replay free representation learning

Rahaf Aljundi (Toyota Motor Europe – Zaventem, BE)

This talk will focus on the effectiveness of representation learning as opposed to directly optimizing a classifier. With that we aim for replay free efficient methods and we explore how and when to adapt pretrained representations.

4.2 Reinventing science as a long-term ensemble learning machine

Matthias Bethge (Universität Tübingen, DE)

Foundation models such as GPT-4 have demonstrated striking task generality based on massively increasing the amount of training data and model capacity. The quest for unifying models in science as well as the strong grounding in empirical data and evaluation of models raises the question for opportunities and limitations of the current avenue to such foundation models. Despite the widespread scientific impact of models like Alphafold-2 and MedPALM, a large range of scientific questions are still hard to approach within a unified benchmarking approach. The impressive flexibility of recent large language models due to their zero-shot and in-context adaptation capabilities may help overcome this limitation – however, they are only developed by a small group of people and not designed for easy updating. In science we want models that are revisable by anyone, calling for the possibility of continual model evaluation and updating. In order to achieve such an efficient continual model extensibility (Mn+1 = f(Mn, U), with n arbitrary large), I argue that the key challenge is to modularize continual learning without sacrificing the power and scalability of current LLMs. A large part of current continual learning research aims at developing a better understanding of how stochastic gradient descent (SGD) learning is affected by the curriculum, i.e. by the order at which the data is processed. The focus lies on avoiding catastrophic forgetting rather than achieving modularity. I argue to focus on “Scalable Compositionality Discovery” (SCD) as the key challenge to overcome the limitations of collective continual foundation model building that could (1) make large scale data-driven learning ubiquitously useful for science, and (2) solve the credit assignment problem underlying catastrophic forgetting. I conclude with a super brief sketch of how current model benchmarking can be turned into an integrative ensemble learning approach for collective model building.

4.3 Beyond Forgetting with Continual Pre-Training

Andrea Cossu (University of Pisa, IT)

Pre-trained models are widely used in continual learning. They allow to leverage general and robust representations that can be then fine-tuned during continual learning. However, the existing continual learning scenarios do not fully exploit the potential of pre-trained models. We will present the Continual Pre-Training scenario, which keeps a pre-trained model updated over time. Under appropriate conditions, Continual Pre-Training proves to be surprisingly resilient to forgetting. We will discuss the relationship between Continual Pre-Training and existing paradigms, as well as its potential impact on both continual learning research and applications.

4.4 Explaining Change – Towards Online Explanations on Data Streams

Fabian Fumagalli (Universität Bielefeld, DE)

Recent advances in deep learning methods have shown impressive improvements in predictive accuracy in many tasks at the cost of interpretability. Explainable Artificial Intelligence (XAI) has emerged to understand the reasoning of such black-box models. However, XAI has mainly considered static learning scenarios, whereas many real-world applications require dynamic models that constantly adapt over time. In extreme cases, models learn incrementally on a data stream, where observations are used only once to update the model and are then discarded. In this talk, we present incremental SAGE, an efficient incremental variant of the well-established model-agnostic global feature importance method SAGE (Covert et al., 2020). We describe a general framework to efficiently compute these feature importance values in a data stream scenario with concept drift and present an open-source implementation of our method. Beyond incremental learning on data streams, we explore and discuss further applications of incremental XAI in other areas of deep continual learning.

4.5 XPM-Explainable Predictive Maintenance

Joao Gama (INESC TEC – Porto, PT)

Predictive Maintenance applications are increasingly complex, with interactions between many components. Black-box models, based on deep-learning techniques, are popular approaches due to their predictive accuracy. This talk presents a neural-symbolic architecture that uses an online rule-learning algorithm to explain when the black-box model predicts failures. The proposed system solves two problems in parallel: (i) anomaly detection and (ii) explanation of the anomaly. For the first problem, we use an unsupervised state-of- the-art autoencoder. For the second problem, we train a rule learning system that learns a mapping from the input features to the reconstruction error of the autoencoder. Both systems run online and in parallel. The autoencoder signals an alarm for the examples with a reconstruction error that exceeds a threshold. The causes of the signal alarm are hard to understand by humans because they are the result of a non-linear combination of the sensor data. The rule that triggers that example describes the relationship between the input features and the autoencoder's reconstruction error. The rule explains the failure signal in that it indicates which sensors contribute to the alarm and allows the identification of the component involved in the failure. The system can present global explanations that model the black-box model and local explanations that describe why the black-box model predicts a failure. We evaluate the proposed system in a real-world case study of Metro do Porto.

4.6 Replay-based continual learning with constant time complexity

Alexander Geppert (Hochschule für Angewandte Wissenschaften Fulda, DE)

This talk describes a new CL approach based on generative replay (GR). The salient point is that GR time complexity does not increase over time but stays constant, under some mild assumptions.

GR protects existing knowledge by having auxiliary generator networks replay/generate samples from previous sub-tasks. At each sub-task, the union of new and replayed data is then used for training a new model (or scholar). The innovation we propose is to replay only samples that cause conflicts with new data. In contrast, existing GR approaches replay all of the previously acquired knowledge, which leads to an unbounded increase in computation time.

In order to achieve constant time-complexity GR, we propose to use a GMM-based generator/solver structure that allows selective modification of existing knowledge only where it overlaps with new data. The same generator/solver can be queried with new data, selectively replaying samples from overlapping areas only. Thus, we can maintain a constant ratio between new and generated samples, irrespective of the number of sub-tasks already processed.

We tested the proposed strategy on CL problems from visual classification and found that it compares very favorably to VAE-based GR, despite vastly inferior model complexity.

4.7 Lifelong Learning: Where Do We Go Next?

Tyler Hayes (NAVER Labs Europe – Meylan, FR)

The last few years have seen immense progress in developing lifelong learning models capable of performing tasks such as incremental image classification (e.g., on ImageNet). However, today's lifelong learning models still lack the necessary capabilities to generalize to and discover novel concepts in an open world. In this talk, I outline several future research directions for lifelong learning, what advantages they offer, and initial research questions to be addressed in these areas.

4.8 Uncertainty Representation in Continual and Online Learning: Challenges and Opportunities

Eyke Hüllermeier (LMU München, DE)

The notion of uncertainty has recently drawn increasing attention in machine learning research due to the field's burgeoning relevance for practical applications, many of which have safety requirements. This talk will elaborate on the representation and adequate handling of (predictive) uncertainty in (supervised) machine learning. In this regard, the usefulness of distinguishing between two important types of uncertainty, often referred to as aleatoric and epistemic, will be elucidated. Finally, some challenges and opportunities of uncertainty handling in the realm of continual learning will be highlighted.

4.9 Let's Get Continual Learning Out of the Lab!

Christopher Kanan (University of Rochester, US)

Continual learning has been a heavily researched topic over the past six years, with mitigation of catastrophic forgetting being the primary focus. However, I argue that there is a lot more to continual learning than catastrophic forgetting. Moreover, many of the systems being created do not have the characteristics needed for real-world applications. In this talk, I outline four real-world applications for continual learning: 1) efficiently updating large neural network models, 2) learning on embedded devices, 3) enabling more efficient learning algorithms, and 4) facilitating applications such as open world learning. I describe the properties that an ideal continual learning method would need for these problem areas. I then describe a new algorithm from my research group that attempts to meet many of these criteria.

4.10 Continual domain generalization/adaptation

Tatsuya Konishi (KDDI – Saitama, JP)

Many studies have been done for the domain-shift in continual learning. Some papers have tackled this issue by techniques of test-time adaptation, but those methods depend on an already pre-trained model. We believe it would be beneficial to propose a continual pre-training procedure that is aware of possible future domain-shifts from the perspective of both domain generalization and adaptation. We present preliminary results about this problem.

4.11 Continual Learning Theory?

Christoph H. Lampert (IST Austria – Klosterneuburg, AT)

We introduce some of the fundamental concepts and results of statistical learning theory in the PAC-Bayesian setting. Afterwards, we discuss the special case of representation learning from multiple tasks and –time permitting– extensions to the continual learning regime.

4.12 Class-Incremental Learning and Open-world Continual Learning

Bing Liu (University of Illinois – Chicago, US)

Continual learning (CL) learns a sequence of tasks incrementally. A challenging setting of CL is class incremental learning (CIL). While it is well known that catastrophic forgetting (CF) is a major difficulty for CIL, we argue that there is also an equally challenging problem of inter-task class separation (ICS). This talk first presents a theoretical investigation on how to solve the CIL problem. The key results are (1) that the necessary and sufficient conditions for good CIL are good within-task prediction and task-id prediction, and (2) that task-id prediction is correlated with out-of-distribution (OOD) detection. The theory thus states that good within-task prediction and OOD detection are necessary and sufficient conditions for good CIL. This theory is also applicable to open-world learning. I will then present a general framework for open world learning, called Self-initiated Open-world continual Learning & Adaptation (SOLA).

4.13 Learning Continually from Compressed Knowledge and Skills

Vincenzo Lomonaco (University of Pisa, IT)

Learning continually from non-stationary data streams is a challenging research topic of growing popularity in the last few years. Being able to learn, adapt, and generalize continually in an efficient, effective, and scalable way is fundamental for a sustainable development of Artificial Intelligent systems. However, an agent-centric view of continual learning requires learning directly from raw data (i.e. by trial and error), which limits the efficiency, effectiveness and privacy of current solutions. Instead, we argue that continual learning systems should exploit the availability of compressed knowledge and skills in the form of trained models made globally available from a decentralized network of independent agents. In this talk, we suggest to investigate this new paradigm, also known as “Ex-Model Continual Learning” (ExML), where an agent learns from a sequence of previously trained models instead of raw data.

4.14 Into the Unknown: Premises, Pitfalls, Promises

Martin Mundt (TU Darmstadt, DE)

Deep neural networks excel in many areas seems to be a common conclusion drawn from their success on predefined training and dedicated test set data. When moving beyond this paradigm to learning data sequentially, we seem to draw similar conclusions when we find techniques that transfer knowledge and avoid forgetting over time. However, the real world is full of novel and unknown experiences, its complexity cannot be captured by benchmarking knowledge accumulation alone. In this presentation, I will talk upon design of lifelong learning systems in open worlds. These systems are able to robustly deal with novel situations and incorporate new knowledge from data streams over time as humans do. To this end, I will dive into symbiotic mechanisms for deep models to prevent erratic predictions for unknown concepts, actively query new data, and avoid rapidly forgetting past knowledge when learning on new tasks. I will then finish by revisiting the challenge of evaluation of such complex systems and means to promote reproducibility.

4.15 Role of CL in large scale learning

Razvan Pascanu (DeepMind – London, GB)

In this talk I will focus on what could be the goals of Continual Learning, particularly for typical Deep Learning settings. Firstly I will show that deep learning is fundamentally computationally inefficient due to interference or forgetting, which leads to concepts being learnt sequentially even if they are all present at once. This leads to the hypothesis that learning efficiently might require us to figure out how to learn continually, which can be a well formed target for continual learning. Afterwards I will describe some limitations of typical train-test setup, and argue that continual learning can be seen as a change of perspective that can allow rephrasing several concepts and find new ways to address these limitations. For example, it can alter how we think about evaluation at large scale. Finally I will enumerate some research directions for continual learning that I feel are receiving less attention than they should.

4.16 Transfer-learning-based exemplar-free incremental learning

Adrian Popescu (CEA LIST – Nano-INNOV, FR)

The effect of catastrophic forgetting is strong when storage of exemplars for past classes is impossible. Most existing methods designed for this scenario implement variants of fine tuning with knowledge distillation to reduce forgetting. This presentation discusses transfer-learning-based methods, which use a fixed model learned with the initial classes and the update only the classification layer during the incremental process. Experiments with different datasets and incremental splits show that transfer-based methods obtain competitive performance, while being much faster to train than mainstream fine-tuning methods. These results resonate with past works which show that simple methods can be highly effective in incremental learning, and question our progress in the exemplar-free scenario.

4.17 Repetition and Reconstruction in Continual Learning

James M. Rehg (Georgia Institute of Technology – Atlanta, US)

This talk describes some recent advances that shed light on the role of forgetting in continual learning (CL). First, we introduce CL with repeated exposures, in which sequentially-presented concepts are allowed to repeat a small number of times. We show that simple memory-based CL methods can converge to accuracy approaching batch learning in this setting. Second, we introduce a class of continual reconstruction tasks which do not suffer from forgetting in either the single or repeated exposure settings This finding is based on a novel SOTA method for single image shape reconstruction (Thai 20). We further show that shape reconstruction can be used as a proxy task for continual classification, resulting in SOTA performance. We close by developing some links between 3D reconstruction and self-supervised learning.

