DAGSTUHL
REPORTS

Volume 9, Issue 5, May 2019

Dagstuhl Reports, Vol. 9, Issue 5 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
http://www.dagstuhl.de/dagpub/2192-5283

Publication date
November, 2019

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 `http://dnb.d-nb.de`.

Digital Object Identifier: 10.4230/DagRep.9.5.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.

Editorial Office
Michael Wagner*(Managing Editor)*
Jutka Gasiorowski *(Editorial Assistance)*
Dagmar Glaser *(Editorial Assistance)*
Thomas Schillo *(Technical Assistance)*

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

Report from Dagstuhl Seminar 19191

Software Evolution in Time and Space: Unifying Version and Variability Management

Edited by

Thorsten Berger[1], Marsha Chechik[2], Timo Kehrer[3], and Manuel Wimmer[4]

1 **Chalmers and University of Gothenburg, SE,** thorsten.berger@chalmers.se
2 **University of Toronto, CA,** chechik@cs.toronto.edu
3 **HU Berlin, DE,** timo.kehrer@informatik.hu-berlin.de
4 **Johannes Kepler Universität Linz, AT,** manuel.wimmer@jku.at

Abstract

Effectively managing versions and variants of software systems are among the main challenges of software engineering. Over the last decades, two large research fields, Software Configuration Management (SCM) and Software Product Line Engineering (SPLE), have focused on addressing the version and the variant management, respectively. Yet, large-scale systems require addressing both challenges in a unified way. The SCM community regularly faces the need to support variants, while SPLE needs versioning support. However, neither community has been successful in producing unified version and variant management techniques that are effective in practice. This seminar aimed at establishing a body of knowledge of version and variant management techniques. Together with industrial practitioners, we invited researchers from both fields to conceive an ontology of SCM and SPLE concepts, to identify open problems, and to elicit and synthesize practitioners' challenges and requirements. These outcomes provided the basis to create a research agenda, research infrastructure, and working groups, and finally, to establish a benchmark for evaluating future research results. As such, the seminar enabled research on enhanced version and variant management techniques that will ultimately be adopted in practice.

Seminar May 5–10, 2019 – http://www.dagstuhl.de/19191

2012 ACM Subject Classification Software and its engineering → Software configuration management and version control systems, Software and its engineering → Software product lines, Software and its engineering → Empirical software validation

Keywords and phrases software configuration management, versioning, variability management, software product lines, empirical evaluation

Digital Object Identifier 10.4230/DagRep.9.5.1

Edited in cooperation with Wardah Mahmood

Software Evolution in Time and Space: Unifying Version and Variability Management, *Dagstuhl Reports*, Vol. 9, Issue 5, pp. 1–30
Editors: Thorsten Berger, Marsha Chechik, Timo Kehrer, and Manuel Wimmer
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Thorsten Berger
Marsha Chechik
Timo Kehrer
Manuel Wimmer

Overview and Motivation

Modern software systems evolve rapidly and often need to exist in many variants. Consider the Linux kernel with its uncountable number of variants. Each variant addresses different requirements, such as runtime environments ranging from Android phones to large super-computers and server farms. At the same time, the Linux kernel frequently boasts new versions, managed by thousands of developers. Yet, software versions–resulting from evolution in time–and variants–resulting from evolution in space–are managed radically differently. Version management relies on a version control system (Git) and sophisticated workflows–concepts that have been developed for decades in the field of software configuration management (SCM) [12, 24, 23]. Variant management in the Linux kernel relies on techniques known from the field of software product line engineering (SPLE) [27, 11, 13], such as an integrated software platform, a variant-aware build system [7], an interactive configurator tool [30], and a model-based representation [9, 8, 17, 1] of all kernel features [4, 28]. The Linux kernel is exemplary for many large-scale, variant-rich, and rapidly evolving software systems in industry [5, 3, 32], especially in the domains of embedded, cyber-physical, automotive, and avionics control systems.

Despite decades of research in both fields, the effective evolution of variant-rich systems is still an open problem. Three main challenges exist. First, while version control systems are well-integrated into development processes, product-line engineering requires investment into additional tooling and different processes that are difficult to adopt. In fact, organizations rarely adopt product line engineering from scratch [6], but rather use readily available version control systems with their branching and forking facilities–a strategy known as clone & own [15, 10]. While this strategy is simple, it does not scale with the number of variants, and then requires evolving (i.e., re-engineering) cloned variants into a product-line platform [2]. Second, evolving product-line platforms is substantially more complex than evolving single variants, mainly since developers need to work on all variants at the same time [25]. Third, the granularity of tracking versions of variants is still unclear. While the whole platform can be versioned, ideally, versioning at the level of features should be supported.

In summary, SCM and SPLE are two widely established, yet actively researched software engineering disciplines offering a variety of concepts to deal with software versions and variants [16, 14, 18, 21]. Yet, despite various attempts [22, 34, 33, 20], none of the two disciplines has been successful in establishing unified solutions addressing both problems at the same time–mainly due to the isolation of both communities and due to the absence of realistic and widely accepted requirements on how to evaluate the effectiveness of techniques for managing both versions and variants.

Goals of the Seminar

This Dagstuhl Seminar aimed at establishing a body of knowledge on unified version and variant management. We invited leading practitioners and researchers from both disciplines to discuss each other's challenges, solutions, and experiences. The seminar's goals were to: (i) survey state-of-the-art SCM and SPLE concepts and map both areas' terminologies and open problems, (ii) gather industrial and academic challenges and requirements on integrated version and variant management, (iii) survey and assess existing evaluation approaches, and (iv) stablish a research agenda, research infrastructure, and working groups. To guide future research, the participants also discussed the basis to work on improved evaluation approaches–as benchmarks for new version and variant management techniques. As such, the long-term goal of the seminar was to enable the development and evaluation of enhanced version and variant management techniques that will be adopted in practice.

Week Overview

Monday. After an introduction of all participants, the seminar started off with general talks on versioning and variability. Bernhard Westfechtel set the stage with an introduction into version management concepts and workflows, which already illustrated some overlap with variability management concepts. For instance, directed deltas are conceptually similar to compositional variation mechanisms (e.g., feature modules or delta modules), and the construction of versions in intensional versioning can be related to the configuration-based derivation of individual variants from a product-line platform. The seminar continued with a talk by Don Batory, who discussed the integration of version control systems, variability management techniques, and integrated development environments (IDEs) based on ideas centering around a better representation and execution of program refactorings in versioned and variant-rich software systems. The talk by Thorsten Berger (actually given on Tuesday, since the introduction round and discussions for the other talks took more time) followed up on the concepts introduced in the previous talks and presented a survey on variation control systems, which support developers managing variant-rich systems in terms of features. Such variation control systems go back to the end of the 1970s with concepts and prototypes developed in the SCM community, but never made it into the mainstream. The talk surveyed their concepts and discussed problems likely prohibiting their adoption. Thereafter, we enjoyed three talks on industrial perspectives given by our industrial practitioners: Henrik Lönn (Volvo), Danilo Beuche (pure::systems), and Ramesh S. (General Motors; talk also given on Tuesday for timing reasons), confirming and explaining the gaps between academia and industry.

Tuesday. The day started with an introduction into the prospective breakout groups for the afternoon, followed by the talk of Christoph Seidl on versioning of product lines relying on a representation of feature versions in a new dialect of feature models, called Hyper Feature Models. Thereafter, the breakout sessions on four relevant topics took place, specifically: on a conceptual model to map SPLE and SCM concepts, on operations for managing versions and variants, on analyses of versions and variants, on workflows for managing versions and variants, and on first-class support of variability and versioning in programming languages. A benchmarking group was discussed, but abandoned in favor of first working on the foundations before discussing benchmarking techniques to evaluate prospective unified techniques for versioning and variability. The breakout group discussions continued until the afternoon, before the remaining talks from Monday were given (Thorsten Berger and Ramesh Sethu), followed by lightning talks from Shurui Zhou and Sandro Schulze. Shurui discussed the relevance of version and variability management in the domain of engineering AI-based systems, where models and large dataset need to be managed. Sandro proposed a round-trip-engineering process relying on unified management of versioning and variability, relying on automated extraction of variability information from cloned variants (which should be integrated into a platform in a round-trip-engineering manner).

Wednesday. We started the day with a talk by Daniel Strüber on benchmarking scenarios and a survey of existing benchmarks. In fact, it is a common consensus of the community that the lack of strong, landmark benchmarks hinders the progress in both communities (SCM and SPLE). Thereafter, Yi Li presented his work on slicing of the history of software codebases along features, where features are represented by test cases to help identifying the relevant code in a longitudinal manner. Thomas Thüm then presented a vision on the–ideally automated–synchronization of cloned variants as followed by the VariantSync research project which is led by Thomas and Timo Kehrer. Thomas also presented a very first prototypical implementation of the VariantSync tool. The approach shares, based on audience feedback, ideas with the Virtual Platform, proposed by researchers in 2014 [1]. In the afternoon, the majority of the participants continued their discussion on their group trip to the city of Trier and a dinner at a local winery.

Thursday. The day began with a talk by Gabriele Taentzer, presenting a generalizing framework for transformations of software product lines, relying on the formalism of category theory. Another talk was given by Julia Rubin on equivalence checking of variants based on behavior instead of structural characteristics of changes. Thereafter, the breakout groups continued their discussions until the later afternoon, where the results were presented to the other seminar participants. After dinner, two lightning talks were given by Paulo Borba and Iris Reinhartz-Berger. Paulo discussed the detection of semantic merge conflicts in the light of avoiding unwanted feature interactions, and Iris presented insights from two research projects on behavior-derived variability analysis and mechanisms recommendation.

Friday. The last day of the seminar started with a talk by Lukas Linsbauer on his work towards a feature-oriented and distributed version-control system, relying on the variant-integration tooling ECCO. We then had a closing discussion, re-iterating the main challenges we identified throughout the seminar, as well as discussing future work.

Outcome of the Seminar

The seminar established breakout groups who continued their discussion after the seminar and already published two papers [7, 1] at the VariVolution workshop, hosted at the Systems and Software Product Line Conference (SPLC). In addition, a paper accepted at the main track of SPLC on benchmarking, relying on input from the seminar participants via a survey [31], and providing an initial infrastructure for community-oriented benchmark creation,[1] can be seen as a core outcome of the seminar.

A core topic of the final discussion was the teaching of SPLE and SCM concepts–an important means to eventually improve the handling of versions and variants in practice. One of the problems identified is that, while SCM is covered sufficiently, the relevant variability-

[1] https://bitbucket.org/easelab/evobench

management concepts are not taught at the Bachelor's level in the majority of universities. However, the discussants believe that practicing feature-oriented analysis and design early in the curriculum would be beneficial, where currently object-oriented analysis and design is dominating. Interestingly, based on the experience of the discussants, SPLE is still seen as something rather futuristic by students, which is somewhat surprising, given that building highly configurable systems and software platforms are established practices, so perhaps there is a perception and awareness problem that teaching needs to address. Naturally, a course teaching SPLE at the Bachelor's level should also teach the relevant SCM concepts. A closely related topic discussed is that of teaching architectures, especially those of product lines, which is not really in the focus of current software architecture courses. Of course, it is generally difficult to talk to students about software architecture, since, as a discussant explains, a relevant abstract concept that students do not immediately perceive as relevant in the course of the studies. In contrast, with compilers and databases, students obtain some hands-on experience, which allows them to relate more closely to, especially with respect to a future job in the industry. This calls for close collaboration with industry in SPLE courses.

Establishing benchmarks turned out to be a more difficult problem than expected. Benchmarking was prominently discussed, as well as input elicited for a set of 11 high-level benchmarking scenarios defined by some of the seminar participants and organizers before the seminar. The participants plan to follow-up on creating concrete benchmarks upon the infrastructure created.[1]One idea is to build a web application to contribute specific benchmark data (e.g., code integration examples, comprising the original code variants and the final result as a ground truth) to establish a community benchmark. Another interesting comment was that the currently published case studies and experience reports about variability management and product lines are relatively old and do not provide sufficient technical details. Furthermore, they also do not highlight the problems associated with clone & own and the need for product-line migration techniques adequately. This discussion is a call to arms for improving the benchmarking situation in the SCM and SPLE community.

Last but not least, an important outcome of the final discussion session of the seminar is the need for a commonly agreed set of core concepts, mechanisms and practices–a well-documented Body of Knowledge (BOK) of our discipline. Currently, only some aspects of versioning in time and space are partially covered by the Software Engineering BOK (SWEBOK). However, for promoting a consistent view of our discipline worldwide and beyond our discipline borders as well as for having a foundation for a consistent curriculum development, a dedicated BOK or an extension of the SWEBOK may be necessary as a community effort.

References

1 Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. *Feature-Oriented Software Product Lines.* Springer, Berlin Heidelberg, 2013.

2 Wesley K. G. Assunção, Roberto E. Lopez-Herrejon, Lukas Linsbauer, Silvia R. Vergilio, and Alexander Egyed. Reengineering legacy applications into software product lines: a systematic mapping. *Empirical Software Engineering*, 22(6):2972–3016, 2017.