4.18 Using Generative Models for Continual Learning

Andreas Tolias (Baylor College of Medicine – Houston, US)

Continual learning is a key feature of natural intelligence, but an unsolved problem in deep learning. Particularly challenging for deep neural networks is “class-incremental learning”, whereby a network must learn to distinguish between classes that are not observed together. In this short talk, I will discuss two ways in which generative models can be used to address the class-incremental learning problem. The first one is “generative replay” (e.g., van de Ven et al., 2020 Nat Commun). With this approach, typically two models are learned: a classifier network and an additional generative model. Then, when learning new classes, samples from the generative model are interleaved – or replayed – along with the training data of the new classes. The second approach is “generative classification” (e.g., van de Ven et al., 2021 CVPR-W). With this approach, rather than using a generative model indirectly for generating samples to train a discriminative classifier on (as is done with generative replay), the generative model is used directly to perform classification using Bayes’ rule.

4.19 How we applied Continual Learning for Long-sequence Neural Rendering

Tinne Tuytelaars (KU Leuven, BE)

The focus in most literature on Continual Learning lies on image classification problems. In that context, it makes sense to reason about the learning process in terms of the learned representation (penultimate layer of the network), which is the part that is shared over all tasks. It's often argued that a good representation makes it easy to learn new tasks and leads to minimal forgetting. It is not clear though how these observations generalize to continual learning beyond classification tasks. In this work, we apply continual learning in a very different context, that of neural rendering. We argue there is an opportunity for continual learning in this setting if one wants to process long-sequences, as it is impossible to load all views for all timestamps in memory simultaneously, multiple views of the same timestamp are required in the same batch to learn effectively from intersecting rays, and repeatedly decoding and transferring views to/from memory is expensive. The standard architecture used for Neural Radiance Fields is not well suited for continual learning though, as the model itself is basically the representation: all properties of the dynamic scene are stored implicitly in the model parameters. Instead, we show that switching to an image-based rendering pipeline gives much better results, as it allows a good balance between what to store implicitly (the learned part) and what to store explicitly (the training views). This results in better transfer and good results when combined with a ray-based replay scheme. This, for the first time, makes it possible to handle dynamic scenes of 1000+ frames with low storage requirements and good quality.

4.20 The "Stability Gap"

Gido van de Ven (KU Leuven, BE)

Continually learning from a stream of non-stationary data is challenging for deep neural networks. When these networks are trained on something new, they tend to quickly forget what was learned before. In recent years, considerable progress has been made towards overcoming such catastrophic forgetting, predominantly thanks to an approach called "replay". With replay, examples of past tasks are stored in a memory buffer and later revisited when the network is trained on new tasks. Strikingly, even with just a handful of stored samples per task, replay still performs very strongly. Replay seems to work so well that it has even been suggested that forgetting is no longer a major issue in continual learning. A recent discovery of us challenges this (De Lange et al., 2023 ICLR). Surprisingly, we found that replay still suffers from substantial forgetting when starting to learn a new task, but that this forgetting is temporary and followed by a phase of performance recovery. We demonstrate empirically that this phenomenon of transient forgetting – which we call the "stability gap" – is consistently observed with replay, even in relatively simple toy problems.

4.21 Projected Functional Regularization for Continual Learning

Joost van de Weijer (Computer Vision Center – Barcelona, ES)

Recent self-supervised learning methods are able to learn high-quality image representations and are closing the gap with supervised approaches. However, these methods are mostly used as a pre-training phase over IID data. In this talk, we focus on self-supervised methods for continual learning of visual feature representations. I introduce, Projected Functional Regularization (PFR) where a separate temporal projection network prevents forgetting of previously learned representations without jeopardizing plasticity. The main advantage of the new regularization method over existing methods is that it does not penalize the learning of new knowledge, and as a results can reach a better plasticity-stability trade-off.

4.22 Knowledge Accumulation in Continually Learned Representations and the Issue of Feature Forgetting

Eli Verwimp (KU Leuven, BE)

During this presentation, I will present and discuss how continual learners learn and forget representations. We have observed two phenomena: knowledge accumulation, i.e. the improvement of a representation over time, and feature forgetting, i.e. the loss of task-specific representations. To better understand both phenomena, we introduced a new analysis technique called task exclusion comparison. If a model has seen a task and it has not forgotten all the task-specific features, then its representation for that task should be better than that of a model that was trained on similar tasks, but not that exact one. Our experiments show that most task-specific features are quickly forgotten, in contrast to what has been suggested in the past. Further, we demonstrate how some continual learning methods, like replay, and ideas from representation learning affect a continually learned representation.

4.23 Prediction Error-based Classification for Class-Incremental Learning

Michal Zajac (Jagiellonian University – Kraków, PL)

Class-incremental learning (CIL) is a particularly challenging variant of continual learning, where the objective is to discriminate between all classes presented during the incremental learning process. Existing solutions often suffer from excessive forgetting and imbalance of the scores assigned to classes that have not been seen together during training. In our work, we introduce a novel approach, Prediction Error-based Classification (PEC), which differs from traditional discriminative and generative classification paradigms. PEC determines a class score by measuring the prediction error of a model trained to replicate the outputs of a frozen random neural network on data from that class. Our empirical results show that PEC performs strongly and is on par or better than all considered rehearsal-free baselines, including those based on discriminative and generative classification, across multiple CIL benchmarks.

5 Working groups

5.1 Evaluation (Part 1)

Alexander Geppert (Hochschule für Angewandte Wissenschaften Fulda, DE)

Various aspects of evaluation procedures in CL were discussed, such as the proper and improper way of tuning hyper-parameters, the use of simple datasets like MNIST, and what useful evaluation measures for CL could be. It was commonly felt that new evaluation measures should also reflect what CL can contribute in terms for real-world applicability. For example, consistency, speed or compute-time/energy benefits achievable by CL when training large-scale models could be metrics to be used. We raised the issues of CL benefiting data privacy, and the application of CL to other modalities beyond vision. The general difficulty of evaluating models on large-sale data, as well as difficulties with the very concept of dataset were raised.

5.2 Evaluation (Part 2)

Andrea Cossu (University of Pisa, IT)

State of the art is useful provided that we study hard problems where it is “hard to cheat”. In particular, in continual learning the state of the art should be associated to a precisely specified setup. This is also due to the fact that, in continual learning, it is especially easy to cheat. Toy problems like MNIST can be useful, although some phenomena may only be visible at a certain scale. Surely, MNIST-like problems are useful as sanity checks before proceeding with more complex benchmarks. MNIST may still be relevant in extreme setups (e.g., online, replay-free, single-class learning). Continual learning is sometimes modality-specific. This is especially true for computer vision, where heavy use of augmentations restricts the applicability of continual learning strategies to other modalities.

5.3 Reproducibility

Alexander Geppert (Hochschule für Angewandte Wissenschaften Fulda, DE)

The session discussed how reproducibility in CL could be improved by, e.g., organizing a special track at a conference. The general goals of such an undertaking, as well as the target population of potential authors were discussed, as well as questions about what papers submitted to such a track could discuss. It was agreed that, despite a focus on reproducing results, papers should contain newness realized by, e.g., supplementary experiments, extended hyper-parameter searches or an application to other datasets. Finally, issues concerning the workflow of the submission and the review process were discussed.

5.4 Online Learning and Continual Learning

Andrea Cossu (University of Pisa, IT)

In online learning, there is no notion of generalization. Instead, algorithms are evaluated on regret. Online learning algorithms make a decision in each step (or datapoint). While online learning only cares about what happens at the current moment, continua llearning cares about what happened during model lifetime. More, continua learning mainly works with neural networks. As such, it usually requires lots of data, making it difficult to relearn something. This requires to mitigate forgetting . Online learning, instead, does not have this requirement because relearning happens quickly. Can we come up with real data streams that have natural distribution? For example, data from Twitter can provide hundreds or thousands of datapoints per second, with gradual drift. Unfortunately, the Twitter API does not allow to extract this data anymore. One other difference is that, continual learning with replay always considers that distribution for a certain task remains stationary (the input-output mapping does not really change). It is still unclear whether or not continual learning and online learning can be integrated together.

5.5 Optimization in continual learning

Vincenzo Lomonaco (University of Pisa, IT)

The group discussed whether or not environments with piecewise iid data and environments with constant drift are really different for the optimization process. One possible solution would include the approximation of the static setting (only a patch) to make SGD work, for example by approximating a global static target function or by performing local optimization related to a moving target function. The usage of constraint optimization processes may help in maintaining important properties. A completely different solution would depart from the usual end-to-end training by leveraging separate objectives for different tasks and representations.

The group also discussed the role of bias for optimization in biology. It could be important to put similar bias into the model (like memory consolidation). In this sense, local learning is not similar to back-propagation which is global.

Participants

- Rahaf Aljundi
Toyota Motor Europe – Zaventem, BE
- Shai Ben-David
University of Waterloo, CA
- Matthias Bethge
Universität Tübingen, DE
- Andrea Cossu
University of Pisa, IT
- Fabian Fumagalli
Universität Bielefeld, DE
- Joao Gama
INESC TEC – Porto, PT
- Alexander Geppert
Hochschule für Angewandte Wissenschaften Fulda, DE
- Tyler Hayes
NAVER Labs Europe – Meylan, FR
- Paul Hofman
LMU München, DE
- Eyke Hüllermeier
LMU München, DE
- Christopher Kanan
University of Rochester, US
- Tatsuya Konishi
KDDI – Saitama, JP
- Dhireesha Kudithipudi
University of Texas – San Antonio, US
- Christoph H. Lampert
IST Austria – Klosterneuburg, AT
- Bing Liu
University of Illinois – Chicago, US
- Vincenzo Lomonaco
University of Pisa, IT
- Martin Mundt
TU Darmstadt, DE
- Razvan Pascanu
DeepMind – London, GB
- Adrian Popescu
CEA LIST – Nano-INNOV, FR
- James M. Rehg
Georgia Institute of Technology – Atlanta, US
- Andreas Tolias
Baylor College of Medicine – Houston, US
- Tinne Tuytelaars
KU Leuven, BE
- Gido van de Ven
KU Leuven, BE
- Joost van de Weijer
Computer Vision Center – Barcelona, ES
- Eli Verwimp
KU Leuven, BE
- Michal Zajac
Jagiellonian University – Kraków, PL

Report from Dagstuhl Seminar 23131

Software Bug Detection: Challenges and Synergies

Marcel Böhme*[1], Maria Christakis*[2], Rohan Padhye*[3], Kostya Serebryany*[4], Andreas Zeller*[5], and Hasan Ferit Eniser†[6]

1 **MPI-SP – Bochum, DE & Monash University – Melbourne, AU.** `marcel.boehme@mpi-sp.org`
2 **TU Wien, AT.** `maria.christakis@tuwien.ac.at`
3 **Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US.** `rohanpadhye@cmu.edu`
4 **Google – Mountain View, US.** `kcc@google.com`
5 **CISPA – Saarbrücken, DE.** `zeller@cispa.saarland`
6 **MPI-SWS – Kaiserslautern, DE.** `hfeniser@mpi-sws.org`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 23131 "Software Bug Detection: Challenges and Synergies". This seminar brought together researchers from academia and industry working on various aspects of software bug detection, with two broad goals: identifying challenges in practical deployment of bug-finding tools and discovering new synergies among bug-finding techniques and research methods. The seminar focused discussion on bug-finding tools and their relevance and adoption in industry.

Seminar March 26–31, 2023 – https://www.dagstuhl.de/23131

2012 ACM Subject Classification Software and its engineering; Software and its engineering → Software testing and debugging; Software and its engineering → Software verification

Keywords and phrases Bug Finding, Coverage, ML4SE, Oracles, Software Testing, Software Verification

Digital Object Identifier 10.4230/DagRep.13.3.92

1 Executive Summary

Rohan Padhye (Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US)
Marcel Böhme (MPI-SP – Bochum, DE & Monash University – Melbourne, AU)
Maria Christakis (TU Wien, AT)
Kostya Serebyany (Google – Mountain View, US)
Andreas Zeller (CISPA – Saarbrücken, DE)

Software bugs are inevitable when engineering complex systems, and the cost of their consequences can be enormous. Over the past several decades, there has been tremendous progress in advancing the state-of-the-art in automatic bug finding. Popular techniques include static analysis, dynamic analysis, formal methods and specification, verification, symbolic execution, fuzzing, and search-based test generation. However, with the rapid growth of new application domains and the ever-increasing complexity of software, practitioners are rarely faced with a one-size-fits-all solution for finding bugs in their software. Domain-specific

* Editor / Organizer
† Editorial Assistant / Collector

trade-offs must be made in choosing the right technique, in configuring a tool to work for a particular context, or in combining multiple approaches to provide better assurances. Currently, this is largely a manual activity and the burden is mainly on practitioners.

This Dagstuhl Seminar brought together researchers from academia and industry working on various aspects of software bug detection, with two broad goals: identifying challenges in practical deployment of bug-finding tools and discovering new synergies among bug-finding techniques and research methods.

The seminar focused discussion on bug-finding tools and their relevance and adoption in industry. Other questions that came up included: What are effective approaches to discover software bugs as fast as possible? How can we formally verify the absence of bugs? Which guarantees do our approaches provide about the correctness, reliability, and security of the software when no bugs are discovered? Which concerns do practitioners have when bug finding tools are integrated into their development process? What are effective approaches to automatically mitigate, diagnose, or repair certain kinds of bugs?