3 Jonatas Ferreira Bastos, Paulo Anselmo da Mota Silveira Neto, Padraig O'Leary, Eduardo Santana de Almeida, and Silvio Romero de Lemos Meira. Software product lines adoption in small organizations. *Journal of Systems and Software*, 131(Supplement C):112–128, 2017.

4 Thorsten Berger, Daniela Lettner, Julia Rubin, Paul Grünbacher, Adeline Silva, Martin Becker, Marsha Chechik, and Krzysztof Czarnecki. What is a Feature? A Qualitative Study of Features in Industrial Software Product Lines. In *SPLC*, 2015.

5 Thorsten Berger, Divya Nair, Ralf Rublack, Joanne M. Atlee, Krzysztof Czarnecki, and Andrzej Wasowski. Three cases of feature-based variability modeling in industry. In *MODELS*, 2014.

6 Thorsten Berger, Ralf Rublack, Divya Nair, Joanne M. Atlee, Martin Becker, Krzysztof Czarnecki, and Andrzej Wąsowski. A Survey of Variability Modeling in Industrial Practice. In *VaMoS*, 2013.

7 Thorsten Berger, Steven She, Krzysztof Czarnecki, and Andrzej Wąsowski. Feature-to-Code mapping in two large product lines. In *SPLC*, 2010.

8 Thorsten Berger, Steven She, Rafael Lotufo, Andrzej Wąsowski, and Krzysztof Czarnecki. Variability modeling in the real: A perspective from the operating systems domain. In *ASE*, 2010.

9 Thorsten Berger, Steven She, Rafael Lotufo, Andrzej Wąsowski, and Krzysztof Czarnecki. A Study of Variability Models and Languages in the Systems Software Domain. *IEEE Transactions of Software Engineering*, 39(12):1611–1640, 2013.

10 John Businge, Openja Moses, Sarah Nadi, Engineer Bainomugisha, and Thorsten Berger. Clone-based variability management in the Android ecosystem. In *ICSME*, 2018.

11 Paul Clements and Linda Northrop. *Software Product Lines: Practices and Patterns.* Addison-Wesley, Boston, MA, 2001.

12 Reidar Conradi and Bernhard Westfechtel. Version models for software configuration management. *ACM Comput. Surv.*, 30(2):232–282, 1998.

13 Krzysztof Czarnecki and Ulrich W. Eisenecker. *Generative Programming: Methods, Tools, and Applications.* Addison-Wesley, Boston, MA, 2000.

14 Danny Dig, Kashif Manzoor, Ralph Johnson, and Tien N Nguyen. Refactoring-aware configuration management for object-oriented programs. In *ICSE*, 2007.

15 Yael Dubinsky, Julia Rubin, Thorsten Berger, Slawomir Duszynski, Martin Becker, and Krzysztof Czarnecki. An exploratory study of cloning in industrial software product lines. In *CSMR*, 2013.

16 Jacky Estublier, David Leblang, André van der Hoek, Reidar Conradi, Geoffrey Clemm, Walter Tichy, and Darcy Wiborg-Weber. Impact of software engineering research on the practice of software configuration management. *ACM Transactions on Software Engineering and Methodology*, 14(4):383–430, 2005.

17 Kyo Kang, Sholom Cohen, James Hess, William Nowak, and Spencer Peterson. Feature-oriented domain analysis (FODA) feasibility study. Technical Report SEI-90-TR-21, CMU, 1990.

18 Timo Kehrer, Udo Kelter, and Gabriele Taentzer. A rule-based approach to the semantic lifting of model differences in the context of model versioning. In *ASE*, 2011.

19 Jacob Krueger, Wanzi Gu, Hui Shen, Mukelabai Mukelabai, Regina Hebig, and Thorsten Berger. Towards a better understanding of software features and their characteristics: A case study of Marlin. In *VaMoS*, 2018.

20 Vincent J. Kruskal. Managing multi-version programs with an editor. *IBM Journal of Research and Development*, 28(1):74–81, 1984.

21 Philip Langer, Manuel Wimmer, Petra Brosch, Markus Herrmannsdörfer, Martina Seidl, Konrad Wieland, and Gerti Kappel. A posteriori operation detection in evolving software models. *Journal of Systems and Software*, 86(2):551–566, 2013.

22 Lukas Linsbauer, Thorsten Berger, and Paul Grünbacher. A classification of variation control systems. In *GPCE*, 2017.

23 Stephen A. MacKay. The state of the art in concurrent, distributed configuration management. In *SCM-4 and SCM-5*, 1995.

24 Axel Mahler. Configuration management. Chapter Variants: Keeping Things Together and Telling Them Apart. Wiley, 1995.

25 Jean Melo, Claus Brabrand, and Andrzej Wąsowski. How does the degree of variability affect bug finding? In *ICSE*, ACM.

26 Mukelabai Mukelabai, Damir Nešić, Salome Maro, Thorsten Berger, and Jan-Philipp Steghöfer. Tackling combinatorial explosion: A study of industrial needs and practices for analyzing highly configurable systems. In *ASE*, 2018.

27 David Parnas. On the design and development of program families. *IEEE Transactions on Software Engineering*, 2(1):1–9, 1976.

28 Leonardo Passos, Jesus Padilla, Thorsten Berger, Sven Apel, Krzysztof Czarnecki, and Marco Tulio Valente. Feature scattering in the large: A longitudinal study of Linux kernel device drivers. In *MODULARITY*, 2015.

29 Christopher Pietsch, Timo Kehrer, Udo Kelter, Dennis Reuling, and Manuel Ohrndorf. SiPL–A Delta-Based Modeling Framework for Software Product Line Engineering. In *ASE*, 2015.

30 Julio Sincero, Horst Schirmeier, Wolfgang Schröder-Preikschat, and Olaf Spinczyk. Is the Linux kernel a software product line. In *Workshop on Open Source Software and Product Lines*, 2007.

31 Daniel Strueber, Mukelabai Mukelabai, Jacob Krueger, Stefan Fischer, Lukas Linsbauer, Jabier Martinez, and Thorsten Berger. Facing the truth: Benchmarking the techniques for the evolution of variant-rich systems. In *SPLC*, 2019.

32 Christer Thörn. Current state and potential of variability management practices in software-intensive SMEs: Results from a regional industrial survey. *Information and Software Technology*, 52(4):411–421, 2010.

33 Eric Walkingshaw and Klaus Ostermann. Projectional editing of variational software. In *GPCE*, 2014.

34 Bernhard Westfechtel, Bjørn P. Munch, and Reidar Conradi. A layered architecture for uniform version management. *IEEE Transactions of Software Engineering*, 27(12):1111–1133, 2001.

2 Table of Contents

Executive Summary

Overview of Talks

Lightning Talks

Breakout Groups

3 Overview of Talks

3.1 Version Models for Software Configuration Management

Bernhard Westfechtel (Universität Bayreuth, DE)

Joint work of Reidar Conradi, Bernhard Westfechtel
Main reference Reidar Conradi, Bernhard Westfechtel: "Version Models for Software Configuration Management", ACM Comput. Surv., Vol. 30(2), pp. 232–282, 1998.
URL https://doi.org/10.1145/280277.280280

This talk focuses on the version models underlying both commercial systems and research prototypes of software configuration management systems. It introduced the notion of a version, which is a state of an evolving item. According to different dimensions of evolution, versioning is classified into temporal, logical, and cooperative versioning. In particular, revisions and variants are versions evolving in time and space, respectively. In the case of state-based versioning, a version is specified in terms of the states of versioned items. In the case of change-based versioning, a version is specified in terms of changes being applied to a base version. The version space is a structure which describes the organization of versions, e.g., by version graphs consisting of versions and successor relationships. Finally, we may distinguish between extensional and intensional versioning. In the former case, a version set is defined by explicit enumeration of its members; in the latter case, a version is described by a predicate specifying its desired properties. This distinction is crucial: SCM systems based on extensional versioning focus on the reconstruction of versions which were submitted to the repository; SCM systems based on intensional versioning need to construct versions which may have never been created before, such that specified version properties hold. While traditional SCM systems focus on extensional, temporal, and cooperative versioning, they do provide limited support for variants, which may be represented as branches in a version graphs. However, this approach breaks down in the case of multi-dimensional variations, due to the combinatorial explosion of the number of branches and the multiple maintenance problem of repeating changes on each affected branch. To solve this problem, a number of SCM systems were developed which do support multi-dimensional variation. It turns out that these systems are based on similar concepts, as they were developed in the context of software product line engineering.

3.2 Roadmap to Revolutionize IDE SPL Technology

Don Batory (The University of Texas at Austin, US)

Integrating variability–a.k.a., software product lines (SPLs)–with version control is an exciting and largely unexplored research topic. I encountered this topic in 2007 when visiting a US government facility that created SPLs for a fleet of ships. In following this lead, I encountered yet another technical problem that was similar, but (I thought) more critical to explore–the integration of variability (SPLs) with refactoring–and the need to modernize (Java or OO) IDE support for building SPLs, so that modern OO program development practices can be applied to BOTH SPL's one-of-a-kind programs and SPLs.

In this talk, I explain key ideas my colleagues and I discovered in integrating variability with OO refactoring, and how the challenges are very similar to that of integrating variability with version control. I believe it will be useful to be aware of these similarities as research with version control proceeds, because ultimately what modern SPL tooling requires is an integration of variability with version control AND refactorings.

References

1 Jongwook Kim, Don Batory and Danny Dig, "Refactoring Java Software Product Lines," In *SPLC*, 2017.

2 Don Batory, Invited Presentation: "Program Refactoring, Program Synthesis, and Model-Driven Development," In *ETAPS*, 2007.

3 Danny Dig, PhD Thesis: "Automated Upgrading of Component-Based Applications," 2007.

3.3 Closing The Gap

Danilo Beuche (pure::systems GmbH, DE)

This talk discusses whether there is a gap between the current "main-stream" product line engineering (PLE) research and the challenges in the industrial application of PLE. To that end, it presents typical usage scenarios of the commercial product-line engineering tool pure::variants and shows if and how it deals with some of the challenges encountered. The first challenge is the need to investigate if versioning is a suitable alternative of PLE, and more importantly if it is in fact cheaper than investing in a product-line architecture. The second challenge is the co-evolution of the product line and how to address it. This involves change propagation and merging in multiple variants, and upward propagation of the changes in the variants to the product line base code. The third challenge comes out of the question of whether product-line owners and product-line engineers should be consulted to get their viewpoints. The challenge is to establish if the right people are working in the industry. Finally, a couple of discussion points for the seminar are raised.

3.4 Variants and Versions in a Vehicle Integration Context

Henrik Lönn (Volvo Group Trucks Technology, SE)

Automotive embedded systems are increasingly critical and capable. For this reason, rigorous verification is necessary. Simulation based techniques multiply the test velocity and allow rare and dangerous situations to be exercised. The rich variability of control software and mechanical configurations calls for systematic variability management to secure verification confidence and completeness. This talk will address a model-based approach where software, electronics, and mechanics is represented for the purpose of both simulation and software development towards target execution. Variability is a key aspect, in order to jointly configure the embedded system and its models. Development is performed iteratively and incrementally with continuous integration, and simulations are therefore generated automatically including the variability resolution phase.

3.5 Versioning & Product Lining is Automotive ECS: Challenges and Strategies

Ramesh Sethu (General Motors R&D, US)

There is an ever increasing demand for Automotive Electronics, Control and Software (ECS) Assets in a vehicle: Today's advanced driver assistance systems in automotive systems, like Adaptive Cruise Controllers are giving rise to more complex and safety-critical Level 2 and beyond, features. Potential safety violations and security vulnerabilities are plaguing the industry. To meet the demands of rigorous development of software, more emphasis is being placed on additional life cycle artefacts like requirements, design models, simulation and test results which significantly impacts the resource requirements for building these systems. Reuse and product lining are some of the techniques used in the industry to amortize the costs of developing these systems over larger product portfolio.

The talk highlights and emphasizes the need for versioning and product line mechanisms extended to all life cycle assets, besides software. Software is reasonably structured and the standard techniques used in the software industry for a product lining and versioning may not be easily extended to unstructured and higher dimensional assets like textual requirements and simulation results. There is also a strong need for tracing the relationship between the artefacts and their versions and variants which would help in error analysis and resolution. The amount of information available across the life cycle is enormous and it would be interesting to see whether advances in data analysis can be extended to solve these problems. Efficient management of the large amount of unstructured data is also required. To reduce the overhead of managing large amount of data, opportunities for reducing or containing the revisions and variants are also very important. The last but not the least aspect is the development of appropriate tools for managing the life cycle assets across their revisions and variants.

3.6 Managing Variability in Space and Time in Software Families

Christoph Seidl (Technische Universität Braunschweig, DE)

Main reference Christoph Seidl, Ina Schaefer, Uwe Aßmann: "Integrated management of variability in space and time in software families", in Proc. of the 18th International Software Product Line Conference, SPLC'14, Florence, Italy, September 15-19, 2014, pp. 22–31, ACM, 2014.
URL https://doi.org/10.1145/2648511.2648514

Software product lines (SPLs) and software ecosystems (SECOs) encompass a family of closely related software systems in terms of common and variable assets that are configured to concrete products (variability in space). Over the course of time, variable assets of SPLs and especially SECOs are subject to change in order to meet new requirements as part of software evolution (variability in time). In many cases, both dimensions of variability have to be handled simultaneously, e.g., as not all customers upgrade their respective products immediately or completely. In this presentation, we introduce an integrated approach to manage variability in space and time in software families using Hyper Feature Models (HFMs) with feature versions and combine them with an extension of the transformational variability realization mechanism delta modeling. This allows derivation of concrete software systems from an SPL or SECO configuring both functionality (features) as well as versions.