The seminar was organized to maximize time for open discussion. Seven attendees were invited to give short keynote talks of a topic of their choice, which occurred on mornings of the seminar. The afternoons were reserved for working groups and panel discussions. The topics for these discussions were crowdsourced using an ad-hoc voting system in the main seminar room. Working groups then broke out for discussion in smaller rooms and reconvened with summaries.

Overall, in the opinion of the organizers, the seminar was a huge success. The strong participation from researchers in industry and the diverse set of expertise among researchers in academia enabled open-minded discussion on topics of key importance that are not easily exchanged via traditional conference proceedings. This document summarizes the talks given and working groups conducted in the seminar.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 Reflections on Software Testing Research

Cristian Cadar (Imperial College London, GB)

The talk covered a number of challenges and opportunities for software testing research, including understanding developer communities, benchmarking, incremental progress and research directions, the need for patch testing techniques, the problems on which academia should focus on, the challenges of maintaining tools in academia, and the many misaligned incentives that researchers in the area are facing.

3.2 Beyond the Crash Oracle: Challenges in Deploying Fuzzing to Find Functional Errors at Scale

Alastair F. Donaldson (Imperial College London, GB)

Main reference Alastair F. Donaldson, Hugues Evrard, Paul Thomson: "Putting Randomized Compiler Testing into Production (Experience Report)", in Proc. of the 34th European Conference on Object-Oriented Programming, ECOOP 2020, November 15-17, 2020, Berlin, Germany (Virtual Conference), LIPIcs, Vol. 166, pp. 22:1–22:29, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.
URL https://doi.org//10.4230/LIPIcs.ECOOP.2020.22

Over the last decade, fuzzing has been shown to be extremely effective at finding security critical defects in software and is now deployed at scale by many large companies and open-source projects. However, most of the success of fuzzing at scale is restricted to finding inputs that cause the system under test to crash. While testing with respect to the "crash oracle" is important for finding vulnerabilities (especially when the crash oracle is boosted with compile-time instrumentation to detect undefined behaviour), its effectiveness for finding deep functional errors is limited – errors that lead to the system under test doing the wrong thing – but doing so without actually crashing.

The talk covered why designing and scaling up fuzzing techniques for finding functional errors is so much harder than in the context of crashes. Issues include the manual effort required to build smart input generators and mutators, the need to respect input validity constraints not only during generation/mutation, but also during test-case reduction, and the difficulty associated with de-duplicating bug-triggering test cases. Throughout the talk, the speaker drew on his experience designing GraphicsFuzz, a metamorphic fuzzing technique for GPU compilers on which he based a start-up company that was acquired by Google and subsequently used to fuzz Android graphics drivers to find functional bugs.

3.3 Lessons Learned from Designing Software Engineering Methods for Enabling AI

Miryung Kim (UCLA, US)

Software developers are rapidly adopting AI to power their applications. Current software engineering techniques do not provide the same benefits to this new class of compute and data-intensive applications. To provide productivity gains that developers desire, our research group has designed a new wave of software engineering methods.

First, the speaker discussed technical challenges of making custom hardware accelerators accessible to software developers. She showcased HeteroGen, an automated program repair and test input generation method for making heterogeneous application development with FPGA accessible to software developers. Second, she discussed technical challenge of designing automated testing and debugging methods for big data analytics. She also showcased BigTest, symbolic-execution based test generation for Apache Spark.

At the end, the speaker shared the lessons learned from designing SE methods that target big data and HW heterogeneity and discuss open problems in this data and compute-intensive domain.

3.4 Dependencies everywhere!

Anders Møller (Aarhus University, DK)

Modern software critically relies on reusable, open source software packages. They enable fast development of advanced applications, but also introduce major challenges with incompatibilities, security vulnerabilities, and breaking changes. In this talk, the speaker described ongoing work at coana.tech on building new program analysis tools that can assist library and application developers by providing accurate information about how dependencies are being used.

3.5 Hits and Misses From a Decade of Program Analysis in Industry

Peter O'Hearn (University College London, GB)

The speaker talked about hits and misses from a decade of program analysis in industry.

3.6 Daunting and Exciting Reality of Industrial Bug Hunting

Dmitrii Viukov (Google – München, DE)

In this talk, the speaker shared his experience deploying various bug detection tools and fuzzing at Google over the last decade. The talk touched on the goals, context, constraints and everyday life of the team. Then the talk proceeded to the common properties of the tools that found (and not found) adoption and concludes with a look into the future and what types of tools we are looking for.

3.7 From Bug Detection to Bug Mitigation and Elimination: the Role of Tools in Memory Safety

Anna Zaks (Apple Computer Inc. – Sunnyvale, US)

A key ingredient to securing a platform is mitigation and elimination of memory safety errors in software. Dynamic and static bug-finding tools can be used to find many memory-safety bugs such as buffer overflows, use-after-frees, and data races. However, most such tools assume no change in the environment – the compiler, the language, libraries, operating system and the hardware – which limits their impact. In this talk, the speaker described how to co-design language security features with the rest of the software stack and how that helps design tools and techniques for mitigation and elimination of whole classes of memory-safety bugs. She also talked about the new role bug-finding techniques like program analysis can play in a world where safer languages, libraries, and security mitigations are available.

4 Working groups

4.1 Oracles 3

Eric Bodden (Universität Paderborn, DE)

In this working group, attendees discussed oracles in different domains and ways to generalize them. For instance, one widely considered oracle type is memory corruption because it is very generic, easy to specify and detect (particularly if they lead to a crash). However, we also need better oracles in other domains such as in API testing, big data analytics and so on. One can utilize error handling code in APIs as oracles and test them using directed fuzzing techniques. In big data analytics, the available options to tackle the oracle problem are metamorphic and differential testing which come with their own limitations. A high-level idea was to define simpler proxy properties, whose violations indicate a problem whereas conformances are not very interesting, in domains where specifying complete oracles is hard. As a result, discussions converged towards the conclusion of benefiting the most suitable oracle option per domain such as heuristics, proxy properties, error handling code, metamorphic or differential testing.

4.2 Oracles 1

Hasan Ferit Eniser (MPI-SWS – Kaiserslautern, DE)

Metamorphic and differential testing techniques are examples of the most widely applied techniques to overcome the well-known oracle problem. However, manual effort for crafting metamorphic operations, little or no (e.g. coverage) feedback, and the non-existence of multiple subjects under test limit applicability of these techniques in practice. For this reason, "assertion" based oracles still play an important role in bug detection. Nonetheless, they also come with their own limitations. For example, hyperproperties are hard to express with assertions. Using ML to determine successful and failing executions or human-in-the-loop based solutions can also be considered for this problem. One example usage of ML can be to rank rules that serve as oracles for a particular program. Other open problems in this area include oracles for patch generation and dealing with bugs in specifications.

4.3 Severity Analysis

Caroline Lemieux (University of British Columbia – Vancouver, CA)

In a world of finite developer resources, we cannot prioritize the investigation and fixing of all potential bugs. The goal of severity analysis is to rank bugs by severity, so as to direct developer time to those bugs whose fix is most likely to improve the software system.

We considered two perspectives on severity. The first is severity from a security perspective. We discussed a few factors that should go into judging the severity of a bug from a security perspective. The second one is exploitability (currently, this is manually done by human analysts when a bug is e.g. submitted to MITRE for ranking as a CVE) Higher exploitability implies higher severity. And the last one is ease of discoverability where we assume easier to discover usually means higher severity, as bad actors may be more likely to discover it.

The perspective of functional severity is slightly different. When considering the severity of a functional (i.e., not likely to have security impacts, but which may crash or give a wrong result) bug, we may care more about user experience for example frequency of bug-revealing inputs in the usual input distribution. In this case, the more commonly seen in practice implies more severe bugs.

Additionally in both cases, how many different paths there are to reveal bug may relate to severity: higher number of potentially reaching paths means more severe.

4.4 Competitions and Evaluations

Rohan Padhye (Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US)

Joint work of Rohan Padhye, Dirk Beyer

This working group described the importance of competitions and standardized evaluation methodologies in supporting research on automated bug-finding. This approach has been very successful in accelerating research in domains such as SMT solvers. The discussion first addressed the goals such an evaluation, which can be to find as many bugs as possible in as little time as possible, as well as the use of other metrics such as code coverage. Care should also be taken to distinguish between general-purpose bug finding tools and customized tools that search for special classes of bugs. The group then identified several efforts in this space, such as the DARPA Cyber Grand Challenge (CGC) [1] dataset which contains 50 programs each having one bug, the LAVA-M [2] dataset of artificial bugs, the MAGMA [3] data-set containing historical bugs, and the FuzzBench [4] data-set containing known bugs in real programs that fuzzers can find. Some concerns with using such data sets include representativeness of bugs, whether the bugs in artificial datasets are realistic and likely to occur in practice, whether tools might overfit to these benchmarks, the computational effort required to find small numbers of bugs in large programs, and various constrained imposed by competitions such as a strict time limit. The group agreed that finding new bugs in real target is that are previously well-tested is a good way of demonstrating a technique's effectiveness, but it is not always possible to find such results. Formats such as registered reports, which allow papers to be peer-reviewed before evaluation results are available, appear to mitigate some of the risks with expecting authors to find such previously unknown bugs with new tools even if the technique is novel and sound.

References

1 Lee, N. (2015). Darp's cyber grand challenge (2014–2016). Counterterrorism and Cybersecurity: Total Information Awareness, 429-456.

2 Dolan-Gavitt, B., Hulin, P., Kirda, E., Leek, T., Mambretti, A., Robertson, W., ... & Whelan, R. (2016, May). Lava: Large-scale automated vulnerability addition. In 2016 IEEE symposium on security and privacy (SP) (pp. 110-121). IEEE.

3 Hazimeh, A., Herrera, A., & Payer, M. (2020). Magma: A ground-truth fuzzing benchmark. Proceedings of the ACM on Measurement and Analysis of Computing Systems, 4(3), 1-29.

4 Metzman, J., Szekeres, L., Simon, L., Sprabery, R., & Arya, A. (2021, August). Fuzzbench: an open fuzzer benchmarking platform and service. In Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering (pp. 1393-1403).

4.5 Correctness and Verification

Rohan Padhye (Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US)

Joint work of Dirk Beyer

This working group discussed the importance of correctness guarantees in automated program analysis tools used for improving software quality. The properties to check could include functional correctness or things like reachability. Many static techniques need to use some form of underapproximation or overapproximation for scalability. Dynamic techniques like testing or fuzzing do not cover all paths and so provide no guarantees, but continuous integration is important. There is a scope to include some form of verification in CI. Deductive verification techniques can scale using composition, but require annotations from developers. Proof-of-concept projects from the operating systems domain provide some hope. Incremental verification is also a promising direction. The group discussed directions that researchers can investigate to increase the adoption and applicability of verification techniques. A theme that emerged is that verification must be thought not only as correctness, but as something to integrate into processes, so we need ways to integrate techniques and run them in dev processes. The verification process is also where you find out what you want to verify, during the process!

4.6 Reproducibility and Artifacts

Rohan Padhye (Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US)

Joint work of Dirk Beyer

The working group discussed the current state of artifact evaluation in ACM-sponsored conferences along and the relevance of this process. The group first identified what the various badges mean. The "Available" badge indicates that the artifact must have a DOI, must be identifiable, immutable, and long-term available. If an artifact is evaluated by a committee, it can be given a badge of "Functional" or "Reusable" based on criteria set out by the ACM and the artifact evaluation chairs. The artifact can also be considered "Validated" if the results of the accompanying research paper are independently "reproduced" (using author-provided artifacts) or "replicated" (by re-implementing the artifact using the information available in the paper). There seems to be considerable discrepancy between conferences and ACM SIGs on how to interpret these badges. For example, SIGMOD provides artifact "reproduced" badge even when the results are validated by an artifact evaluation committee, which in SIGSOFT conferences is just considered "functional" (reserving the "reproduced" badge for independent studies by future authors). The group agreed that more consistency is needed in issuing these badges. The group also discussed whether the effort of artifact evaluation is justifiable and whether there is any incentive for authors to produce high quality artifacts. Another question that came up was whether future authors who reuse the artifact should cite the artifact DOI or the original paper. The group identified some benefits for the artifact process, such as the fact that it allows students to serve on review committees and experience working with artifacts authored by their peers. Reproducibility and Artifacts

4.7 Oracles 2

Mathias Payer (EPFL – Lausanne, CH)

In the bug detection oracles work group we focused on ways to signal that a bug was triggered. Oracles implement a given policy and enforce this policy by instrumenting the target program with certain checks. These checks then continuously monitor if the policy has been violated.

We started by discussing sanitizers in general and focusing in particular on AddressSanitizer, UndefinedBehaviorSanitizer and ThreadSanitizer. These are the most well-known sanitizers and thoroughly used. In the discussion, they served as the basis to define Oracle policies and we continued towards program specifications.

If you want to go beyond these general sanitizer policies, you will require some specification that defines program behavior. Generalizing this is somewhat challenging and requires additional work from the developer. This can quickly become domain specific, e.g., protecting SQLlite against SQL-injection, scripts against command injection, or websites against cross-site scripting. At a higher level, data-flow may serve as a general policy but would require massive amounts of performance improvements.

Among others we also discussed possible hardware extensions and how to generalize anomaly detection for these different niches.

4.8 User Experience in Fuzzing

Van-Thuan Pham (The University of Melbourne, AU)

In this group, we discussed user experience in three stages of fuzzing: (i) set up fuzzing environments, (ii) run & monitor fuzzing progress, and (iii) analyze the results.

In the first stage, writing test harnesses/fuzz drivers seem to be the most time-consuming task and it could be challenging for developers. It is one of the reasons why the number of fuzz drivers in the open-source software packages is quite limited. Some solutions have been discussed including transforming unit tests to fuzz drivers (e.g., FuzzTest) or generating fuzz drivers in a fully automated manner (e.g., FUDGE, FuzzGen). Some companies like Google have bug bounty programs to reward researchers who contribute their fuzz drivers.

Writing fuzz drivers manually is time-consuming and error prone in the sense that developers might have difficulties in understanding if their fuzz drivers are working properly or not. So, in the second stage, companies like Meta/Facebook have built "health-check" mechanism to measure run-time metrics (e.g., code coverage improvement trajectory) and then flag potential malfunctional fuzz drivers. Moreover, Indeterminism is a challenge in some fuzz targets.

In the third stage, users expect better tools to support triaging the crashes and more detailed report. For instance, they would like to have better fault localization and root cause analysis utilities. They also expect to know about the severity of the identified bugs. They would also like to get suggested patches to fix the bugs. Another question developers would ask is when they should stop fuzzing. For ClusterFuzzLite setup at Google, they use a threshold of 10 mins while in fuzzing research papers, results are normally reported for 24-hour experiments.

4.9 Large Language Models for Bug Detection

Michael Pradel (Universität Stuttgart, DE)

The impressive power of large language models (LLMs) has lead to the question how to use these models for bug detection. We discussed two main directions. First, LLMs could generate inputs for testing programs, in a way similar to fuzzers or automated test generator. A strength of LLMs for this task could be to generate realistic inputs, as the model learns typicaly distributions of inputs from data. A challenge for this idea is how to obtain uncommon inputs from a model trained to predict likely token sequences. Second, LLMs could act in a way similar to static analysis and predict for given piece of code whether the code contains a bug, and ideally, more details on this bug. Initial experiments reported by the participants suggest that both directions are promising and worth exploring.

4.10 Learning-based and Analysis-based Bug Detection

Michael Pradel (Universität Stuttgart, DE)

Machine learning-based techniques for finding bugs and vulnerabilities are showing promising results. At the same time, traditional static and dynamic analysis approaches come with their own benefits. We discussed how to combine the strengths of learning-based and analysis-based bug detection techniques. One interesting direction is to use a trained model as a filter or ranking mechanism applied to warnings reported by a static analysis. For example, such a model could be trained on past warnings and records of how developers reacted to them. Another interesting direction is to feed information an analysis extracts from programs, e.g., aliasing relationships, as an input into a machine learning model. Finally, we discussed ways for more tightly integrating both kinds of approaches, e.g., by having a model query an analysis, or vice versa.

4.11 New Coverage Signals

Kostya Serebryany (Google – Mountain View, US) and Hasan Ferit Eniser (MPI-SWS – Kaiserslautern, DE)

In this working group, we discussed new ways of guiding feedback in the form coverage signals in fuzzers. Custom feedback signals for domain specific problems are almost always helpful to increase bug finding performance. Neuron coverage, which is a coverage metric devised for neural networks is a typical example. However, the challenge here is how to generalize the better coverage signal. For this, we need to think about reasonably abstract state that generalize for a good coverage signal (e.g. depth of stack to guide fuzzer to stack over flows).

4.12 ML and Static Analysis

Dominic Steinhöfel (CISPA – Saarbrücken, DE)

In this working group, we discussed how AI techniques could help in automated software testing. We believe AI-based techniques are too slow to be used as standalone input generators. Yet, they might be useful as an alternative to symbolic execution to solve "fuzzing blockers," i.e., to complement the usual graybox techniques. Similarly, they could help to reduce the overhead in fuzzer harness generation (e.g., for libfuzzer). Orthogonally to these considerations, we think that current advances in AI promise solutions to automatic bug explaining/root causing and other debugging problems such as coming up with an input reaching a chosen line of code. Concerning root causing, we discovered the "ChatDBG" project, which uses GPT to explain errors in interactive debuggers. We investigated this project and were astonished by how trivial the prompt used for GPT was. In conclusion, we think there is much potential in using AI to augment traditional automated testing techniques and address open issues in automated testing and debugging.

4.13 Dependencies

Andreas Zeller (CISPA – Saarbrücken, DE)

Generally speaking, Modularity is a big success of Software Engineering, as it enables the widespread reuse of components we see today. Unfortunately, being dependent on third-party modules also creates new problems, as we need to trust them not to introduce new bugs and vulnerabilities. Solutions discussed included (1) quarantining less trusted modules into sandboxes with least privileges to reduce the risk of vulnerabilities; and (2) being explicit about dependencies for both clients and providers of modules to reduce the risk of breaking compatibility.

Participants

- Cornelius Aschermann
Meta – Seattle, US
- Sébastien Bardin
CEA LIST – L'Hay les Roses, FR
- Lukas Bernhard
CISPA – Saarbrücken, DE
- Dirk Beyer
LMU München, DE
- Eric Bodden
Universität Paderborn, DE
- Marcel Böhme
MPI-SP – Bochum, DE & Monash University – Melbourne, AU
- Herbert Bos
VU University Amsterdam, NL
- Cristian Cadar
Imperial College London, GB
- Sang Kil Cha
KAIST – Daejeon, KR
- Maria Christakis
TU Wien, AT
- Jürgen Cito
TU Wien, AT
- Alastair F. Donaldson
Imperial College London, GB
- Hasan Ferit Eniser
MPI-SWS – Kaiserslautern, DE
- Rahul Gopinath
The University of Sydney, AU
- Alessandra Gorla
IMDEA Software Institute – Madrid, ES
- Reiner Hähnle
TU Darmstadt, DE
- Marc Heuse
marc heuse it security – Berlin, DE
- Christian Holler
Mozilla – Berlin, DE
- Miryung Kim
UC – Los Angeles, US
- Caroline Lemieux
University of British Columbia – Vancouver, CA
- Jonathan Metzman
Google – New York, US
- Peter Müller
ETH Zürich, CH
- Anders Møller
Aarhus University, DK
- Yannic Noller
National University of Singapore, SG
- Peter O'Hearn
University College London, GB
- Hakjoo Oh
Korea University – Seoul, KR
- Alessandro Orso
Georgia Institute of Technology – Atlanta, US
- Rohan Padhye
Carnegie Mellon University – Pittsburgh, US & Amazon Web Services, US
- Mathias Payer
EPFL – Lausanne, CH
- Van-Thuan Pham
The University of Melbourne, AU
- Michael Pradel
Universität Stuttgart, DE
- Manuel Rigger
National University of Singapore, SG
- Kostya Serebryany
Google – Mountain View, US
- Dominic Steinhöfel
CISPA – Saarbrücken, DE
- Dmitrii Viukov
Google – München, DE
- Valentin Wüstholz
ConsenSys – Wien, AT
- Anna Zaks
Apple Computer Inc. – Sunnyvale, US
- Andreas Zeller
CISPA – Saarbrücken, DE
- Lingming Zhang
University of Illinois – Urbana-Champaign, US

Report from Dagstuhl Seminar 23132

AI-Augmented Facilities: Bridging Experiment and Simulation with ML

Peer-Timo Bremer[*1], **Brian Spears**[*2], **Tom Gibbs**[*3], **and Michael Bussmann**[*4]

1 **Lawrence Livermore National Laboratory, US.** bremer5@llnl.gov
2 **Lawrence Livermore National Laboratory, US.** spears9@llnl.gov
3 **Nvidia – Santa Clara, US.** tgibbs@nvidia.com
4 **Helmholtz-Zentrum Dresden-Rossendorf, DE.** m.bussmann@hzdr.de

Abstract

In the last week of March 2023, Schloss Dagstuhl hosted a Dagstuhl Seminar on "AI-Augmented Facilities: Bridging Experiment and Simulation with ML". The seminar brought together experimental and computational scientists, experts on edge and HPC computing, and machine learning and computer science researchers to jointly develop a strategic vision on how to move towards AI-augmented facilities in a unified manner. The goal was to suggest a common research agenda with an emphasis on areas where joint efforts are needed for future progress. Starting with some overarching perspectives the seminar was dominated by lively discussions that resulted in a strategic write-up to be published separately.

Seminar March 26–31, 2023 – https://www.dagstuhl.de/23132
2012 ACM Subject Classification Applied computing → Physical sciences and engineering; Computing methodologies → Machine learning; Computer systems organization → Embedded and cyber-physical systems; Mathematics of computing → Probability and statistics; Information systems → Data management systems
Keywords and phrases Self-driving experiments, Smart facilities, AI models, Multimodal data, Accelerated scientific discovery, AI software stack for experiments
Digital Object Identifier 10.4230/DagRep.13.3.106

1 Executive Summary

Peer-Timo Bremer (Lawrence Livermore National Laboratory, US)
Brian Spears (Lawrence Livermore National Laboratory, US)
Tom Gibbs (Nvidia – Santa Clara, US)
Michael Bussmann (Helmholtz-Zentrum Dresden-Rossendorf, DE)

The Dagstuhl Seminar connected three traditionally different communities: experimental and computational scientists, experts in HPC and edge computing, and machine learning researchers, to discuss a new vision for future AI-augmented facilities. This document summarizes the activities during the week of in person discussion including the outline of a position paper that is under development to publish the joined findings. The seminar proceeded in roughly three stages: an introduction with two keynotes and a general discussion

[*] Editor / Organizer

AI-Augmented Facilities: Bridging Experiment and Simulation with ML, *Dagstuhl Reports*, Vol. 13, Issue 3, pp. 106–131
Editors: Peer-Timo Bremer, Brian Spears, Tom Gibbs, and Michael Bussmann
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

on the goals, an expansive phase of collecting ideas and defining the scope of the position paper, and finally working groups on creating explicit outlines and collecting materials for various sections of the paper.

2 Table of Contents

3 Keynotes and Topic Introduction

The week started with a session to introduce all participants with a little bit of their background to facilitate later discussions and provide an overview of the available expertise. This was followed by two introductory keynotes from the organizers briefly describing the current state of affairs in AI-augmented facilities from both the US (Brian Spears) and the EU (Michael Bussmann) perspective which consumed Monday morning.

3.1 Facilities & AI a US Perspective

Brian Spears (Lawrence Livermore National Laboratory, US)

The pace of data generation in modern science has greatly accelerated, but the pace of transformational discovery is still too slow. This is clear at a variety of state-of-the-art facilities: laser experiments are slow or noisy; advanced manufacturing (AM) is open loop; accelerators need time consuming tuning, sometimes by hand. However, AI-enable self-driving systems can accelerate our science and discovery processes. AI sentinels that help collect data, compare to prediction, and choose next steps can provide a step change in experimental and manufacturing operations. They will bring accelerated closed-loop operations, transformational data rates, physics-informed experiment updates on sub-second timescales, and digital twins for facility modeling and optimization. This not only accelerates discovery, but it deepens the quality of knowledge that we can discover. As examples, it will provide stabilized lasers and autonomous optimization of high-energy-density physics, self-correcting AM processes and high-throughput operations, and repeatable, robust accelerator conditions. Beyond individual systems, self-driving ecosystems composed of interconnected sets of these facilities will offer capabilities greater than the sum of their parts offering rapid discoveries that are hard to conceive in today's slower and isolated science regime. To achieve this goal, the science community needs to work together to build the scientific tools to execute self-driving operations. With community input, like that provided by Dagstuhl, we can make self-driving science systems a reality.

3.2 Facilities & AI an EU Perspective

Michael Bussmann (Helmholtz-Zentrum Dresden-Rossendorf, DE)

In the EU, the use of AI at large-scale research infrastructures is on the rise in a broad variety of fields from Particle Physics to Photon Science, Neutron Science, Life Science, Astrophysics to Laser Science and more. The ESFRI Roadmap and the EU digitalization strategy highlight the importance of data and meta data and the potential of AI. Focusing on the example of Germany, We highlight how the Helmholtz Association as the largest research organization in Europe and in particular the Helmholtz Research Field Matter plan to develop autonomous, intelligent facilities using AI at key points in the data lifecycle of research facilities. The topic Data Management & Analysis and the Helmholtz Incubator for Information and Data Science play key roles, looking at such diverse topics as data lifecycle management, the tight integration of simulation, experiments and machines, online and large-scale data analysis, visual analytics, optimization, automation and resilience. Embedded in a national AI strategy with key components such as the National Research Data Infrastructure, ErUM-Data and many more, embedded in EU-wide and international cooperations and initiatives, the landscape of AI-augmented facilities is being shaped into a EU-wide, cross-community effort to enable excellent science at optimum conditions across the whole spectrum of research infrastructures. We argue that this can be a blueprint for international collaboration on AI-augmented facilities.