References

1 Seidl, Aßmann: Towards Modeling and Analyzing Variability in Evolving Software Ecosystems, In *VaMoS*, 2013.

2 Seidl, Schaefer, Aßmann: Capturing Variability in Space and Time with Hyper Feature Models, In *VaMoS*, 2014.

3.7 A Classification of Variation Control Systems

Thorsten Berger (Chalmers | University of Gothenburg, SE)

Joint work of Lukas Linsbauer, Thorsten Berger, Paul Grünbacher
Main reference Lukas Linsbauer, Thorsten Berger, Paul Grünbacher: "A classification of variation control systems", in Proc. of the 16th ACM SIGPLAN International Conference on Generative Programming: Concepts and Experiences, GPCE'17, Vancouver, BC, Canada, October 23-24, 2017, pp. 49–62, ACM, 2017.
URL https://doi.org/10.1145/3136040.3136054

Version control systems are an integral part of today's software and systems development processes. They facilitate the management of revisions (sequential versions) and variants (concurrent versions) of a system under development and enable collaboration between developers. Revisions are commonly maintained either per file or for the whole system. Variants are supported via branching or forking mechanisms that conceptually clone the whole system under development. It is known that such cloning practices come with disadvantages. In fact, while short-lived branches for isolated development of new functionality (a.k.a., feature branches) are well supported, dealing with long-term and fine-grained system variants currently requires employing additional mechanisms, such as preprocessors, build systems or custom configuration tools. Interestingly, the literature describes a number of variation control systems, which provide a richer set of capabilities for handling fine-grained system variants compared to the version control systems widely used today. In this paper we present a classification and comparison of selected variation control systems to get an understanding of their capabilities and the advantages they can offer. We discuss problems of variation control systems, which may explain their comparably low popularity. We also propose research activities we regard as important to change this situation.

References

1 Stefan Stanciulescu, Thorsten Berger, Eric Walkingshaw, Andrzej Wasowski. Concepts, Operations, and Feasibility of a Projection-Based Variation Control System. In ICSME, 2016.

2 Stefan Fischer, Lukas Linsbauer, Roberto E. Lopez-Herrejon, Alexander Egyed. The ECCO tool: Extraction and composition for clone-and-own. In ICSE, 2015.

3 Felix Schwägerl, Bernhard Westfechtel. SuperMod: tool support for collaborative filtered model-driven software product line engineering. In ASE, 2016.

4 Bjørn Gulla, Even-André Karlsson, and Dashing Yeh. Change-oriented Version Descriptions in EPOS. Softw. Eng. J., 6(6):378–386, 1991.

5 Vincent J. Kruskal. Managing multi-version programs with an editor. IBM Journal of Research and Development, 28(1):74–81, 1984.

6 Vincent J. Kruskal. A blast from the past: Using p-edit for multidimensional editing. In Workshop on Multi-Dimensional Separation of Concerns in Software Engineering, 2000.

7 Anund Lie, Reidar Conradi, Tor Didriksen, Even-Andre Karlsson. Change oriented versioning in a software engineering database. SIGSOFT Softw. Eng. Notes, 14(7):56–65, 1989.

8 Bjørn P. Munch, Jens-Otto Larsen, Bjørn Gulla, Reidar Conradi. Uniform versioning: The Change-Oriented Model. Norwegian Institute of Technology, Trondheim, Norway, 1993.

3.8 Facing the Truth: Benchmarking the Techniques for the Evolution of Variant-Rich Systems

Daniel Strüber (Chalmers & University of Gothenburg, SE)

Joint work of Daniel Strüber, Mukelabai Mukelabai, Jacob Krüger, Stefan Fischer, Lukas Linsbauer, Jabier Martinez, Thorsten Berger
Main reference Daniel Strüber, Mukelabai Mukelabai, Jacob Krüger, Stefan Fischer, Lukas Linsbauer, Jabier Martinez, Thorsten Berger: "Facing the Truth: Benchmarking the Techniques for the Evolution of Variant-Rich Systems," in 23rd International Systems and Software Product Line Conference – Volume A (SPLC'19), September 9–13, 2019, Paris, France. ACM, New York, NY, USA, 12 pages.
URL https://doi.org/10.1145/3336294.3336302

The evolution of software systems in general, and of variant-rich systems in particular, is a challenging task. Many techniques have been proposed in the literature to support developers during software evolution. To advance such techniques and support their adoption, it is crucial to evaluate them against realistic baselines, ideally in the form of generally accessible benchmarks. To this end, we need to improve our empirical understanding of typical evolution scenarios for variant-rich systems and their relevance for benchmarking, and identify gaps in the existing benchmarking landscape. In this work, we establish eleven evolution scenarios in which benchmarks would be beneficial. Our scenarios cover typical lifecycles of variant-rich system, ranging from clone & own to adopting and evolving a configurable product-line platform. For each scenario, we formulate requirements for benchmarking the corresponding techniques. To assess the clarity and relevance of our scenarios, we conducted a community survey with software variability and evolution experts. We also surveyed the existing benchmarking landscape, identifying synergies and gaps with respect to our scenarios. We observed that most scenarios, despite being perceived as important by researchers, are only partially or not at all supported by existing benchmarks–a call to arms for building community benchmarks upon our requirements. We hope that our work raises awareness for benchmarking as a means to advance techniques for evolving variant-rich systems, and that it will lead to a benchmarking initiative in our community.

3.9 Semantic Slicing of Software Version Histories

Yi Li (Nanyang Technological University, SG)

Joint work of Yi Li, Chenguang Zhu, Julia Rubin, Marsha Chechik
Main reference Yi Li, Chenguang Zhu, Julia Rubin, Marsha Chechik: "Semantic Slicing of Software Version Histories", IEEE Trans. Software Eng., Vol. 44(2), pp. 182–201, 2018.
URL http://dx.doi.org/10.1109/TSE.2017.2664824

Software developers often need to transfer functionality, e.g., a set of commits implementing a new feature or a bug fix, from one branch of a configuration management system to another. That can be a challenging task as the existing configuration management tools lack support for matching high-level, semantic functionality with low-level version histories. The developer thus has to either manually identify the exact set of semantically-related commits implementing the functionality of interest or sequentially port a segment of the change history, "inheriting" additional, unwanted functionality. In this talk, we tackle this problem by providing automated support for identifying the set of semantically-related commits implementing a particular functionality, which is defined by a set of tests. We present two approaches, CSER and DEFINER, in a specific implementation for Java projects managed in Git and evaluate its correctness and effectiveness on a set of open-source software repositories. We show that it allows to identify subsets of change histories that maintain the functionality of interest but are substantially smaller than the original ones.

3.10 VariantSync – Automating the Synchronisation of Software Variants

Thomas Thüm (Technische Universität Braunschweig, DE)

Joint work of Thomas Thüm, Timo Kehrer

Today's software is often released in multiple variants to meet all customer requirements. Software product lines have the potential to decrease development costs and time-to-market, and have been actively researched for more than two decades. Nevertheless, practitioners frequently rely on ad hoc reuse based on a principle which is known as clone & own, where new variants of a software family are created by copying and adapting an existing variant. However, if a critical number of variants is reached, their maintenance and evolution becomes impractical, if not impossible, and the migration to a product line is often infeasible. With the research conducted in VariantSync, we aim to enable a fundamentally new development approach which bridges the gap between clone & own and product lines, combining the minimal overhead of clone & own with the systematic handling of variability of software product lines in a highly flexible methodology. The key idea is to transparently integrate the central product-line concept of a feature with variant management facilities known from version control systems in order to automatically synchronize a set of evolving variants. Lifting the underlying techniques employed by version control systems to the abstraction level of features which are shared among variants is an open problem and the main research challenge addressed in VariantSync. We believe that our research results have the potential to effectively change the way how practitioners will develop multi-variant software systems for which it is hard to foresee which variants will be added or released in the future.

3.11 Transformations of Software Product Lines: A Generalizing Framework Based on Category Theory

Gabriele Taentzer (Philipps-University Marburg, DE)

Joint work of Gabriele Taentzer, Rick Salay, Daniel Strüber, Marsha Chechik
Main reference Gabriele Taentzer, Rick Salay, Daniel Strüber, Marsha Chechik: "Transformations of Software Product Lines: A Generalizing Framework Based on Category Theory", in Proc. of the 20th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2017, Austin, TX, USA, September 17-22, 2017, pp. 101–111, IEEE Computer Society, 2017.
URL https://doi.org/10.1109/MODELS.2017.22

Software product lines are used to manage the development of highly complex software with many variants. In the literature, various forms of rule-based product line modifications have been considered. However, when considered in isolation, their expressiveness for specifying combined modifications of feature models and domain models is limited. In this talk a formal framework for product line transformations is presented that is able to combine several kinds of product line modifications presented in the literature. Moreover, it defines new forms of product line modifications supporting various forms of product lines and transformation rules. Our formalization of product line transformations is based on category theory, and concentrates on properties of product line relations instead of their single elements. This framework provides improved expressiveness and flexibility of software product line transformations while abstracting from the considered type of model.

3.12 Client-Specific Equivalence Checking

Julia Rubin (University of British Columbia, CA)

Joint work of Federico Mora, Yi Li, Julia Rubin, Marsha Chechik
Main reference Federico Mora, Yi Li, Julia Rubin, Marsha Chechik: "Client-specific equivalence checking", in Proc. of the 33rd ACM/IEEE International Conference on Automated Software Engineering, ASE 2018, Montpellier, France, September 3-7, 2018, pp. 441–451, ACM, 2018.
URL https://doi.org/10.1145/3238147.3238178

Software is often built by integrating components created by different teams or even different organizations. With little understanding of changes in dependent components, it is challenging to maintain correctness and robustness of the entire system. In this talk, we discuss the effect of component changes on the behavior of their clients. We show that changes in a component are often irrelevant to a particular client and thus can be adopted without any delays or negative effects. Following this observation, we formulate the notion of client-specific equivalence checking and develop an automated technique optimized for checking such equivalence. We evaluate our technique on a set of benchmarks, including those from the existing literature on equivalence checking, and show its applicability and effectiveness.

4 Lightning Talks

4.1 Versioning ML Models & Data in Time and Space

Shurui Zhou (Carnegie Mellon University, US)

Machine Learning (ML) technology has been widely used in a great number of applications. It is common that data scientists test different ML models, hyperparameters, configuration options, and so on, aiming to find the best configurations for the ML task. However, in order to easily track the process and compare results among different experimental settings, data scientists need to manually log different versions of the ML models, configurations, and so on, which is inefficient and error-prone. Besides, current version control techniques in the software-engineering domain, such as git, could not fulfill the requirement for ML-related tasks because of the scale and formats of data beyond only source code. Therefore, we aim to understand the problems in the ML domain and how versioning and variation mechanisms can support ML-related tasks and help data scientists to work and collaborate more efficiently in the future.

4.2 Towards Variability Mining Across Artifacts with Round-Trip Engineering

Sandro Schulze (Otto-von-Guericke University Magdeburg, DE)

This talk discusses some ideas about unifying the information of variability mining from different kind of artifacts. The typical semantics of clone & own are discussed along with the resulting challenges during development and maintenance. These challenges include lack of support for bug fix propagation to different variants, decentralization of information, risk of loss of information for new features, variation points not being expit and redundancy of efforts. A potential solution is to mine variability related information from artifacts of interest and extracting features and variability from those artifacts.These artifacts can be related to code, design or requirements etc. In addition, requirements can be used to create links between the other artifacts. The benefit of this is that knowledge on variability can be propagated. However, the challenge is that requirements are specified in natural language, which might be prone to inaccuracies and in-consistencies. This talk addresses this problem by proposing a similarity based natural language processing technique for variability mining in various software artifacts.

References

1 Anh Nguyen Duc, Audris Mockus, Randy L. Hackbarth, and John Douglas Palframan, "Forking and Coordination in Multi-platform Development: A Case Study", In *ESEM*, 2014.

2 Thorsten Berger, Divya Nair, Ralf Rublack, Joanne M. Atlee, Krzysztof Czarnecki, and Andrzej Wasowski, "Three Cases of Feature-Based Variability Modeling in Industry", In *MODELS*, 2014.

3 Stefan Stanciulescu, Sandro Schulze, and Andrzej Wasowski, “Forked and Integrated Variants In An Open-Source Firmware Project”, In *ICSME*, 2015.