4 Working Groups Results

Starting Monday afternoon the seminar switched to a mixture of interactive working groups followed by sessions to report the results and plan the next agenda items. The sections below will briefly summarize the individual sessions including (subsets of) the raw notes when possible and list or participants where available.

4.1 Monday Afternoon

In two closely related sessions on Monday afternoon all seminar participants first collected a list of prototypical science drivers that motivate the need for AI-augmented facilities. Subsequently, this discussion branched out to list stakeholders and specific applications that could be used later as examples. Finally, the discussion converged on defining more the goal of the seminar more explicitly: To collect the insights, existing solutions, and strategic ideas into a perspective paper to be jointly published. Consequently, the remainder of Monday afternoon was spend creating a first paper outline that simultaneously served as a guide for the schedule on Tuesday morning. An overview of the topics discussed is provided in Figure 1 in form of a topic graph.

4.2 Tuesday Morning

Following the initial paper outline different groups in parallel started to flesh out individual sections. This started with three parallel breakouts on the overall needs, the approach, and the expected outcomes as the cornerstone of the paper.

4.2.1 Group 1: Needs

Working Group 1 was tasked to explore the need for AI augmented facilities in more detail to ultimately serve as the motivation for the perspective. Figure 2 documents some of the notes including a conceptual diagram of the state-of-the-art created as straw-man for the discussion. The list of high level needs collected during the outbrief included:

- Faster Science
- Better Science / per $ or €
- Data Interpretation / optimum operation
- Optimised sciences
- Addressing grand challenges

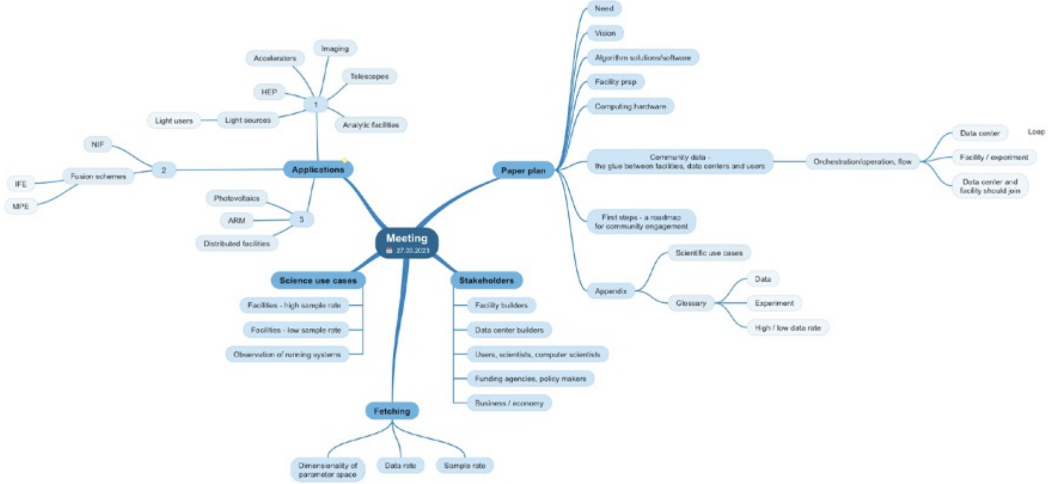


Figure 1 Topic graph of the discussion on Monday afternoon outlining both the science drivers, stakeholders and applications as well as a plan for a perspective paper as the goal for the seminar.

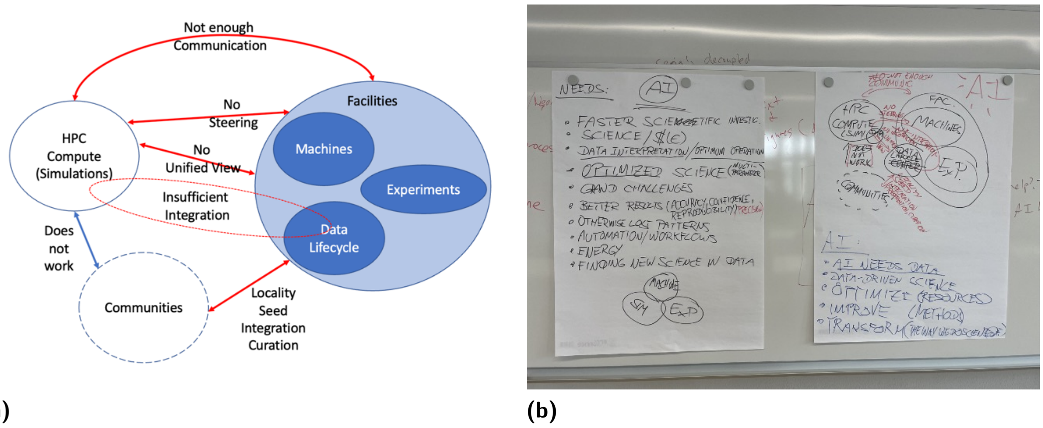
- More accurate, precise and reproducible results
- Finding complex patterns in big data
- Automated workflows for scientific facilities
- Optimise energy efficiency of large facilities
- Finding new and unexpected science in data

4.2.2 Group 2: Approach

The discussion around which approaches might be fruitful to pursue resulted in a set of high level questions that would need to be answered followed by directions for solutions.

Questions:
1. How will we control complex, serial, and decoupled science experiments/observations ?
2. What high-level (abstract) approaches will we use to be more responsive to (said) experiments/observations?
3. How will we respond to grand challenges?
4. What expertise (existing or to be developed) do we need?
5. How do we enable scientists to shorten the "time to science"?

Answers:
1. Match simulation availability to experimental demands → surrogate models to bring physics into the control loop
2. Use inductive AI methods to merge multimodal and heterogeneous data → Autoencoders, deep neural networks, reinforcement learning, etc.
3. Build efficient connections between computing and experiments → couple AI, optimization, ...
4. Engineer and demonstrate robustness and uncertainty quantification for application on real machines and systems → AI techniques for UQ, robustness, to increase machine uptime, etc.
5. Offload experimental data to computing resources
6. Decide next steps based on both computational and experimental knowledge



- FASTER SCIENTIFIC INVESTIG.
- SCIENCE/\$(€)
- DATA INTERPRETATION/OPTIMUM OPERATION
- OPTIMIZED SCIENCE (MULTI-PARAMETER)
- GRAND CHALLENGES
- BETTER RESULTS (ACCURACY, CONFIDENCE, REPRODUCIBILITY, PRECISION)
- OTHERWISE LOST PATTERNS
- AUTOMATION/WORKFLOWS
- ENERGY
- FINDING NEW SCIENCE IN DATA

- AI NEEDS DATA
- DATA-DRIVEN SCIENCE
- OPTIMIZE (RESOURCES)
- IMPROVE (METHODS)
- TRANSFORM (THE WAY WE DO SCIENCE)

Figure 2 (a) Straw-man diagram of the current state of the art and its challenges; (b) Notes of the Needs working group.

4.2.3 Group 3: Envisioned Outcomes

To better understand what success might look like the group members decided to organize the discussion into three time frames: near-term (1-3 years), mid-term (5 years), and long term (10 years) goals.

1-3 Years

- Define Requirements for
  - Computing needs
  - Experimental needs
- Interactive (ML) compute
  - Capability for 5000 Jupyter notebooks
- Edge computing integrated into control system (EPICS++)
- Data and meta-data curation (per domain)
- Connect control to HPC
- New generation of AI-ready control system
- ML enabled simulations
  - Inner loop
  - Outer loop

5 Years

- Self-driving experiment
  - Self-tuning
  - Self-optimizing
  - Archive for stability/reproduction
- Integrated simulations for ML guidance and inference
- Science metrics other than uptime
- Reliable surrogate models updated on the fly
- AI enabled streaming
  - Analysis
  - Anomaly detection
- Distributed experiments
- Robust/ reliable / explainable ML for science

10 Years

- Data standards with translators
- Autonomous collection of data
- Scientist in the loop decision making
- Coupled experiments

4.3 Tuesday Afternoon

Tuesday afternoon developed the paper draft further by exploring the software and hardware needs as well as the necessary changes in large scale facilities. The charges for the different group after the morning discussions were as follows:

- Software needs
  - AI techniques and software tools (classes of AI tools – huge networks, tiny ones, ...)
  - Orchestration for computing and experiment interoperation
  - More ...
- Computing hardware
  - Needs for compute on the instrument
  - Needs for compute at the edge or facility
  - Needs for data center
  - Needs for distributed computing
  - Kinds of architectures, systems
  - More ...
- Facility preparation
  - Making facilities look fully integrated – unified compute and experiment
  - Making facilities AI ready (AI ready diagnostics and instruments)
  - Making facilities prepared for computing
  - Making facilities networked to resources

4.3.1 Group 1: Software Needs

The software working group first collected a list of general capabilities that will be required before discussing in more detail the different dimensions that differentiate various AI approaches.

Necessary capabilities

- AI tools and methods
  - AI benchmarks
  - Tools for knowledge extraction
- Orchestration
  - Online and coupled automation
- ML-Ops and RSE
  - Language of choice: Julia, Python
  - Frameworks to build AI
  - Workflows
  - Notebooks / Interactive development
  - Model / Data parallel
  - Big parallel models
  - Hyperparameter optimisation
  - Model optimisation
- Target users
  - Facilities
  - Scientists / Users
- Automation
- Tools for model calibration and validation
- Tools for continual learning (especially for model drift)

Dimension of AI models

- Time to train
- Time to inference
- Cost and energy
- Size of the model (number of parameters)

  - Data size
  - Compute for training
  - Compute for inference
- Model science performance
  - Accuracy
  - Precision
  - Stability
  - Convergence
  - Fidelity
  - Composability
  - Modularity
- Data
  - Sparsity
  - Multiple modalities
  - Heterogeneity
  - Locality
  - Privacy
- Model capability
  - Multimodal
  - Explainability
  - Deterministic
  - Uncertainty
  - Federated models
  - Dynamic models
- Privacy preserving federated learning as a Service
- Generalisability / Ability to disentanglement (symbolic knowledge / data-driven / representation learning)
- Physics-Informedness
- Procedural knowledge

4.3.2 Group 2: Computing Hardware

The hardware discussion resulted in a list of technology directions that must be considered for AI augmented facilities.

- Object store-type, DB focused, storage
- SWaP (size, weight, and power) based AI-hardware (sensors, embedded systems etc.)
- Chiplet-based (low latency/high bandwidth) embedded-AI accelerator
  - Composability across multiple vendors
- Latency optimized accelerator – multi mode(a)l
- AI hardware for edge training
- Capacity: Large scale, on-demand, aI-training
- Capability system driven AI computing
- Network protocols?!
  - Ethernet

4.3.3 Group 3: Facility Preparation

While full integration and potential distributed “super-facilities” are the long term goal, the facility preparation was discussed in terms of experimental and compute facilities yet with a focus on ensuring inter-facility communication

Experimental Facilities

- AI ready diagnostics / control
  - Digitized
  - Networked vs. online (real-time availability)
  - Compute enabled (ASIC)
  - Sufficient bandwidth
  - (Semi-)autonomous calibration
  - Monitoring and change detection
- Data acquisition system
  - Acquisition
  - Provenance
  - Meta-data
  - System state
- Local compute
  - High precision and AI compute
  - On-site data reduction
  - Large model inference
  - On-site storage
  - Data storage and exploration before transmission
  - Networking
- Software defined systems
- Flexible adjustment of controls (readiness to update)
- Retrain operators to engage with AI

Computing Facilities

- Data transfer / interface with the world
- Networking
- Robotic operating system
- workflow service layer
- Reconfigurable storage
- Hetrogeneous nodes with flexible connection for strong vs. weak scaling
  - High-precision
  - Low-precision
- Make computer center time-responsive
- Data availability and/or streaming dataflow
- Hardware and software policies, i.e., queue priorities

4.4 Wednesday Morning

Wednesday morning focused on some outstanding topics such as communication, first steps, and potential early demonstration targets.

4.4.1 Group 1: Communications and Data Movement

- ESNet
- One platform to file bug in a transparent process
- How to influence Open source packages
  - Resilience
  - Robustness
  - Life-cycle management of the AI software stack
- Storage across sites in a transparent way
  - Access API
  - Lifecycle of data

- High bandwidth vs. low latency
  - What is the priority (depending on the dataflow)
  - Networks are restricted by reality
  - Critical boundary conditions
- Wireless connections
  - Policy issues
  - Security
- Remote control must be viable
- Dealing with asynchronous information

4.4.2 Group 2: First steps

The second group discussed which communities to engage and on which projects to start moving towards the joint vision of AI augmented facilities.

Early Demonstration Targets

- Magnetic confinement fusion: Surrogate models and experimental controls
- “InterTwin”: Multi-Science digital twin engine
- Laser-driven inertial fusion (LLNL, LBNL, Rochester) : AI surrogates
- Rock-IT: Digital twin for photon science with applicationsin catalytics
- BOFAB: Advanced Photon Lightsource + SLAC
- Digital earth
- OPTIMA: Real world evidence + AI for cancer
- ACCLAIM: AI for accelerator research
- ERVM-WAVE: AI for safe operation of Einstein telescope
- Helmholtz AI: Autonomous accelerators
- Synthetic data for particle physics
- MALA: Materials predictions from fundamental physics
- KITTEN: Energy responsible project for accelerators at KIT
- Center of Excellence for Research on AI- and Simulation-Based Engineering at Exascale (RAISE)
- ErUM-Data-Hub: central networking and transfer office for the digital transformation in the exploration of universe and matter.

General topics of interest for first steps

- Determining generalizable patterns of application of AI
- Workshops/Hackathons/Datathons/Studyathons/Ideathons
- Repositories for AI models
- Standardized intermediate representations
- BADGER/SLAC: Control optimizations, BlueSky
- Control standards for robotics
- Openness and interoperability
- High-throughput AI hardware
- Energy + Sustainability + Optimization
- Multi-target optimization
- Multi-X surrogate demonstrations
- Federated learning
- Hosting for training data?

4.4.3 Group 3: 1-3 Year Projects

What are the capabilities and time frames to make these capabilities a regular occurrence, i.e., the experimental/computing facility has an access mode for this

Experimental Facilities

- Proxy hardware (1Y)
- Access to test facilities (1-2Y)
  - connect different science teams
- Triggered analysis (2-3Y)
- Experimental steering (3Y)

Computing Facilities

- Access policies (1Y)
- Make schedules reliable and open (1Y)
- On demand / interactive computing (1-2Y)

Other important initial activities not directly linked to facilities

- Building working groups across multiple professional societies
- Staff education
  - Curriculum development
  - Summer schools
  - Maybe connect to academically minded staff members interested in teaching or even explicitly hiring these
- Workshops to find “killer” applications
- Social aspects
  - Courses aimed to get people comfortable rather than necessarily convey knowledge
  - Schedule coordination
- Joint proposals

5 Perspective Paper Development

Given the discussions outlined above the seminar broke into writing groups largely split by sections that started to develop outline and/or initial text. Below we report on the state of these developments by the end of the seminar. Once finalized the goal is to publish the polished results as a persectives paper in a yet to be determined venue.

5.1 Section 1: Needs

Michael Bussmann (Helmholtz-Zentrum Dresden-Rossendorf, DE)
Jean-Luc Vay (Lawrence Berkeley National Laboratory, US)
Arvind Ramanathan (Argonne National Laboratory – Lemont, US)

Gaps that AI can address

AI has enabled transformational progress in a number of scientific, engineering, and technology domains; however, enabling rapid progress in the adoption of AI strategies will not be entirely realized, unless these critical gaps are addressed.

1. AI is needed for optimizing the life-cycle of scientific data: While commercial applications (such as recommender systems) have ready access to quality datasets, scientific data is often more decentralized, distributed and less deterministic. Hence, one of the key needs of AI is in identifying the critical aspects from data, including achieving the ability to optimally reduce and represent vast scientific datasets, while also identifying potentially anomalous data points in them. Further, AI methods need to automatically enable the transformation of data → (actionable) information → (inferable) knowledge, which can significantly impact today's scientific processes.
2. AI needs for enabling close, near real-time coupling of modeling, simulations and experiments: Multimodal, multi-view, multi-scale data is extremely common in scientific processes. Specifically, experiments and sensor networks within integrated research infrastructure produce data at specific length- and time-scales for investigating complex phenomena and AI techniques can potentially act as 'glue' to stitch together such datasets. However, to truly enable experimental design at scale, there needs to be a strong coupling of how simulations and experiments inform each other about the complex phenomena of interest. This in turn requires AI methods to not only act as effective surrogate models, but also to respect foundational scientific principles that produce the underlying data distributions.
3. AI is needed for optimizing facility operations and control: Current scientific instrumentation within research facilities is largely informed by user access and individual research goals, which makes facilities to be extremely expensive and less optimal for cost-effective management. However, costs of managing research infrastructure can be significantly lowered by analyzing user access patterns and modeling, which can be used to optimize how experiments and simulations can be scheduled and automated. In essence, much like the car assembly line in the past, there is a need to industrialize the scientific process through meaningful engagement of scalable automation (such as robotics).
4. AI is needed for cross-domain, data-driven, automated, accelerated discovery: Current scientific progress is enabled in part by serendipitous discoveries within individual labs or consortiums, and still relies on domain-specific expertise drawn from intense training and experience. However, as is now evident, rapid progress in science and technology is enabled by 'cross-pollination' – wherein knowledge drawn from across disciplines and technology can mutually inform and benefit new foundation discoveries. Enabling such discoveries requires access to enormous datasets and analyses of patterns across such data which is enabled by modern AI techniques. Furthermore, instead of having individuals synthesize this knowledge, and design experiments, automated, data-driven knowledge distillation and hypotheses generation can direct experimental campaigns that can in turn accelerate the pace of scientific discoveries to address some of the aforementioned grand challenges.

5.2 Section 2a: Vision – Approach

Brian Spears (LLNL – Livermore, US)
Sunita Chandrasekaran (University of Delaware – Newark, US)
Matthew Streeter (Queen's University of Belfast, GB)

The pace and quality of scientific discovery can be greatly improved by erasing the boundary between experimentation and computation. Each step in the scientific process can be revisited with newer hypotheses through the application of machine learning to accelerate scientific discovery, which then enables orchestration of the whole activity. The scientist is then the conductor of this orchestra, and is placed back in control of the increasing complexity and huge scale of modern science.

Coupling scientific computing with large-scale experimentation using modern AI methods will introduce a new class of unified, AI-augmented facility. In such a facility, AI surrogates can be used to capture the intricacies of detailed simulations, but in a way that is accessible to experiments in seconds, not hours or months. Such facilities will then have access to simulations as a real-time commodity for informing experiments. Likewise, AI-guided analysis using representation learning can capture and reduce complicated diagnostic information on the fly. This promises to provide distilled and interpretable empirical data back to the computational world, again in moments, not months. AI-driven representations of both computational and experimental science, scientists can decide far more effectively – about the best next simulation to run or the most insightful next experiment to execute. In fact, these decisions can be accelerated and improved by using formal optimization algorithms that exploit compact AI representations to navigate to superior experimental outcomes; the decisions can even be made semi-autonomous or automated, allowing an optimized loop to execute well-informed experiments incredibly rapidly, keeping pace with modern facilities that perform experiments on the timescales of minutes down to milliseconds.

Its tightly AI-coupled components and efficient information processing ecosystem allow for scientific discovery on much shorter timescales than previously imaginable. Likewise, the AI-enabled operating environment simplifies the execution of simulation and experiment. This frees expert users of the previously disconnected facilities to think and innovate rather than labor over job submission, diagnostic data reduction, and similar operational tasks. Furthermore, non-expert users will find access to complicated science machines to have been democratized, allowing these new facilities to serve a far wider range of scientific communities.

We envision the technical roadmap of our AI-enabled facility to be an energy-efficient framework/design/infrastructure driven by novel hardware and software components that are connected together to form a unified system. With a goal to offer an AI-acceleration solution everywhere, newer dedicated AI-hardware will be built to address the need for a variety of commuting fronts on the diagnostic end, near the facility, at a data center and across a distributed compute network. We will also build software for orchestrating a variety of operations in both high performance compute and experiments. Such AI-orchestrated software will provide the necessary infrastructure to drive informed decisions thus enabling newer science. Facilities augmented by AI will also have a modernized strategy for handling volumes of heterogeneous data that carries context from generation until analysis.

5.3 Section 2b: Vision – Outcomes

Roger H. French (Case Western Reserve University – Cleveland, US)
Niko Bier (DLR, Institute of Aerodynamics and Flow Technology, DE)
Shantenu Jha (Brookhaven National Lab – Upton, US& Rutgers University – Piscataway, US)

AI will be applied at different levels in scientific facilities.

1. On a level close to and integrated into e.g. optical sensors AI will be used to directly provide scientific information rather than raw data to the facility. This will not only lead to significantly decreased requirements in bandwidth and data storage but will also drastically speed up the scientific experiment as such.
2. On a facility level AI will be used to generate digital twins of the facility and thus to control scientific experiments. This will significantly increase the efficiency of a facility by better coordinating the tools and sensors involved during the experiment.Moreover the status of the facility and its components will be accessible in real-time and predictive maintenance will become more efficient and will increase the availability and safety of the facility. The digital twin of a facility will also allow for the use of virtual sensors making it possible to collect previously not accessible data.
3. On an overall level distributed scientific facilities and HPC resources will be merged together into an AI-driven facility. In these facilities AI will be used to orchestrate the experiments. It will not only significantly speed up scientific experiments but will also be used for planning and conducting of experiments. AI will not only perform experiments better, but will conduct better experiments.

In such a new "generative science" completely new scientific questions will emerge and appropriate experiments will be conducted. Scientists will be able to focus on understanding and answering fundamental questions on the basis of an unprecedented quality of scientific experiments and their results.

5.4 Section 3: Software

Ravi Madduri (Argonne National Laboratory – Lemont, US)
Jeyan Thiyagalingam (Rutherford Appleton Lab. – Didcot, GB)

The software requirements for the AI Augmented Facilities are driven by users and operators of the facilities. In Software Development Life Cycle (SDLC), these processes are typically referred to as stakeholder analysis and requirement analysis. For the purpose of this document, we identified facility operators and scientists using the facilities as stakeholders. This section offers requirement analysis and identities of current methodologies and places where AI can augment and lead to better results.

We assume that AI-Augmented facilities have the following stages, regardless of the scientific domain (whether they are, e.g., photonic or laser or environmental or datacenter facilities) or whether they are centralized or distributed facilities.

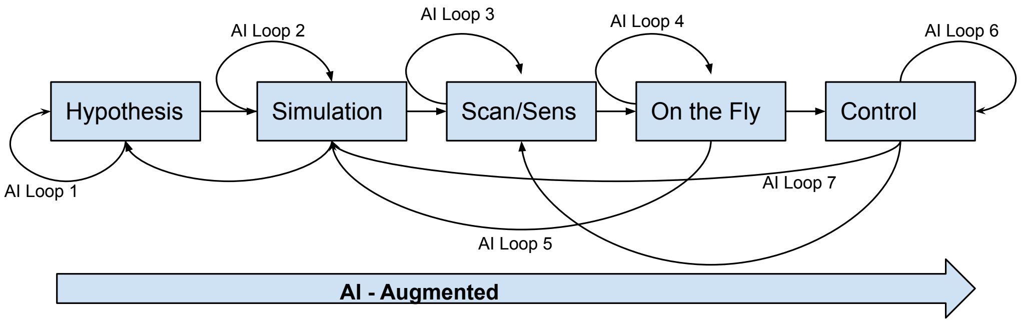


Figure 3 Examples of AI Loops at different stages: Synthetic GAN to simulate a facility that help validate or come up hypothesis (Omniverse) – AI Loop 1, Surrogate Models that help come up with better results from Simulation – AI Loop2, etc.

1. Basic hypothesis or expectation of the scientific outcomes from experiments or data acquisition
2. Simulation studies to understand the hypothesis better, and to develop insights into expected results off the potential experiments or data acquisition
3. Empirical data/fielding of experiments
4. Consolidation and interpretation of the data for developing better understanding or to underpin scientific discoveries
5. Tuning or controlling of the data acquisition step for improving the quality of the data

In the contemporary setting, some of these are human-controlled, which may eventually be replaced by an automated system component in the context of an AI-augmented facility. There are a number of challenges here that different stakeholders face here,especially in the absence of any AI-specific capabilities. These include the following:

- **Expensive Simulations:** The simulations can be a serious bottleneck to the overall progress, especially when they are of high-fidelity in nature, or when high-quality outputs are desirable,
- **Divergence between Simulated and Experimental Data:** The actual results from the experiments may as well be different to the hypothesis or expected results, owing to a number of reasons
  - Scope-insensitive calibration of the data acquisition instruments or sensors,
  - Incorrect a priori of hypothesis for simulations, and
  - Physical limitations of the data acquisition, such as potential dosage level or acquisition resolution of the detectors.
- **Data Analysis and Enhancements:** Understanding, interpreting and analyzing the experimental or sensed data, and correlating that with the a priori formed as part of the hypothesis, and
- **Feedback-driven Optimal Control** of the experimental facility. Optimal and stable control of the instruments are a serious practical challenge, especially in tolerant-sensitive contexts.