4.3 Behavior-Derived Variability Analysis and Mechanisms Recommendation

Iris Reinhartz-Berger (University of Haifa, IL)

Software reuse, the use of existing artifacts in order to produce new software, has many benefits, including increased productivity, reduced costs and time-to-market, and improved quality. In cases of similarly behaving systems, reuse is more challenging, but yet profitable. In this presentation, two projects were introduced. SOVA–Semantic and Ontological Variability Analysis–proposes to combine semantic and ontological considerations that reflect system behavior, rather than implementation, as manifested in different development artifacts (e.g., requirements, test cases, and code). The approach includes behavior extraction, similarity calculation, and variability analysis, producing feature diagrams which depict the perceived variability in the system behaviors. More about SOVA can be found online [1]. VarMeR–Variability Mechanisms Recommendation–supports the analysis, visualization, and recommendation of behavior-derived software reuse based on existing product artifacts. This is done by considering polymorphism-inspired variability mechanisms: parametric, subtyping, and overloading. The outcomes are represented as a multi-layer similarity graph, where the nodes are reusable subjects (products, packages, classes, behaviors), and the edges recommend on potential reuse options in the form of the considered variability mechanisms. More about VarMeR can be found online [2].

References

1 SOVA, https://sites.google.com/is.haifa.ac.il/sova
2 VarMeR, https://sites.google.com/is.haifa.ac.il/varmer

4.4 Checking Feature Interaction and Code Integration Conflicts

Paulo Borba (Federal University of Pernambuco, BR)

Joint work of Paulo Borba, Roberto Barros, Léuson Silva, Rodrigo Bonifácio, Guilherme Cavalcanti, Rafael Mota, João Pedro Moisakis

In this talk we explore the similarities between semantic code integration conflicts and unintended, application-specific feature interaction. By relating the notions of feature and code contribution (changes submitted by a developer), and of interaction and interference, we propose to leverage semantic conflict detection techniques for checking feature interaction. As an initial step, we study the potential of using information flow analysis for detecting semantic conflicts when integrating developers contributions. The overall idea is to detect information flow between developers’ contributions. We discuss initial promising results and a number of challenges involved.

4.5 Towards a Feature-Oriented and Distributed Version Control System

Lukas Linsbauer (Johannes Kepler University Linz, AT)

This work relates to several areas of research that use different concepts to achieve similar goals, such as Version Control Sytems (VCS) and Software Configuration Management, Clone & own and Product Lines, and Traceability. VCS keep track of revisions (sequential versions, such as bug fixes) and facilitate collaboration among developers. It is also the responsibility of VCS to manage variants (concurrent versions, such as customer-specific variants). However, current VCS provide insufficiently coarse mechanisms (such as branches, which are essentially clones). Fine-grained variant management (based on individual features that trace to specific implementation artifacts) requires additional (often artifact type specific) mechanisms (such as simple preprocessors for text files or more complex product line platforms). We argue that feature-based variant management is needed to not only retrieve versions that have been explicitly stored (extensional versioning), but also to configure a system based on features and retrieve combinations of features that have not explicitly been stored (intensional versioning). We believe that such feature-based development is even better suited for distributed development than branch-based development (as is evidenced by the extensive use of feature-branches in current practice). Moreover, we observe that systems usually consist of different types of implementation artifacts in addition to source code which is why a generic variability mechanism is needed that is not limited to a specific type of artifacts. This work aims to unify related concepts of the above mentioned areas of research to combine their advantages while avoiding their drawbacks. The goal is to enable a novel feature-oriented and distributed development workflow. We therefore design, implement and evaluate a Feature-Oriented and Distributed Version Control System that supports both extensional and intensional versioning via revision and variant management mechanisms based on features for heterogeneous types of implementation artifacts.

References

1 ECCO, https://jku-isse.github.io/ecco.

5 Breakout Groups

5.1 Analysis Group

Members

Goetz Botterweck, Timo Kehrer, Mukelabai Mukelabai, Ina Schaefer, Klaus Schmid, Leopoldo Teixeira, Thomas Thüm, Mahsa Varshosaz, and Eric Walkingshaw.

5.1.1 Motivation and Goals

There are numerous analyses to cope with variation in space (i.e., product-line analyses) and others that cope with variation in time (i.e., regression analyses). While both kinds of techniques have developed largely independent of each other, the common idea is to exploit the similarities between variants (product-line analysis) and versions (regression analysis) in order to save analysis efforts and to avoid doing entire analysis from scratch. To that

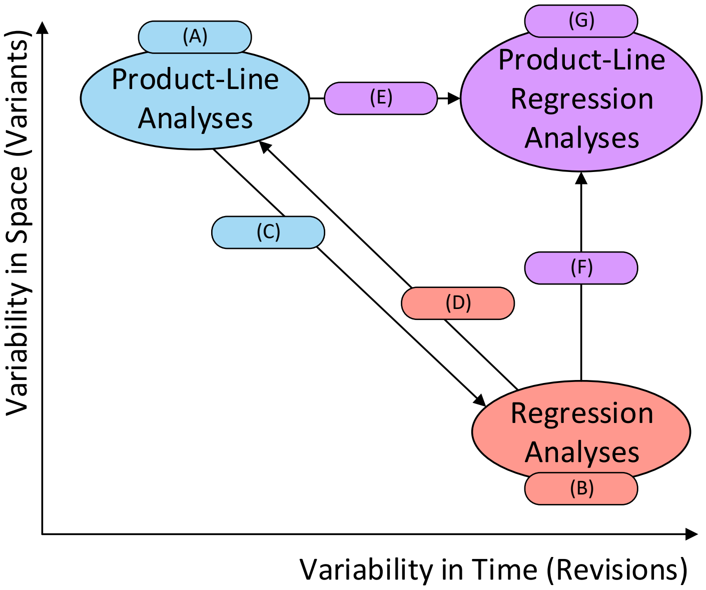
(A)
Product-Line Analyses
(E)
(G)
Product-Line Regression Analyses
(C)
(D)
(F)
Regression Analyses
(B)
Variability in Space (Variants)
Variability in Time (Revisions)

Figure 1 Efficient analyses for two dimensions of variability.

end, in this breakout group, we discussed to which extent product-line analyses can be applied to revisions and, conversely, where regression analyses can be applied to variants. In addition, we discussed challenges related to the combination of product-line and regression analyses. The overall goal is to increase the efficiency of analyses by exploiting the inherent commonality between variants and revisions.

5.1.2 Summary of Results and Open Challenges

An overview of the course of our discussions during the seminar and the corresponding results is shown in Figure 1. It structures analysis techniques along the two dimensions of variability, i.e., variability in time (leading to revisions) and variability in space (leading to variants).

Our starting point was to recall the basic nature of classical product-line analysis (A), devoted to the analysis of variants [9, 8, 6], and regression analysis (B), designed to efficiently analyze revisions [10, 4, 2, 1, 3]. Next, we discussed the application of traditional techniques in both directions, that is, (C) the application of product-line analyses to revisions and (D) the application of regression analyses to variants. In fact, there is a lot of potential for re-using product-line analyses for revisions and for re-using regression analyses for variants, some of which has been already exploited [5, 4]. Finally, we discussed how product-line analyses and regression analyses can be applied to both dimensions of variability. As shown in Figure 1, such product-line regression analyses can be realized by (E) applying product-line analyses to revisions of variants, (F) applying regression analyses to revisions of variants, or (G) combinations of product-line analyses and regression analyses.

As a result of our survey of existing analysis techniques, we identified a couple of challenges and promising directions for future research. In particular, the application of product-line analyses to variation in time seems to be a new application area for many existing product-line analyses. That is, research results of the product-line community could be reused by communities working on regression analyses. While regression analyses have often been applied to variation in space, we also summarized common challenges for their application to variants. With product-line regression analysis, we denote analyses that cope with variation in both dimensions, namely time and space. For that purpose, two-dimensional lifting of traditional analyses is necessary with respect to both dimensions of variation. We believe that it requires a community effort to identify which strategies for lifting lead to the most efficient analyses, perhaps also paving the way for an integration of multiple strategies.

5.1.3 Further Reading

A more thorough discussion of our literature survey as well as the identified challenges towards efficient analysis of variation in time and space can be found in Thüm et al. [7], a paper that has been written by the breakout group members after the Dagstuhl seminar and that has been recently accepted at the VariVolution workshop at SPLC 2019.

References

1 Steven Arzt and Eric Bodden. Reviser: Efficiently Updating IDE-/IFDS-based Data-flow Analyses in Response to Incremental Program Changes. In *ICSE*, 2014.

2 Larissa Braz, Rohit Gheyi, Melina Mongiovi, Márcio Ribeiro, Flávio Medeiros, Leopoldo Teixeira, and Sabrina Souto. A Change-Aware Per-File Analysis to Compile Configurable Systems with #ifdefs. *Computer Languages, Systems & Structures*, 54:427–450, 2018.

3 Benny Godlin and Ofer Strichman. Regression Verification. In *DAC*, 2009.

4 Wolfgang Heider, Rick Rabiser, Paul Grünbacher, and Daniela Lettner. Using Regression Testing to Analyze the Impact of Changes to Variability Models on Products. In *SPLC*, 2012.

5 Sascha Lity, Malte Lochau, Ina Schaefer, and Ursula Goltz. Delta-oriented model-based SPL regression testing. In *PLEASE*, 2012.

6 Mukelabai Mukelabai, Damir Nešić, Salome Maro, Thorsten Berger, and Jan-Philipp Steghöfer. Tackling combinatorial explosion: A study of industrial needs and practices for analyzing highly configurable systems. In *ASE*, 2018.

7 Thomas Thüm, Leopoldo Teixeira, Klaus Schmid, Eric Walkingshaw, Mukelabai Mukelabai, Mahsa Varshosaz, Goetz Botterweck, Ina Schaefer, and Timo Kehrer. Towards efficient analysis of variation in time and space. In *VaMoS*, 2019.

8 Thomas Thüm, Sven Apel, Christian Kästner, Ina Schaefer, and Gunter Saake. A Classification and Survey of Analysis Strategies for Software Product Lines. ACM Comput. Surv. 47(1):6:1–6:45, 2014.

9 Alexander von Rhein, Sven Apel, Christian Kästner, Thomas Thüm, and Ina Schaefer. The PLA Model: On the Combination of Product-Line Analyses. In *VaMoS*, 2013.

10 Shin Yoo and Mark Harman. Regression Testing Minimization, Selection and Prioritization: A Survey. Softw. Test., Verif. Reliab. 22(2):67–120, 2012.

5.2 Conceptual Modeling Group

Members

Sofia Ananieva, Thorsten Berger, Andreas Burger, Timo Kehrer, Heiko Klare, Anne Koziolek, Henrik Lönn, Ramesh Sethu, Gabriele Taentzer, and Bernhard Westfechtel.

5.2.1 Motivation and Goals

SCM and SPLE are two widely established yet actively researched software engineering disciplines offering a variety of concepts to deal with software variability in time and space.

Research on SCM has proposed versioning models which define the artifacts to be versioned as well as the way in which these artifacts are organized, identified and composed to configurations [4]. Nowadays version control systems such as Subversion [9] or Git [7] are file-based, organizing versions of files in a directed acyclic version graph. Variants of a software artifact or an entire software system are represented by parallel development branches, where each of these branches has its own chronological evolution. Instead of managing variants (a.k.a. products) as clones in parallel branches, SPLE advocates to create a product-line platform that integrates all the product features and contains explicit variation points realized using variability mechanisms such as conditional compilation or element exclusion [5, 3]. However, neither research community has been successful in producing unified management techniques that are effective in practice.

As a step towards overcoming this unfortunate situation, the goal of this breakout group was to conceive a conceptual yet integrated model of SCM and SPLE concepts. This provides discussion grounds for a wider exploration of a unified methodology supporting software evolution in both time and space. The value and possible usage scenarios of a conceptual model are twofold. It may be instantiated to characterize and classify existing approaches, to structure the state-of-the-art and to map and align both communities' core concepts. It may also pinpoint open issues and serve as a vehicle for evaluating different integration strategies on a high-level of abstraction.

5.2.2 Summary of Results

In Figure 2, we present a basic conceptual model of variability in time and space, which was developed during the seminar. It is composed of three differently colored parts corresponding to (i) concepts for variability in time (blue), (ii) concepts for variability in space (green), and (iii) concepts common to both (red). Clearly, although the SCM and SPLE communities have developed largely independently of each other, they share a set of common concepts, notably the idea of composing a system from fragments which serve as units of versioning and as re-usable assets, respectively. These common concepts of system descriptions served as a starting point of our discussion on a conceptual model, before we explored those concepts which we consider to be specific to one of the disciplines. As for variability in time, the key concept of a revision is applied to each fragment, and a sequence of revisions related through predecessor and successor relationships represents the chronological evolution of a fragment. As for variability in space, the key idea is to define a set of explicit variation points which may be bound to concrete fragments in order to instantiate a product.

Finally, we elaborated on an idea of how those concepts could be combined for managing variability in time and space. Figure 3 represents an extension to the introduced model and depicts the proposed integration of variability in space and time. In essence, integration is achieved through the concept of a versioned item, which represents a higher-level versioning of the introduced concepts by putting them under revision control. In this sense, the versioned item acts as a super class for the fragment, for the variation point and for the product line itself.

5.2.3 Discussion and Future Work

To validate that our model is *general* and *appropriate* in the sense that we are able to map its elements to actual approaches for describing such variability, we will apply the model to existing approaches, such as Ecco [6], SuperMod [10], DeltaEcore [11] or SiPL [8] in future work. Several design decisions in the conceptual model were subject to intensive discussion and may be validated when instantiating the model in future work.

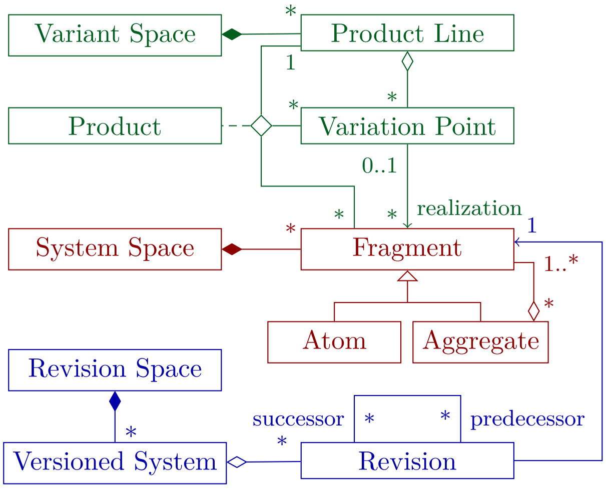

Figure 2 A basic conceptual model of variability in time (blue), variability in space (green), and shared concepts (red).

One central subject of discussion is whether branches in revision control systems are a concept of variability in time to support temporary divergence for concurrent development, or whether they represent a realization of variability in space, as they support the existence of products at the same point in time. For the time being, we chose to follow the former option.

Another subject of discussion which requires future validation is whether or not to consider the product a subclass of the *versioned item*. According to Antkiewicz et al. [2], product derivation is either fully automated or followed by manual post-processing (corresponding to the so-called *governance levels* L5 and L6). In the case of fully automated product derivation (L6), a product represents a fully derived artifact for which revision control becomes superfluous since the product line is already put under revision control in the extended model. When manual post-processing takes place (L5), a product does not represent a fully derived artifact anymore for which revision control becomes reasonable again.

Additionally, the semantics of several concepts is only defined through the mechanisms that operate on them. For example, for the configuration of a product from a product line, variation points and fragments are expressed in our model, but constraints that define which variation points and fragments may be selected have to be ensured by a configuration mechanism. The same applies to the unifying concept of our extended model. To define what the relations between revisions of product lines, variation points and fragments are, a mechanism that defines how they can be combined has to be defined. Designing such a mechanism, based on the presented model, should be the next step towards a unifying concept for variability in space and time.

5.2.4 Further Reading

A more detailed description of our conceptual model can be found in Ananieva et al. [1], a paper which has been written by the breakout group members after the Dagstuhl seminar and which has been recently accepted at the VariVolution workshop at SPLC 2019. The paper also provides an example instantiation of our conceptual model and a more thorough discussion of related work.

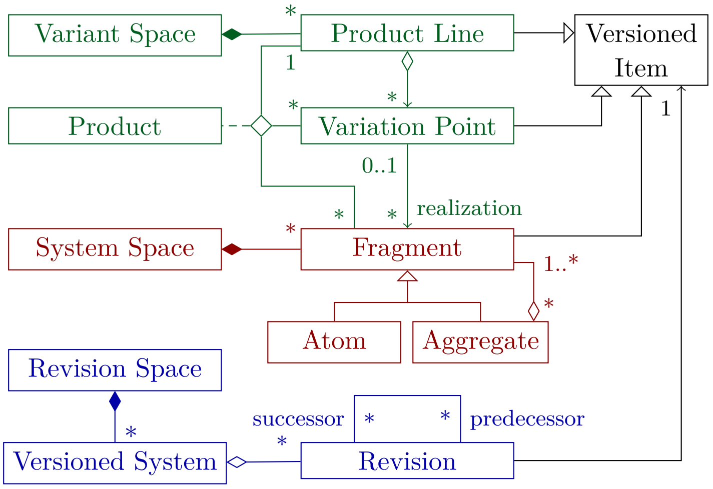

Figure 3 An extended conceptual model (w.r.t. Figure 2) for combining concepts of variability in space and time.

References

1 Sofia Ananieva, Timo Kehrer, Heiko Klare, Anne Koziolek, Henrik Lönn, Ramesh Sethu, Andreas Burger, Gabriele Taentzer, and Bernhard Westfechtel. Towards a conceptual model for unifying variability in space and time. In *VaMOS*, 2019.

2 Michał Antkiewicz, Wenbin Ji, Thorsten Berger, Krzysztof Czarnecki, Thomas Schmorleiz, Ralf Lämmel, Stefan Stănciulescu, Andrzej Wąsowski, and Ina Schaefer. Flexible product line engineering with a virtual platform. In *ICSE*, 2014.

3 Paul Clements and Linda Northrop. *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2001.

4 Reidar Conradi and Bernhard Westfechtel. Version models for software configuration management. *ACM Comput. Surv.*, 30(2):232–282, June 1998.

5 Krzysztof Czarnecki and Ulrich W. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, 2000.

6 Stefan Fischer, Lukas Linsbauer, Roberto E. Lopez-Herrejon, and Alexander Egyed. The ECCO tool: Extraction and composition for clone-and-own. In *ICSE*, 2015.

7 Jon Loeliger and Matthew McCullough. *Version Control with Git: Powerful tools and techniques for collaborative software development*. O'Reilly Media, Inc., 2012.

8 Christopher Pietsch, Timo Kehrer, Udo Kelter, Dennis Reuling, and Manuel Ohrndorf. SIPL–a delta-based modeling framework for software product line engineering. In *ASE*, 2015.

9 C. Michael Pilato, Ben Collins-Sussman, and Brian W. Fitzpatrick. *Version Control with Subversion: Next Generation Open Source Version Control*. O'Reilly Media, Inc., 2008.

10 Felix Schwägerl and Bernhard Westfechtel. Supermod: Tool support for collaborative filtered model-driven software product line engineering. In *ASE*, 2016.

11 Christoph Seidl, Ina Schaefer, and Uwe Aßmann. Integrated management of variability in space and time in software families. In *SPLC*, 2014.

5.3 Workflow Group

Members

Don Batory, Danilo Beuche, Paulo Borba, Paul Grünbacher, Jacob Krüger, Ralf Lämmel, Lukas Linsbauer, Sarah Nadi, Iris Reinhartz-Berger, Sandro Schulze, Stefan Stanciulescu, Daniel Strüber, Shurui Zhou, and Andrzej Wasowski.

5.3.1 Scope

The task of the group was to investigate the role of workflows in the context of the general seminar theme of unifying version and variability management. An initial discussion revealed that there are many different workflows, and defining a general workflow that would work in different domains would not be useful. For instance, several participants discussed the differences between open source software (OSS) and software development processes in industry. Also, participants pointed out the need for workflows supporting both developers/teams working on single platforms as well as workflows guiding the interactions and exchange between multiple development platforms, for instance, in the context of software ecosystems. The discussion also showed that there are many good reasons to create forks ("clone & own engineering"), which also requires specific workflows. For instance, in such a context, awareness becomes particularly important to understand what's happening in the different clones.

However, despite these diverse characteristics, the discussion showed that the different workflows share common elements ("operations"). The group continued with identifying candidate operations as elements of workflow. Examples are: Create Clone, Add Feature to Clone, Create Pull Request, Accept Pull Request, Modify Feature Implementation, or Checkout Configuration to name but a few.

A better understanding of these operations as building blocks for workflows is essential, so the group proceeded with a scenario-based approach.

5.3.2 Scenario-Based Investigation of Two Workflows

The group investigated two real-world workflows more closely by defining them in terms of scenarios to better understand their characteristics. The first scenario covers an OSS workflow about the typical development process of a new feature in Marlin. It was defined based on a paper by workgroup member Jacob Krüger et al [5]. The second scenario describes a typical industrial workflow. It was defined based on the seminar talk and additional information by workgroup member Danilo Beuche. For instance, here are examples of steps from the OSS workflow scenario:

- Alice clones the platform (not a variant) to create a feature
- Alice implements the feature in the clone. She creates a number of commits in her fork. The changes are marked as belonging to the feature.
- Alice creates a pull request, which checks the integration by checking for feature interactions, dead code, feature model inconsistencies, and so on. During the check, a "conflict" appears.
- Alice syncs with upstream (that is, she pulls) and resolves the conflict. This may include resolving inconsistencies in the feature model, make all code alive again, and so on.
- Bob reviews the pull request. He confirms that the commit satisfies the given requirements and that the feature has the right level of granularity, whether it is placed in the right place in the model, whether it has correct dependencies.

- An automated test is performed by the continuous integration (CI) system. Tests are run on the maximum and minimum configuration with the feature enabled and disabled. Cynthia assesses Bob's review and the CI results. She decides to merge the pull request in.

After defining the scenarios the team mapped the candidate operations to the different scenario steps. The purpose of this mapping process was to better understand the scope and purpose of the different operations and to discuss their properties with respect to the seminar topic of integrating version and variability management. In particular, the operations were refined and "feature-ized" by defining signatures. For example: CreateClone (in: existing platform; out: copy of the existing platform), AddFeature (in: featureName, in: featureDependencies), CommitToFeature (in: commits; in: presCond; in: repository).

5.3.3 Challenges and Next Steps

Finally, the group discussed how these workflows and operations could possibly be implemented. The group explored alternatives such as extending an existing version control system like Git; extending a Variation Control System (ECCO, VTS, and so on); or the development of a new system from scratch.

The discussion about future work already covered confirming and evaluating the workflows and operations. There was an agreement to study more industrial workflows, different OSS workflows, and workflows from other domains. In this regard participants raised the issue of the high cognitive complexity of the envisioned workflows. Finally, the team discussed possible benchmarks and case study systems. TurtleBot (TUB) and the Pick-and-Place Unit (TUM, JKU) were named as examples.

For further readings, we refer to the following list of references.

References

1 Michal Antkiewicz, Wenbin Ji, Thorsten Berger, Krzysztof Czarnecki, Thomas Schmorleiz, Ralf Lämmel, Stefan Stanciulescu, Andrzej Wąsowski, and Ina Schaefer. Flexible product line engineering with a virtual platform. In *ICSE*, 2014.

2 Daniel Hinterreiter, Lukas Linsbauer, Florian Reisinger, Herbert Prähofer, Paul Grünbacher, and Alexander Egyed. Feature-oriented evolution of automation software systems in industrial software ecosystems. In *ETFA*, 2018.

3 Wenbin Ji, Thorsten Berger, Michal Antkiewicz, and Krzysztof Czarnecki. Maintaining feature traceability with embedded annotations. In *SPLC*, 2015.

4 Jing Jiang, David Lo, Jiahuan He, Xin Xia, Pavneet Singh Kochhar, and Li Zhang. Why and how developers fork what from whom in github. *Empirical Software Engineering*, 22(1):547–578, 2017.

5 Jacob Krüger, Mukelabai Mukelabai, Wanzi Gu, Hui Shen, Regina Hebig, and Thorsten Berger. Where is my feature and what is it about? A case study on recovering feature facets. *Journal of Systems and Software*, 152:239–253, 2019.

6 Lukas Linsbauer, Thorsten Berger, and Paul Grünbacher. A classification of variation control systems. In *GPCE*, 2017.

7 Lukas Linsbauer, Alexander Egyed, and Roberto Erick Lopez-Herrejon. A Variability-Aware Configuration Management and Revision Control Platform. In *ICSE*, 2016.

8 Lukas Linsbauer, Roberto Erick Lopez-Herrejon, and Alexander Egyed. Variability extraction and modeling for product variants. *Software and System Modeling*, 16(4):1179–1199, 2017.

9 Rachel Potvin and Josh Levenberg. Why google stores billions of lines of code in a single repository. *Commun. ACM*, 59(7):78–87, 2016.

10 Baishakhi Ray and Miryung Kim. A case study of cross-system porting in forked projects. In *FSE*, 2012.

11 Baishakhi Ray, Christopher Wiley, and Miryung Kim. REPERTOIRE: a cross-system porting analysis tool for forked software projects. In *FSE*, 2012.

12 Luyao Ren, Shurui Zhou, Christian Kästner, and Andrzej Wąsowski. Identifying redundancies in fork-based development. In *SANER*, 2019.

13 Thomas Schmorleiz and Ralf Lämmel. Similarity management of 'cloned and owned' variants. In *SAC*, 2016.

14 Stefan Stanciulescu, Thorsten Berger, Eric Walkingshaw, and Andrzej Wasowski. Concepts, operations, and feasibility of a projection-based variation control system. In *ICSME*, 2016.

15 Stefan Stanciulescu, Sandro Schulze, and Andrzej Wasowski. Forked and integrated variants in an open-source firmware project. In *ICSME*, 2015.

16 Shurui Zhou, Ştefan Stănciulescu, Olaf Leßenich, Yingfei Xiong, Andrzej Wasowski, and Christian Kästner. Identifying features in forks. In *ICSE*, 2018.

5.4 Languages Group

Members

Don Batory, Marsha Chechik, Shahar Maoz, Julia Rubin, Christoph Seidl, and Manuel Wimmer.

5.4.1 Motivation and Goals

This breakout group investigated the idea of having first-class language support for features and variants as well as for versioning. Currently, these aspects are mostly managed outside the base programming and modeling languages, which may come with several drawbacks. For instance, the intention of a feature is likely to be lost, e.g., by generic preprocessor directive, tools that want to perform analyses or changes have to spend a lot of time on deciphering the (company/project specific) conventions how a feature is represented within or for a particular language.