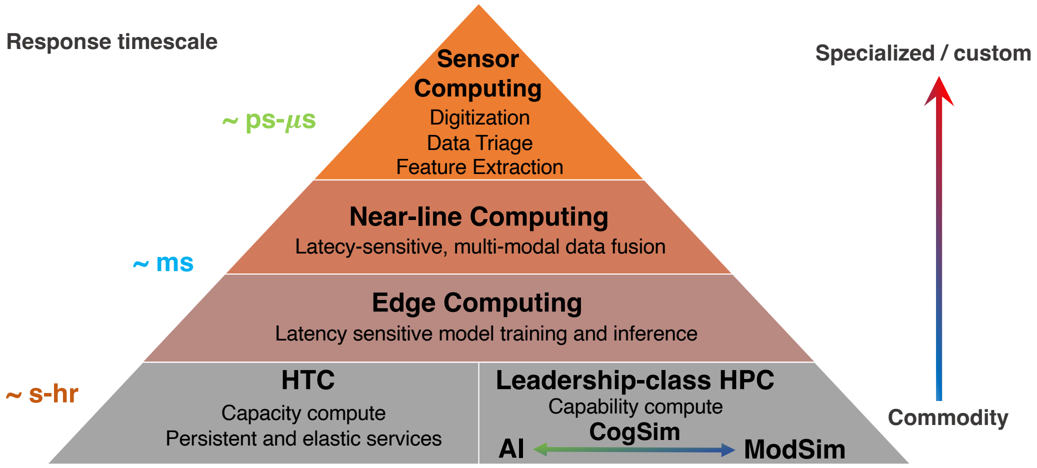
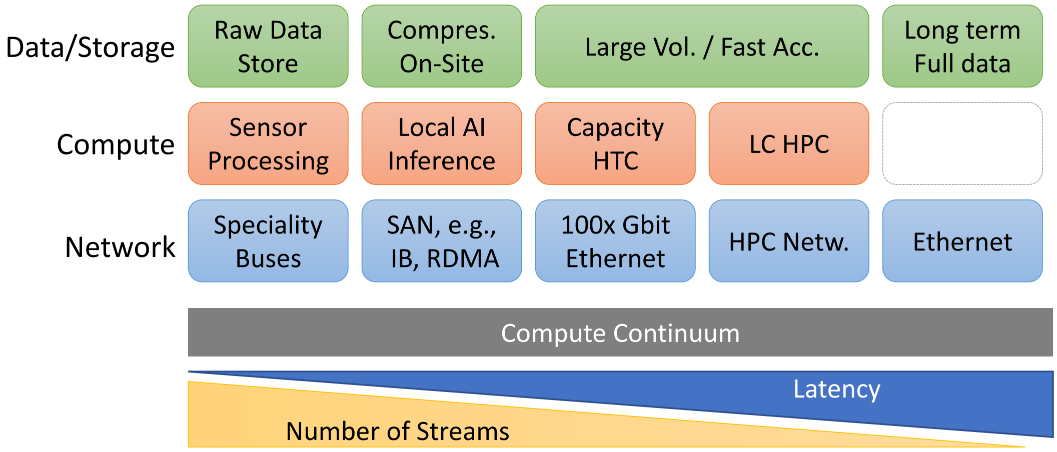


Figure 4 Current State-of-the-Art Hardware supporting Data Intensive Facilities.

5.5 Section 4: Hardware

Brian Spears (LLNL – Livermore, US)
Martin Schulz (TU München – Garching, DE)
Andrea Santamaria Garcia (KIT – Karlsruher Institut für Technologie, DE)



Data/Storage	Raw Data Store	Compres. On-Site	Large Vol. / Fast Acc.		Long term Full data
Compute	Sensor Processing	Local AI Inference	Capacity HTC	LC HPC	
Network	Speciality Buses	SAN, e.g., IB, RDMA	100x Gbit Ethernet	HPC Netw.	Ethernet

Figure 5 Current state-of-the-art of hardware systems supporting the design and operation of AI augmented facilities. It features a clearly segmented technology stack separated in compute, storage and network (horizontal) and technologies at different locations (vertical).

On the hardware level we are currently seeing three trends that will radically impact AI augmented facilities in how they are built, operated, and used:

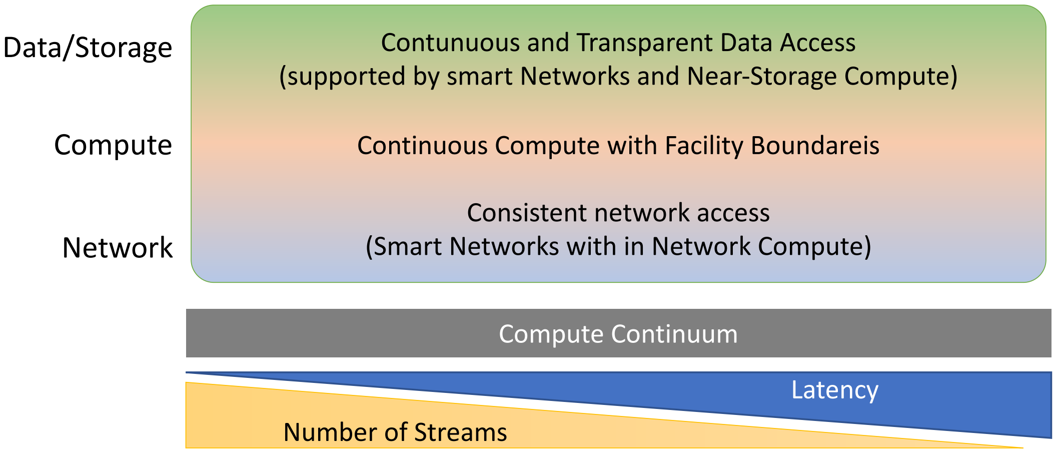

Data/Storage
Compute
Network
Contunuous and Transparent Data Access
(supported by smart Networks and Near-Storage Compute)
Continuous Compute with Facility Boundareis
Consistent network access
(Smart Networks with in Network Compute)
Compute Continuum
Latency
Number of Streams

Figure 6 Future hardware vision: integrated systems and APIs making storage and compute available independent of location and hardware system, supporting fused augmented facilities.

- Modern AI accelerators are moving from data-center-only use towards facility / on-premise systems making both inference and learning capabilities available closer to the data source.
- Convergence of compute and network in the form of SmartSwitches and SmartNICs, which enables processing on the fly and independent of actual location in a decentralized fashion
- Convergence of compute and storage in the form of Near or In Storage Compute, enabling low-latency data processing at the various and distributed storage location.

Figure 4 illustrates the hierarchy of needs for an AI-enhanced, integrated, experimental and compute facility.

- At the top of the hierarchy is the set of primary data inputs from the experiments represented as a set of sensors. The hardware needs here are extremely low-latency, working with single or small number of samples, high data capture rates, and most importantly engagement with a set of custom / bespoke / specialized sensors that have unique demands such as custom analog signaling that requires high-speed analog to digital converters (ADCs), proprietary formats, etc. Some of the key demands for AI at this level are **feature extraction, data processing / conditioning, local controletc. Primarily inference tasks**. Another class of sensors are those that are deployed in a Size, Weight, and Power (SWaP) constrained environment / facility. Typical examples of these are battery powered and may be geographically distributed (e.g. drones, buoys, . . .).
- The next level of the hierarchy is the latency-optimized AI-accelerator that is designed for data-fusion of heterogenous, multi-modal data near the input sensors, forming a “near-line” compute capability. These accelerators need to be optimized to process “single samples” of data (i.e. all of the data from a single event / shot / experiment), and cannot afford to batch multiple samples together prior to processing. Another way to think about this is that they need to be optimized for streaming data sets, where decisions and analysis have to be produced for each sample in real-time. [keywords: streaming, real-time, latency-optimized] Primarily focused on AI inference tasks.
- Integrating the output of the real-time sensor streams output from the near-line accelerators are the near-experiment computing resources. These resources are now expected to work with batches of data samples, and start to blend workloads that include both inference of more complex / larger models as well as the training or fine-tuning of existing or new models

- The bottom of the compute hierarchy is anchored by two classes of computing needs / ecosystems: Leadership-class High Performance Computing (LC HPC) aka capability computing and capacity-based High Throughput Computing (HTC) with persistent and elastic services.
  - HPC resources are intended for full-system, monolithic, modeling and simulation (ModSim) jobs and training of large neural networks (e.g. foundation models). They are designed with highly-interconnected accelerators (e.g. GPUs) using proprietary network architectures (InfiniBand, Slingshot, etc.) Traditionally, these are run in a batch-scheduled manner. Additionally, there is a need to establish new HPC system architecture designs that are optimized for traditional modeling and simulation codes (ModSim), AI training and inference workloads, as well as hybrid cognitive simulations (CogSim) workflows. Optimizations for ModSim include accelerators with high-precision data types (64b arithmetic), high cross-section bandwidth networking, and high-bandwidth write-optimized parallel file systems. HPC optimized for AI training and inference will leverage low-precision accelerators, with read-optimized parallel file systems and support for advanced object-stores. CogSim HPC systems will require a blend of both capabilities.
  - HTC resources are designed to serve the needs of many users, models, or data streams. Persistent and elastic services are designed to capture experimental data in real-time with the ability to meet data surges from the experiment. Additionally, they provide the ability to update / refine / fine-tune AI models in the background as new data is captured.

Fundamental shifts needed in the AI-accelerator architecture space:

1. Engagement with customized sensors requires AI-accelerators to interface with a myriad of signaling protocols, formats, and data types (including varying precisions). The current state of the practice is to develop custom accelerator sub-systems that are able to funnel the outputs of these sensors into standard AI-accelerators such as GPUs or a specializable systems such as a blended FPGA / AI-accelerator (such as a Xilinx Versal) through a series of customized ADC capture cards, FPGAs and ASICs. These data capture sub-systems can become quite complex and are as unique as the sensors themselves. The emergence of chiplet-based AI accelerator design would enable a fundamental shift in how research teams would be able to interface state of the art AI accelerators with their custom sensors. Chiplet-based architectures offer the promise of allowing multiple vendors to integrate proprietary hardware into unique hardware processes in a timely and affordable manufacturing process. Advances in this field would enable sensor manufacturers and research teams to more easily and efficiently couple SOTA AI accelerator hardware directly (or closely) to their sensors and thus unleash new capabilities that can be exploited at the sensor.
2. AI accelerators in the near-line computing regime will need to be tuned for real-time, streaming workloads, where they need to produce inference results on only a single sample of data that contains a large volume of heterogeneous data fields. Frequently these accelerators will need to be able to meet hard latency limits and execute neural network models (inference) with predictable performance characteristics.
3. AI-accelerators close to the sensor may need to meet the demands of a Size, Weight, and Power (SWaP) constrained environments. For these systems, the operational cost in terms of inference requests per Watt will be a key design metric.

5.6 Section 5: Facilities Upgrades/Actions

Kyle Chard (University of Chicago, US)
Derek Mariscal (LLNL – Livermore, US)
Rafael Ferreira da Silva (Oak Ridge National Laboratory, US)

Cross-cutting Requirements. There are several cross-cutting requirements that span experimental and computational facilities, broadly spanning the computing continuum from edge to data center.

Connectivity. Central to integration is the need for unimpeded access between experimental and computational facilities, such that experiment results can be rapidly available to AI models running nearby or at compute facilities and that AI informed decisions can be enacted at experiment facilities. Ideally, we require both in-bound and out-bound network access between participating entities, with minimal restrictions (e.g., firewalls, NAT). In many cases, regulations and policies will limit the degree to which connectivity is permitted; however, this represents an opportunity to explore approaches to reduce friction. We refer to the success of the Science DMZ model in science as a way of providing minimally impeded access to scientific data and resources.

Security. Traditionally, compute and experimental facilities have rigid security models that prevent automation. To attain semi/full automation it is necessary that facilities implement flexible and interoperable security models that enable remote and automated actions, while ensuring accountability for actions. As we move towards automated and AI-based techniques, there is a need for delegatable access, via which humans may permit operations to be performed within some bound and for some period of time. We further require methods to audit operations, restrict the scope of access, and revoke access rapidly.

Data. AI is dependent on data and as such movement between experiment and/or compute facilities is integral in enabling AI-augmented experiments. Important data movement characteristics may different significantly between use cases, for example, to deliver streams of experiment data for analysis, to move trained models to the edge, or to move huge amounts of simulation data to available compute resources. We require common interfaces (e.g., Globus, HTTPS, common messaging protocols) and methods to securely access, move, and share data between participants.

Storage. The plethora of data storage options spanning experiments to HPC presents obstacles for accessing and preserving important data. Methods are required to a) access storage that is physically and logically distributed; and b) optimize storage for different tiers of data (e.g., ephemeral storage of intermediate data vs long-term preservation of experiment data). Data access requires a consistent view of data irrespective of storage, and interfaces via which data can be accessed by new and traditional methods (e.g., using HTTP for integrated data visualization in existing tools).