Having first-class language support may allow for immediate feedback of feature effects as well as may simplify tool building if there is a unique and formally defined concept provided by the base language.

5.4.2 Summary of Results and Open Challenges

Language support for Features/Variants. First, the group investigated existing work which goes into the direction of having first-class language support for features and variants. For instance, Matlab/Simulink provides the modeling concept "Variant Subsystems" which is somewhat similar to features where the code generator may disregard parts of the model.

Second, the group investigated different possibilities of feature concepts for programming languages where a specific focus was on Java as example language, but also other options going beyond Java have been considered to be more general. In the following, a summary of different options is provided.

- With preprocessor-like comments such as supported by Antenna, one is very flexible which elements are affected by a feature, but the intention of the feature is lost.
- With if statements, one is limited which elements a feature may affect. Again, the intention of the feature is lost and the variability is pushed to runtime.

- Java annotations would allow to maintain the intention of a feature, but may still be somewhat limited in which elements they can affect (although in newer versions of Java, several improvements have been provided with respect to annotation targets).
- Partial classes as provided for instance in C#, would allow to define 150% models. The solution would be flexible in which elements it can affect, however, the intention of a feature is not explicitly maintained. Furthermore, the solution would work well for optional features, but alternatives would be harder to realize as all partial classes are combined by the compiler.
- Syntax and compiler extensions as provided by Scala or Groovy would allow to build conservative extensions of languages in order to be backward compatible. New keywords may be introduced for defining features and variants which may be used in a flexible manner with respect to which elements they can affect. The intention of a feature is maintained without having to maintain a dialect of the base language. However, it seems politically complicated to get such extensions into the core language specifications in near future.

As a potential future work, investigating the usage of annotations for introducing feature and variant concepts to existing languages seems promising as a first step. A concrete outcome should be the creation and population of a library of annotations that can be enforced, analyzed, and applied by existing IDEs. As a second step, interactions with the community of a language may be necessary to reach consensus about the concepts and their integration in the base language such as assembling Java Specification Request (JSR) in the case of Java.

Finally, we discussed if there should be a "required interface" for features, which specifies the feature realized in a language element. For instance, if a class is realizing two features, one may provide the following explicit syntax:

```
class A realizes FeatureA, FeatureB
```

Language Support for Versioning. The discussion group also investigated first-class language support for versioning of artifacts. Several use cases came up during the discussions: (*i*) trace changes for later manual inspection, (*ii*) suggest changes that need approval, (*iii*) maintain backward compatibility if it is of utmost importance, and (*iv*) negotiate "contracts" between service provider/user.

The group also investigated existing work which already proposes first-class language support for versioning. For instance, AutomationML comes with integrated versioning support [1]. Change-oriented programming by Peter Ebraert [2] advocates tracing changes to language artifacts but has no version support directly integrated with languages.

Finally, the group concluded–based on the discussion and previous experiences of the group members–that the use cases for programming languages are not so clear or pressing. However, the use cases for (system) modeling languages seem more relevant and here clear needs could be identified. Thus, future work in this area is anticipated.

References

1 Stefan Biffl, Emanuel Mätzler, Manuel Wimmer, Arndt Lüder, and Nicole Schmidt. Linking and versioning support for automationml: A model-driven engineering perspective. In *INDIN*, 2015.

2 Peter Ebraert, Jorge Vallejos, Pascal Costanza, Ellen Van Paesschen, and Theo D'Hondt. Change-oriented software engineering. In *ICDL*, 2007.

5.5 Operations Group

Members

Don Batory, Paulo Borba, Yi Li, Wardah Mahmood, Shahar Maoz, Sarah Nadi, Julia Rubin, Klaus Schmid, Christoph Seidl, and Manuel Wimmer.

5.5.1 Motivation and Goals

For building, using, and combining tools in order to deal with versions and variants of systems, a set of well-established and formalized operators is required. Previous work on database integration and evolution focused on the establishment of model management operators [1], which have a well-defined signature as well as clearly defined pre- and postconditions. In addition, a kernel set of artifact types on which they operate is defined.

For instance, let us introduce two different operators which may be of frequent use in versioning and variant scenarios and which build on each other:

```
Differencing Operator: diff(m1, m2) : diffModel
Patching Operator: patch(m2, diff(m0, m1)) : model
```

Based on such a set of operators, more complicated scenarios can be supported by orchestrating the operators to so-called model management scripts. Thus, workflows can be defined based on a set of basic building blocks. Similar support seems beneficial to manage artifacts in versioning and variant management systems or even a combination of it (cf. also the report of the workflow group for similar discussions).

5.5.2 Summary of Results and Open Challenges

When reasoning about operators for versioning and variant management, it becomes evident that such operators may act differently for a file (which is currently the main artifact type in versioning systems) than for a feature. Furthermore, current scenarios such as building a product that combines feature X in v1.1 and feature Y in v2.0 may be hard to achieve with the current version control system operations. Thus, the question came up if there are operators needed for variant management that need fundamental changes to version control systems. The discussion group agreed that this is not required, but instead there should be a layer on top of version control systems for variant management.

Another important aspect for variant management is that one wants to abstract from the file level. Imagine a user wants to look at the history in terms of feature edits (e.g., these three commits are actually changing feature X) instead of tracking changes in individual files. Another scenario is when checking-in, the version control system may ask if this change is related to feature A. Similar, when modifying feature A, one may want to know that no other features are affected.

The group also agreed that it is hard to define operations in isolation, because they highly depend on the variation mechanism and the concrete notion of feature as well as on the specific workflows one may want to achieve. Thus, as future work, the operations should be derived from the common workflows as discussed and reported by the workflow group.

References

1 Sergey Melnik, Erhard Rahm, and Philip A. Bernstein. Rondo: A programming platform for generic model management. In *SIGMOD*, 2003.

Participants

- Sofia Ananieva
FZI – Berlin, DE
- Sven Apel
Universität des Saarlandes, DE
- Don Batory
University of Texas – Austin, US
- Thorsten Berger
Chalmers and University of Gothenburg, SE
- Danilo Beuche
pure-systems GmbH – Magdeburg, DE
- Paulo Borba
Federal University of Pernambuco – Recife, BR
- Götz Botterweck
University of Limerick, IE
- Andreas Burger
ABB – Ladenburg, DE
- Marsha Chechik
University of Toronto, CA
- Paul Grünbacher
Johannes Kepler Universität Linz, AT
- Timo Kehrer
HU Berlin, DE
- Heiko Klare
KIT – Karlsruher Institut für Technologie, DE
- Anne Koziolek
KIT – Karlsruher Institut für Technologie, DE
- Jacob Krüger
Universität Magdeburg, DE
- Ralf Lämmel
Facebook – London, GB
- Yi Li
Nanyang TU – Singapore, SG
- Lukas Linsbauer
Johannes Kepler Universität Linz, AT
- Henrik Lönn
Volvo – Göteborg, SE
- Wardah Mahmood
Chalmers University of Technology – Göteborg, SE
- Shahar Maoz
Tel Aviv University, IL
- Mukelabai Mukelabai
University of Gothenburg, SE
- Sarah Nadi
University of Alberta – Edmonton, CA
- Iris Reinhartz-Berger
Haifa University, IL
- Julia Rubin
University of British Columbia – Vancouver, CA
- Ina Schaefer
TU Braunschweig, DE
- Klaus Schmid
Universität Hildesheim, DE
- Sandro Schulze
Universität Magdeburg, DE
- Christoph Seidl
TU Braunschweig, DE
- Ramesh Sethu
General Motors – Warren, US
- Stefan Stanciulescu
ABB – Baden-Dättwil, CH
- Daniel Strüber
Chalmers University of Technology – Göteborg, SE
- Gabriele Taentzer
Universität Marburg, DE
- Leopoldo Teixeira
Federal University of Pernambuco – Recife, BR
- Thomas Thüm
TU Braunschweig, DE
- Mahsa Varshosaz
IT University of Copenhagen, DK
- Eric Walkingshaw
Oregon State University – Corvallis, US
- Andrzej Wasowski
IT University of Copenhagen, DK
- Bernhard Westfechtel
Universität Bayreuth, DE
- Manuel Wimmer
Johannes Kepler Universität Linz, AT
- Shurui Zhou
Carnegie Mellon University – Pittsburgh, US

Report from Dagstuhl Seminar 19192

Visual Analytics for Sets over Time and Space

Edited by

Sara Irina Fabrikant[1], Silvia Miksch[2], and Alexander Wolff[3]

1 **Universität Zürich, CH,** sara.fabrikant@geo.uzh.ch
2 **TU Wien, AT,** miksch@ifs.tuwien.ac.at
3 **Universität Würzburg, DE, https://orcid.org/0000-0001-5872-718X**

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19192 "Visual Analytics for Sets over Time and Space", which brought together 29 researchers working on visualization (i) from a theoretical point of view (graph drawing, computational geometry, and cognition), (ii) from a temporal point of view (visual analytics and information visualization over time, HCI), and (iii) from a space-time point of view (cartography, GIScience). The goal of the seminar was to identify specific theoretical and practical problems that need to be solved in order to create dynamic and interactive set visualizations that take into account time and space, and to begin working on these problems.

The first 1.5 days were reserved for overview presentations from representatives of the different communities, for presenting open problems, and for forming interdisciplinary working groups that would focus on some of the identified open problems as a group. There were three survey talks, ten short talks, and one panel with three contributors. The remaining three days consisted of open mic sessions, working-group meetings, and progress reports. Five working groups were formed that investigated several of the open research questions. Abstracts of the talks and a report from each working group are included in this report.

Seminar May 5–10, 2019 – http://www.dagstuhl.de/19192
2012 ACM Subject Classification Human-centered computing → Visualization, Theory of computation → Design and analysis of algorithms
Keywords and phrases Geovisualization, graph drawing, information visualization, set visualization, visual analytics
Digital Object Identifier 10.4230/DagRep.9.5.31
Edited in cooperation with Philipp Kindermann

1 Executive Summary

Sara Irina Fabrikant
Silvia Miksch
Alexander Wolff

Seminar Goals

Increasing amounts of data offer great opportunities to promote technological progress and business success. Visual analytics aims at enabling the exploration and the understanding of large and complex data sets by intertwining interactive visualization, data analysis, human-computer interaction, as well as cognitive and perceptual science. Cartography has for thousands of years dealt with the depiction of spatial data, and more recently geovisual analytics researchers have joined forces with the visual analytics community to create visualizations to help people to make better and faster decisions about complex problems that require the analysis of big data.

Set systems comprise a generic data model for families of sets. A *set* is defined as a collection of unique objects, called the set elements, with attributes, membership functions, and rules. Such a complex data model asks for appropriate exploration methods. As with many types of data, set systems can vary over time and space. It is important, however, not to treat time and space as usual variables. Their special characteristics such as different granularities, time primitives (time points vs. intervals), hierarchies of geographic or administrative regions need to be taken into account. Visualizing and analyzing such changes is challenging due to the size and complexity of the data sets.

Sets systems can also be seen as hypergraphs where the vertices represent the ground elements and the edges are the sets. However, compared to conventional graphs that represent only binary relations (that is, sets with two elements), the visualization of general hypergraphs has received little attention. This is even more so when dealing with dynamic hypergraphs or hypergraphs that represent spatial information.

In this seminar, we aimed at bringing together researchers from the areas of visual analytics, information visualization and graph drawing, geography and GIScience, as well as cartography and (spatial) cognition, in order to develop a theory and visualization methods for set systems that vary over time and space.

Seminar Program

As the topic of the seminar was interdisciplinary and the participants had very different scientific backgrounds, we introduced the main themes of the seminar in three separate sections: “Sets in Time”, “Sets in Space”, and “Graph Drawing and Set Visualization”. Each section consisted of a survey talk and three to four short talks. The three sections were followed by a panel discussion. For the survey talks, we explicitly asked the presenters to give a balanced overview over their area (rather than to focus on their own scientific contributions).

On the second day of the seminar, we collected a number of challenging open problems. Then we formed five groups, each of which worked on a specific open problem for the remainder of the seminar. The work within the groups was interrupted only a few times; in order to share progress reports, listen to open-mic talks, and to discuss possible future activities. These plenary meetings helped to exchange the different visions of the working groups.

We now list the items of the program in detail.

1. Section “Sets in Time” (for abstracts, see Section 3)
   - Peter Rogers gave an excellent survey talk about techniques for visualizing sets over time. He illustrated possible challenges and opportunities in this research area.
   - Philipp Kindermann presented some results and open questions in simultaneous orthogonal graph drawing.
   - Wouter Meuleman talked about spatially and temporally coherent visual summaries.
   - Tamara Mchedlidze introduced a data-driven approach to quality metrics of graph visualizations.
   - Margit Pohl discussed perception considerations of space and time in cognitive psychology and their implications for the design of visualizations.

2. Section "Sets in Space" (for abstracts, see Section 4)
   - Sara Fabrikant gave an inspiring survey talk about space discussed from a cartographer‘s view.
   - Natalia Andrienko elaborated about evolving sets in space.
   - Somayeh Dodge discussed dynamic visualization of interaction in movement of sets.
   - Jan-Henrik Haunert introduced fast retrieval of abstracted representations for sets of points within user-specified temporal ranges.
3. Section "Graph Drawing and Set Visualization" (for abstracts, see Section 5)
   - André Schulz very nicely surveyed the area of drawing graphs and hypergraphs and sketched the main challenges in this area.
   - Michalis Bekos gave a short overview of graph drawing beyond planarity.
   - Sabine Cornelsen talked about general support for hypergraphs.
   - Martin Nöllenburg introduced plane supports for spatial hypergraphs.
4. The panel discussion was entitled "Visual Analytics for Sets over Time and Space: What are the burning scientific questions? An interdisciplinary perspective." André Skupin, Steven Kobourov, and Susanne Bleisch each gave a short statement about the central questions of his or her area; see Section 6. Afterwards we had a fruitful and interesting discussion, which led to a productive open problem session.
5. The working groups formed around the following open problems:
   - "Concentric Set Schematization",
   - "From Linear Diagrams to Interval Graphs",
   - "Thread Visualization",
   - "Clustering Colored Points in the Plane", and
   - "Flexible Visualization of Sets over Time and Space".

The reports of the working groups were collected by Michalis Bekos, Steven Chaplick, William Evans, Jan-Henrik Haunert, and Christian Tominski; see Section 7.

Future Plans

During our seminar, plans for a follow-up seminar were discussed in a plenary meeting. The seminar-to-be will aim at integrating the approaches for set visualization that have been taken by the different communities (geovisualization, information visualization, and graph drawing, including industry and research). Susanne Bleisch, Steven Chaplick, Jan-Henrik Haunert, and Eva Mayr are currently discussing the precise focus and a title to match that focus.

Among the 29 participants of the seminar, 24 participated in the survey that Dagstuhl does at the end of every seminar. Many answers were in line with the average reactions that Dagstuhl collected over a period of 60 days before our seminar (such as the scientific quality of the seminar, which received a median of 10 out of 11 – "outstanding"). A few questions, however, received different feedback. For example, due to the interdisciplinary nature of the seminar, we had more frequent Dagstuhl visitors than usually: a third of the participants of the survey had been to Dagstuhl at least seven times. It was also interesting to see that more participants than usually stated that our seminar had inspired new research ideas, joint projects or publications, that it had led to insights from neighboring fields, and that it had identified new research directions.

In spite of the organizers' attempt to have a diverse group of participants, all survey participants were from academia and only two rated themselves as "junior". Not surprisingly, some participants suggested to have more PhD students, more people from industry, and generally more people from applications rather than from (graph drawing) theory. The last free text comment in the survey reads: “Once again, a great week at Schloss Dagstuhl – thank you!”

Acknowledgments

We all enjoyed the unique Dagstuhl atmosphere. In particular, it was great to have the opportunity to use a separate room for each working group. We thank Philipp Kindermann for collecting the self-introductory slides before the seminar and for assembling this report after the seminar.

2 Table of Contents

3 Overview of Talks about “Sets in Time”

3.1 Techniques for Visualizing Sets over Time

Peter Rodgers (University of Kent – Canterbury, GB)

This talk surveyed the current work and potential new avenues when visualizing both set and time data simultaneously. The motivation for this work comes from set based data that changes over time in research areas such as Social Media, Biosciences and Medicine. We consider mental map preservation over effective layout, adding dynamic aspects to current set visualization methods, and scalability issues.

The initial sections of the talk looked at the state-of-the-art in set visualization and time visualization. The set visualization summary largely came from a survey [2] and briefly outlined the wide variety of set visualization methods, from Euler-like, region oriented, line based, glyph and node-link based, to name just a few. The overview of time visualization methods were, again, largely based on a survey [1]. Techniques for time visualization are less diverse than set visualization, being broadly classified into linear and cyclical methods. The survey then overviewed the few existing visualization techniques that can claim to visualize both time and sets: TimeSets [6], Time-Sets [5], Hypenet [8], Bubble Sets [3], Dynamic Euler Diagrams [7], Linear Representations [9], and Circos [4].

An important take home message is that the number of visualization methods that consider both sets and time is small. Hence, given the demand for such techniques this area is a potentially fruitful research area. We consider that developing dynamic versions of existing set visualizations would be a rich seam of new ideas. Merging current set and time visualizations is another promising route to visualizing this complex data.

References

1 W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer London, 2011. doi:10.1007/978-0-85729-079-3.

2 B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. The state-of-the-art of set visualization. *Computer Graphics Forum*, 35(1):234–260, nov 2015. doi:10.1111/cgf.12722.

3 C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with iso-contours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, nov 2009. doi:10.1109/tvcg.2009.122.

4 M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. Circos: An information aesthetic for comparative genomics. *Genome Research*, 19(9):1639–1645, jun 2009. doi:10.1101/gr.092759.109.

5 M. Masoodian and L. Koivunen. Temporal visualization of sets and their relationships using time-sets. In *2018 22nd International Conference Information Visualisation (IV)*. IEEE, jul 2018. doi:10.1109/iv.2018.00025.

6 P. H. Nguyen, K. Xu, R. Walker, and B. W. Wong. TimeSets: Timeline visualization with set relations. *Information Visualization*, 15(3):253–269, oct 2015. doi:10.1177/1473871615605347.

7 P. Rodgers, P. Mutton, and J. Flower. Dynamic euler diagram drawing. In *2004 IEEE Symposium on Visual Languages – Human Centric Computing*. IEEE, 2004. doi:10.1109/vlhcc.2004.21.

8 P. Valdivia, P. Buono, and J.-D. Fekete. Hypenet: Visualizing dynamic hypergraphs. In A. P. Puig and T. Isenberg, editors, *EuroVis 2017 – Posters*. The Eurographics Association, 2017. doi:10.2312/eurp.20171162.

9 T. von Landesberger, S. Bremm, N. Andrienko, G. Andrienko, and M. Tekusova. Visual analytics methods for categoric spatio-temporal data. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, oct 2012. doi:10.1109/vast.2012.6400553.

3.2 Simultaneous Orthogonal Graph Drawing

Philipp Kindermann (Universität Würzburg, DE)

Joint work of Patrizio Angelini, Steven Chaplick, Sabine Cornelsen, Giordano Da Lozzo, Giuseppe Di Battista, Peter Eades, Philipp Kindermann, Jan Kratochvíl, Fabian Lipp, Ignaz Rutter
Main reference Patrizio Angelini, Steven Chaplick, Sabine Cornelsen, Giordano Da Lozzo, Giuseppe Di Battista, Peter Eades, Philipp Kindermann, Jan Kratochvíl, Fabian Lipp, Ignaz Rutter: "Simultaneous Orthogonal Planarity", in Proc. of the Graph Drawing and Network Visualization – 24th International Symposium, GD 2016, Athens, Greece, September 19-21, 2016, Revised Selected Papers, Lecture Notes in Computer Science, Vol. 9801, pp. 532–545, Springer, 2016.
URL https://doi.org/10.1007/978-3-319-50106-2_41

We introduce and study the OrthoSEFE-k problem: Given k planar graphs each with maximum degree 4 and the same vertex set, is there an assignment of the vertices to grid points and of the edges to paths on the grid such that the same edges in distinct graphs are assigned the same path and such that the assignment induces a planar orthogonal drawing of each of the k graphs?

We show that the problem is NP-complete for $k \geq 3$ even if the shared graph is a Hamiltonian cycle and has sunflower intersection and for $k \geq 2$ even if the shared graph consists of a cycle and of isolated vertices. Whereas the problem is polynomial-time solvable for $k = 2$ when the union graph has maximum degree five and the shared graph is biconnected. Further, when the shared graph is biconnected and has sunflower intersection, we show that every positive instance has an OrthoSEFE-k with at most three bends per edge.

3.3 Spatially and Temporally Coherent Visual Summaries

Wouter Meulemans (TU Eindhoven, NL)

Joint work of Juri Buchmüller, Wouter Meulemans, Bettina Speckmann, Kevin Verbeek, Jules Wulms

When exploring large time-varying data sets, visual summaries are a useful tool to identify time intervals of interest for further consideration. A typical approach is to represent the data elements at each time step in a compact one-dimensional form or via a one-dimensional ordering. Such 1D representations can then be placed in temporal order along a time line. There are two main criteria to assess the quality of the resulting visual summary: (1) how well does the 1D representation capture the structure of the data at each time step, and (2) how coherent or stable are the 1D representations over consecutive time steps or temporal ranges? We focus on techniques that create such visual summaries using 1D orderings for time-varying spatial data. Specifically, we consider the case of moving 2D point objects.

We first analyze three *orientation-based* shape descriptors on a set of continuously moving points: the first principal component, the smallest oriented bounding box and the thinnest strip. If we bound the speed with which the orientation of the descriptor may change, this may lower the quality of the resulting shape descriptor. We first show that there is no *stateless algorithm*, an algorithm that keeps no state over time, that both approximates the minimum cost of a shape descriptor and achieves continuous motion for the shape descriptor. On the other hand, if we can use the previous state of the shape descriptor to compute the new state, then we can define "chasing" algorithms that attempt to follow the optimal orientation with bounded speed. Under mild conditions, we show that chasing algorithms with sufficient bounded speed approximate the optimal cost at all times for oriented bounding boxes and strips.

To compute visual summaries, we introduce a stable and efficient ordering for moving points which is based on principal components. Our method allows us to make an explicit trade-off between the two quality criteria. We conduct computational experiments that compare our method to various state-of-the-art approaches for computing 1D orderings for spatial data, based on a set of well-established quality metrics that capture the two main criteria. The experiments show that our *Stable Principal Component* (SPC) algorithm outperforms existing methods: the spatial quality of SPC is essentially equivalent to the methods that perform best for this criterion, the run time of SPC is a fast as the fastest methods, and SPC is more stable than all other methods tested.

3.4 A Data-Driven Approach to Quality Metrics of Graph Visualizations

Tamara Mchedlidze (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Moritz Klammler, Tamara Mchedlidze, Alexey Pak
Main reference Moritz Klammler, Tamara Mchedlidze, Alexey Pak: "Aesthetic Discrimination of Graph Layouts", in Proc. of the Graph Drawing and Network Visualization – 26th International Symposium, GD 2018, Barcelona, Spain, September 26-28, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 11282, pp. 169–184, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-04414-5_12

Graph Visualization is a research area concerning automatic creation of pictorial representations of graphs. A node-link diagram (also called graph layout) is one of the most intuitive of these representations: the nodes are represented as 2 or 3-dimensional objects and edges as (poly-) lines or curves connecting the adjacent nodes. Node-link diagrams are used in a number of fields, including social science, bioinformatics, neuroscience, electronics, software engineering, business informatics and humanities.

Central to the Graph Visualization is the notion of the quality metric – a measure that formalizes how readable, clear and aesthetically pleasing a graph layout is. Some examples of simple quality metrics include number of edge crossings, edge crossing angle, drawing resolution. More complex quality metrics are the energy of a corresponding system of physical bodies and a linear combination of simple quality metrics. Quality metrics are utilized by network visualization algorithms to produce readable and aesthetically pleasing graph layouts.

In this talk I consider an alternative perspective on the quality metrics of graph layouts, by addressing the following question: "Of two given layouts of the same graph, which one is more aesthetically pleasing?". With that, I admit that "the ultimate" quality metric may not exist and one can hope for at most a (partial) ordering of layouts with respect to their aesthetic value. I introduce a neural network-based discriminator model trained on a labeled data set that decides which of two layouts has a higher aesthetic quality. The model demonstrates a mean prediction accuracy of 97.58%, outperforming discriminators based on an energy function and on the linear combination of popular quality metrics.

3.5 Perception of Space and Time – Implications for the Design of Visualizations

Margit Pohl (TU Wien, AT)

There is a considerable amount of research in cognitive psychology concerning the perception of space and time. Some of this research is relevant for the design of visualizations representing spatial and temporal data, although it should be mentioned that the application of basic research from psychology in visualization design is not always straightforward.

Human visual perception has several characteristics that are important for the design of visualizations. Gibson's ecological approach of visual perception implies that visual perception is related to movement in space. Perception is not a sequence of static pictures but a continuous flow of images while people move in the environment (optic flow field). When people are interested in objects in their environment they will move closer or go up and down a larger object (e.g., a house). In interfaces, such processes (zooming, panning, scrolling, ...) are mimicked. In contrast to other interaction possibilities, these are more natural and intuitive. This does not imply that less natural interaction possibilities are not effective, but they are less intuitive and have to be learned.

When people navigate in an environment they tend to develop schematic mental models of their environments. These mental models are incomplete and might be erroneous. Nevertheless, human navigation is generally very successful because they use information from the environment to continuously adapt their navigation (situatedness of spatial mental models).

Results from research on the perception of time is less relevant for the design of visual representations of temporal information. One important aspect in this context is the fact that humans use space as a metaphor for the representation of time. Using timelines is a very common way to reason about time. Another possibility is the usage of animation, although this should be designed appropriately. A specific challenge in the context of the visualization of spatial and temporal information is the combination of those two. When space is used as a metaphor for temporal information, this might not be compatible with the representation of geographic information. Especially in the context of very complex data, the usage of animation might be advisable.