Compute. AI-enabled methods will require that computation be fluid, such that it may be executed where it makes the most sense (e.g., near to a detector, where simulation data reside). Enabling such fluidity requires a) common interfaces to execute tasks (e.g., simulation, model training, inference) across the computing continuum, from edge to HPC; and b) portable packaging such that codes may be easily executed without herculean efforts to configure and contextualize environments. Community efforts such as Superfacility and funcX lay the groundwork to provide common APIs, while work in container technologies such as Shifter will support the need for portable codes.

Policies. Perhaps the most challenging problem to address is the need to support the varied policies across participating organizations, between users, and in consideration of AI-managed actions. Understanding of the requirements for defining a trustability model between facilities is a first step towards enabling an AI augmented automated facility.

Experimental Facilities. Experimental facilities will in many cases require updates to enable AI-augmentation. This includes diagnostics with integrated edge computing resources for rapid data distillation, remote communication with and precise control over machine inputs, robust pipelines for two-way communications with high performance computing, and new tools for monitoring the system/diagnostics health and calibration.

Detectors/measurements. Detector measurements and sensors collect data that will inform AI about the real world. Crucially, the detectors must be able to deliver electronic signals (i.e. the information must be readily digitized). For diagnostics without direct AI-ready interfaces, edge hardware must be deployed to provide for real-time high-accuracy data processing, localized storage (depending on network pipeline resources and latency needs), and access to diagnostic controls. Such hardware can include GPUs, FPGAs, or other specialized hardware that can perform complex calculations quickly (discussed in HW). By processing data at the edge, AI will be able to consume heterogenous data to drive the experimental process while retaining full-fidelity data for validation.

Controls access. In AI augmented experimental facilities, it will be necessary to interface with machine and diagnostic controls in ways that are often inaccessible to users. Common, open-source control interfaces such as EPICS [ref], BlueSky [ref] are examples for enabling control and data acquisition from heterogeneous hardware[ref] that could be readily accessed via AI while retaining proprietary interfaces. New operation policies for experimental facilities will need to be developed to allow control of high-value systems (accelerators, robotics, lasers, etc.) with AI with robust limitations that prioritize machine safety while allowing transformative capability for fine control.

Robot control. AI augmented experimental and computational facilities often use robots to automate tasks such as sample preparation, data collection, and analysis. To control these robots, it is essential to have a common interface that allows for arbitrary control of the robotics. This can be achieved by developing custom software that integrates with the robot's control system and is again amenable to autonomous control.

Asynchronous, event-based operation. Keeping pace with high experiment throughputs and rapid AI-based decisions will require movement to a more asynchronous model of communication and control. Event-based models have been transformative in industry as a way of decoupling monolithic applications and improving performance. Moving to event-based models will require common methods to a) detect events; b) propagate events across distributed systems; and c) make decisions based on complex heterogeneous event streams. For many applications, it will be important that events include timestamps such that they can be ordered. Consistency across event generators is application specific.

Compute Resources. Computational facilities often target the deployment of large-scale computing systems to solve complex problems from a diverse set of computational science domains. As the use of AI solutions has significantly increased in these domains, there is a need to provide specialized support to heterogeneous computing architectures and flexible policies that address the different set of requirements.

Flexible and available allocation. Reservations and dedicated resources are crucial for ensuring that researchers have access to the resources they need when they need them (i.e., urgent computing). To this end, there is a need for the development of policies that allow different levels of reservations based on the urgency of the application. On the other hand, it is also necessary to support pre-emptive mechanisms to meet unforeseen requirements or impromptu workloads.

Hardware. Specialized hardware (e.g., GPUs, TPUs, and upcoming DPUs) is used to accelerate compute- and data-intensive tasks including data analysis, machine learning training, and AI inference. Specialized inference hardware is used for real-time analysis and processing of experimental data. Enabling support for these resources requires specialized software for bridging these resources to traditional HPC and storage resources.

Policies (access). Policies are important to ensure that resources are used fairly and efficiently. In an AI augmented facility, policies will have to be extended to encompass the heterogeneous set of resources and requirements from near real-time to long-term campaigns that might involve both HPC and specialized hardware. Additionally, policies will have to be flexible regarding experiment/access management, i.e. in order to enable steering across facilities there might be needed to provide support to generic/global accounts or provide specialized APIs that can manage/access external resources from/to computing nodes.

5.7 Section 6: First Steps

Marina Ganeva (Forschungszentrum Jülich, DE)
Tom Gibbs (NVIDIA Corp. – Santa Clara, US)

The first steps include programmatic proof of concept (POC) starter projects that are co-led between the Data Center and Selected Experimental Facilities, community outreach with workshops and centers of excellence along with training and development events such as hackathons.

The POC projects will develop and demonstrate the new workflows that connect the Science Data center with the Experimental facility, where AI is used as part of the workflow that connects them.. An example of a program is the SuperFacility Project at NERSC[1]. Each POC project must have scope that is sufficient to demonstrate the potential for improved science and identify features and requirements that can't be met with the current infrastructure.

A key feature of the new workflows are AI algorithms is that once trained at the HPC Data Center are fast enough and accurate to be deployed at the experiments for control and/or improvement of experiment operation. Another feature is Active Learning that may be executed at the Data Center and/or the experimental facility that can be used in conjunction with experiment operation. Simulation workflows are critical for explainability and validation of AI approaches as well as for generation of synthetic training datasets.

Examples of projects include improving the control of a Fusion Experiment, Multi Messenger Astrophysics, Improving the process for drug discovery with a biology lab or data analysis/evaluation on the fly as a neutron/x-ray/electron/light scattering experiment executes. Key features have already been identified that will need to be acted on in the near term to allow for the new workflow include the following:

[1] https://www.nersc.gov/research-and-development/superfacility/

- The Data Center Facility will need to include features that allow interactive usage that is interrupt driven along with the current batch oriented usage model for long running jobs.
- Both the Data Center and Experiment will need to be able to support the bandwidth and latency for data transfer. Reconfigurable storage will also be needed to support the experiment as it executes, which acts as a workflow buffer between the storage at the experiment and persistent platform storage at the data center.
- The Experiment hardware/software should allow for automated control. Curated data in the form that it can be used by AI will also be required.
- Funding agencies will need to initiate projects to pursue the new use cases for a sufficient period of time to develop a working example.

Workshops that cross domains as well as specific to a given domain will be needed to promote the new methods, socialize new concepts, identify key requirements and avoid redundancy. Where possible Centers of Excellence, Data Hubs and other vehicles should be developed for concentrated effort on specific projects within a community. The workshops and COEs should be balanced with Hackathons and bootcamps that can serve to help train users on the new tools and approaches. Additionally, hackathons and bootcamps allow for collective efforts to address challenging problems relevant for multiple facilities (inverse problem-related issues, uncertainty quantification, etc.) that result in code that can be evaluated.

5.8 Section X: Figures

Peer-Timo Bremer (LLNL – Livermore, US)
Annika Eichler (DESY – Hamburg, DE)

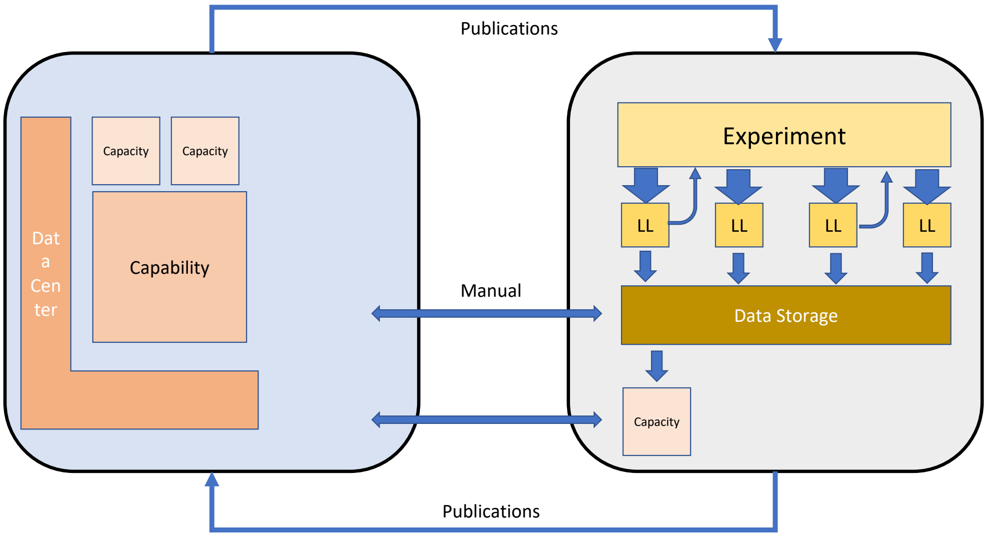


Figure 7 Conceptual drawing of the current state of the art in which computer centers (left) and experimental facilities are connected primarily through laborious manual data transfers and coordinated largely through publications.

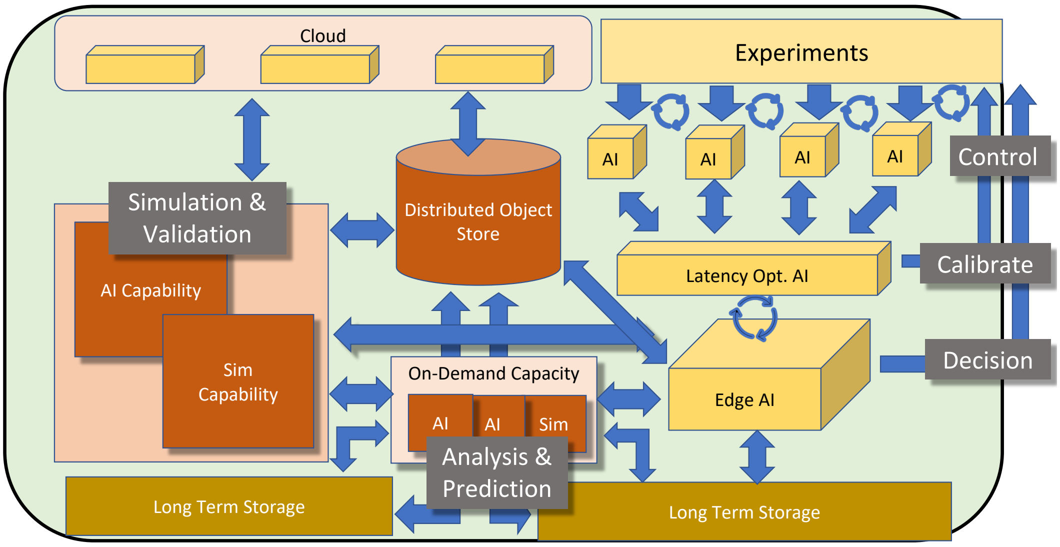


Figure 8 Conceptual drawing of future AI augmented facilities in which AI informs and supports all aspect of an experiment from real-time control to on premise compute, on demand processing, and large-scale simulations capabilties.

6 Acknowledgement

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344, LLNL-TR-854217; MJVS acknowledges support from the Royal Society URF-R1221874; RHF contribution was funded by Department of Energy's National Nuclear Security Administration under Award Number(s) DE-NA0004104; Further support from the U.S. Department of Energy, Office of Science, under Lawrence Berkeley National Laboratory contract number DE-AC02-456 05CH11231; Partially funded by the Center of Advanced Systems Understanding (CASUS), which is financed by Germany's Federal Ministry of Education and Research (BMBF) and by the Saxon Ministry for Science, Culture and Tourism (SMWK) with tax funds on the basis of the budget approved by the Saxon State Parliament.

Participants

- Niko Bier
DLR – Braunschweig, DE
- Martin Boehm
Institut Laue-Langevin – Grenoble, FR
- Peer-Timo Bremer
LLNL – Livermore, US
- Michael Bussmann
Helmholtz-Zentrum Dresden-Rossendorf – Görlitz, DE
- Sunita Chandrasekaran
University of Delaware – Newark, US
- Kyle Chard
University of Chicago, US
- Annika Eichler
DESY – Hamburg, DE
- Rafael Ferreira da Silva
Oak Ridge National Laboratory, US
- Roger French
Case Western Reserve University – Cleveland, US
- Marina Ganeva
Forschungszentrum Jülich, DE
- Tom Gibbs
NVIDIA Corp. – Santa Clara, US
- Maria Girone
CERN – Geneva, CH
- Shantenu Jha
Brookhaven National Lab – Upton, US & Rutgers University – Piscataway, US
- Thomas Kühne
Universität Paderborn, DE
- Ravi Madduri
Argonne National Laboratory – Lemont, US
- Derek Mariscal
LLNL – Livermore, US
- Arvind Ramanathan
Argonne National Laboratory – Lemont, US
- Andrea Santamaria Garcia
KIT – Karlsruher Institut für Technologie, DE
- Martin Schulz
TU München – Garching, DE
- Brian Spears
LLNL – Livermore, US
- Matthew Streeter
Queen's University of Belfast, GB
- Jeyan Thiyagalingam
Rutherford Appleton Lab. – Didcot, GB
- Brian Van Essen
LLNL – Livermore, US
- Jean-Luc Vay
Lawrence Berkeley National Laboratory, US