4 Overview of Talks about "Sets in Space"

4.1 Evolving Sets in Space

Natalia V. Andrienko (Fraunhofer IAIS – Sankt Augustin, DE) and Gennady Andrienko (Fraunhofer IAIS – Sankt Augustin, DE)

Topic 1: Evolving spatial clusters (sets) of point events. Sets emerge, grow and shrink (cardinality and/or spatial extent), merge, split, disappear. Currently we use 3 visual displays: animated map, space-time cube, and bars along a time line (Gantt chart); each shows only a part of the information. Problem: the views are hard to link for getting the full picture. How to support visual exploration in a better way?

Topic 2: Subgroups in coordinated movement of multiple entities (e.g. shoal of fishes, football players). Synchronous coordination: continuous changes; subgroups of similarly moving entities emerge and change over time (e.g., entities separate from a group and join other groups). Problem: how to support visual exploration of subgroup formation and evolution?

Asynchronous coordination: multiple actors perform a sequence of activities towards a common goal, e.g., football players perform a sequence of passes during a game. Groups exist over certain time intervals. Changes are discrete: from interval to interval. Problem: how to support detecting repeated patterns of grouping and understanding the contexts in which they occur.

4.2 Dynamic Visualization of Interaction in Movement of Sets

Somayeh Dodge (University of Minnesota – Minneapolis, US)

Movement is a spatiotemporal process which involves space, time, and context. This presentation highlighted methods to visualize movement of sets and their interaction in space and time. 2D and 3D dynamic and interactive visualizations are created to highlight interaction among moving entities using an open-source visualization package, called DYNAMOvis to capture patterns of interaction. Direct interaction of entities are identified as the proximity of entities in space and time using spatial and temporal buffers. Visual variable color is used to highlight when entities are close together or meet at the same location and time. The geographic of context of movement and the interaction is visualized as background maps and satellite images. Time is captured using animation and the third dimension of space-time cube. As a a case study the presentation showed how the methods can be applied to highlight interaction between two tigers with adjacent home ranges.

4.3 Interactive Exploration of Spatio-Temporal Point Sets

Jan-Henrik Haunert (Universität Bonn, DE)

Joint work of Annika Bonerath, Jan-Henrik Haunert, and Benjamin Niedermann
Main reference Annika Bonerath, Jan-Henrik Haunert, and Benjamin Niedermann: "Computing alpha-shapes for temporal range queries on point sets". EuroCG 2019.
URL http://www.eurocg2019.uu.nl/papers/28.pdf

In my talk I present an overview of current developments of the Geoinformation Group at the University of Bonn, including algorithmic approaches to map generalization and label placement. A focus of my talk will be a new approach for the interactive exploration of spatio-temporal data. Generally, the aim is to develop data structures that can be repeatedly queried to obtain simple visualizations of parts of the data. In particular, the data is assumed to be a set of points each associated with a time stamp and the result of each query is visualized by an α-shape, which generalizes the concept of convex hulls. Instead of computing each shape independently, a simple data structure is proposed that aggregates the α-shapes of all possible queries. Once the data structure is built, it particularly allows a user to query single α-shapes without retrieving the actual (possibly large) point set.

5 Overview of Talks about "Graph Drawing and Set Visualization"

5.1 Survey on Graph and Hypergraph Drawing

André Schulz (FernUniversität in Hagen, DE)

Over the last 30 years graph drawing became an active area of research that bridges problems from graph theory and computational geometry. I present some classical graph drawing problems and discuss quality measures, prominent graph classes and drawing styles. Then a few selected representative graph drawing problems and algorithms are explained.

The second part of the talk covers hypergraphs and their connections to set visualization. I explain the differences between subset-based, partition-based and edge-based methods. I highlight the tight border between feasible and infeasible problems on the example on displaying 2 partitions simultaneously.

5.2 A Short Overview of Graph Drawing Beyond Planarity

Michael Bekos (Universität Tübingen, DE)

Beyond planarity is a new research direction in Graph Drawing, which is currently receiving increasing attention. Its primary motivation stems from recent cognitive experiments showing that the absence of specific kinds of edge-crossing configurations has a positive impact on the human understanding of a graph drawing. Graph drawing beyond planarity is concerned with the study of non-planar graphs that can be drawn by locally avoiding specific edge-crossing configurations or by guaranteeing specific properties for the edge crossings.

In this context, several classes of "beyond-planar graphs" have been introduced and studied, e.g., k-planar graphs, k-quasi planar graphs, right-angle-crossing graphs, fan-planar graphs, and fan-crossing free graphs. These classes of graphs have been mainly studied both in terms of their combinatorial properties and in terms of algorithms able to recognize and draw them.

In this talk, we will give a short overview of this new research direction aiming at covering a range of topics that are concerned with aspects of graph drawing and network visualization beyond planarity, such as combinatorial aspects of beyond-planar graphs, relationships between classes of beyond-planar graphs, and the complexity of the recognition problem for certain classes of beyond-planar graphs.

5.3 Supports for Hypergraphs

Sabine Cornelsen (Universität Konstanz, DE)

Joint work of Jan Christoph Athenstädt, Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, Arnaud Sallaberry
Main reference Jan Christoph Athenstädt, Sabine Cornelsen: "Planarity of Overlapping Clusterings Including Unions of Two Partitions", J. Graph Algorithms Appl., Vol. 21(6), pp. 1057–1089, 2017.
URL https://doi.org/10.7155/jgaa.00450
Main reference Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, Arnaud Sallaberry: "Path-based supports for hypergraphs", J. Discrete Algorithms, Vol. 14, pp. 248–261, 2012.
URL https://doi.org/10.1016/j.jda.2011.12.009
Main reference Ulrik Brandes, Sabine Cornelsen, Barbara Pampel, Arnaud Sallaberry: "Blocks of Hypergraphs – Applied to Hypergraphs and Outerplanarity", in Proc. of the Combinatorial Algorithms – 21st International Workshop, IWOCA 2010, London, UK, July 26-28, 2010, Revised Selected Papers, Lecture Notes in Computer Science, Vol. 6460, pp. 201–211, Springer, 2010.
URL https://doi.org/10.1007/978-3-642-19222-7_21

A support of a hypergraph is a graph such that each hyperedge induces a connected subgraph. Per definition, a hypergraph is (vertex-)planar if it has a planar support. A c-planar support is a planar support with a planar embedding in which no cycle composed of vertices of a hyperedge S encloses a vertex not in S. In an Euler diagram of a hypergraph each hyperedge S is represented by a simple closed region $R(S)$ bounded by a simple closed curve enclosing exactly the vertices in S. A c-planar support is related to an Euler diagram in which for any two hyperedges S_1, S_2 we have that (a) each connected component of $R(S_1) \cap R(S_2)$ contains a vertex and (b) $R(S_1) \subset R(S_2)$ if $S_1 \subset S_2$.

It can be decided in polynomial time whether a hypergraph H admits a c-planar support if the underlying graph H_2 of hyperedges of size two is already a support. In general, it is NP-complete to decide whether a hypergraph has a c-planar support, even if H_2 is biconnected and induces at most two connected components on each hyperedge. A cactus support is always c-planar. Using a decomposition into blocks, it can be decided in polynomial time whether a hyperedge admits a cactus support.

A support is path-based if each hyperedge contains a spanning path. It can be decided in polynomial time whether a hyperedge has a path-based tree support.

5.4 Short Plane Supports for Spatial Hypergraphs

Martin Nöllenburg (TU Wien, AT)

Joint work of Thom Castermans, Mereke van Garderen, Wouter Meulemans, Martin Nöllenburg, and Xiaoru Yuan
Main reference Thom Castermans, Mereke van Garderen, Wouter Meulemans, Martin Nöllenburg, Xiaoru Yuan: "Short Plane Supports for Spatial Hypergraphs", in Proc. of the Graph Drawing and Network Visualization – 26th International Symposium, GD 2018, Barcelona, Spain, September 26-28, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 11282, pp. 53–66, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-04414-5_4

A spatial hypergraph $H = (V, \mathcal{S})$ is a hypergraph on a set of points $V \subset \mathbb{R}^2$ with $\mathcal{S} \subset 2^V$. A support graph of H is a graph $G = (V, E)$ on the same vertex set V such that for every hyperedge $S \in \mathcal{S}$ the induced graph $G[S]$ is connected. Support graphs are useful geometric structures for various types of set visualizations such as line sets or KelpFusion [1, 2], which enclose or span each hyperedge. We are interested in short supports that use a small amount of ink, that have no edge crossings and that are actually trees. We concentrate on instances with two hyperedges with non-empty intersection.

In the talk I first present a simple sufficient condition for the existence of plane tree supports, based on minimum spanning trees on the non-empty intersection of all hyperedges. However, this idea may lead to plane tree supports being longer by a linear factor than the shortest plane tree support. In fact, it is NP-hard to minimize the total edge length of plane tree supports even if $\mathcal{S}$ contains just two hyperedges. From a practical point of view, I sketch two heuristic algorithms, using either iterated minimum spanning tree computations or local search, possibly relaxing the requirement of planarity or being a tree. An experimental evaluation showed that the local search performs quite well in terms of quality and is still reasonably fast in practice. The extension of this problem to temporal hypergraphs satisfying stability constraints of the support graphs over time is open.

References

1 B. Alper, N. Henry Riche, G. Ramos, and M. Czerwinski. Design study of linesets, a novel set visualization technique. *IEEE Trans. Visualization and Computer Graphics* 17(12):2259–2267, 2011, 10.1109/TVCG.2011.186.

2 W. Meulemans, N. Henry Riche, B. Speckmann, B. Alper, and T. Dwyer. Kelpfusion: A hybrid set visualization technique. *IEEE Trans. Visualization and Computer Graphics* 19(11):1846–1858, 2013, 10.1109/TVCG.2013.76.

6 Panel Discussion

The panel discussion was entitled "Visual Analytics for Sets over Time and Space: What are the burning scientific questions? An interdisciplinary perspective."

6.1 There's a Space for That!

André Skupin (San Diego State University, US)

URL https://datascience.bigknowledge.net/explorer/

The relevance of cartography and geographic thinking extends far beyond the traditional bounds of applying mapping and spatial analysis to study phenomena in geographic space. With some imagination, the elements of any domain can be seen as simultaneously existing in a multitude of spaces, such as attribute space, network space, or knowledge space. Visual analytics then becomes about making any such space accessible to human cognition in order to support more informed decision-making. Once such an overarching spatial viewpoint is in place, any data set can be transformed into engaging artifacts that support research, education, and practice. For example, geographic cells monitored via multi-temporal satellite images can be seen as traversing paths through multi-spectral space. Wildfire events cause rapid shifts of cells, while post-fire recovery carves a slow and steady path over the course of several years. Thanks to dimensionality reduction and visual analytics, we can now SEE this. Meanwhile, natural language processing and machine learning can be leveraged into producing detailed domain base maps, as I presented here for the Data Science & Analytics domain. Individual documents or whole repositories – such as the abstracts of all Dagstuhl Seminars – could now be explored with the ease of everyday mapping interfaces.

6.2 VASet over Time and Space – A GeoVis Perspective

Susanne Bleisch (FH Nordwestschweiz – Muttenz, CH)

Main reference Arzu Çöltekin, Susanne Bleisch, Gennady Andrienko, Jason Dykes: "Persistent challenges in geovisualization – a community perspective". International Journal of Cartography 3(sup1):115–139 (2017).
URL https://doi.org/10.1080/23729333.2017.1302910

The recent paper "Persistent challenges in geovisualization – a community perspective" [1] analyzes and summarizes the input on persistent challenges from four different expert workshops and contrasts them with more top-down research agendas. One of the identified points is also relevant and important for set visualizations – that is, matching data to tasks to visualization types to make it easier to work visually with these data sets.

Additionally, I argue that on the visualization continuum from exploration to communication we too often lean towards optimization of the visualization, which requires knowledge of the task, data insight, the user, etc., and is thus on the communication side. Doing exploration is more difficult as the data and tasks may be ill-defined or, with regard to the tasks, more or less unknown. Similarly with knowing about the users. User knowledge from evaluation is generalized so that we can design for the average user or the majority of users. But our future goal should be personalization. This is ideally not based on preference but rather on performance in terms of gaining knowledge about the data. One potential way to achieve more exploration and more personalization–where in both cases the effectiveness of gaining insights counts–is the combination of the results of exploratory visualizations through new and/or additional forms of visualizations, i.e., by finding the common and constant or the diverging results.

References

1 Çöltekin, Arzu and Bleisch, Susanne and Andrienko, Gennady and Dykes, Jason (2017). Persistent challenges in geovisualization – a community perspective. *International Journal of Cartography.* 3:sup1, 115–139, .

7 Working Groups

7.1 Concentric Set Schematization

Michael Bekos (Universität Tübingen, DE), Fabian Frank (University of Arizona – Tucson, US), Wouter Meulemans (TU Eindhoven, NL), Peter Rodgers (University of Kent – Canterbury, GB), and André Schulz (FernUniversität in Hagen, DE)

Various techniques have been developed to visualize sets, either on a geographically accurate base map or in an abstract non-spatial layout. However, geographic accuracy is often not necessary for overview tasks or tasks focusing on set structures. Yet, completely discarding spatial context may also hide structure or patterns.

Schematic maps have been successful in various applications, such as metro maps, by simplifying and abstracting spatial relations to a minimum functional level, thereby clarifying and emphasizing structure in data while not disregarding (geographic) space.

In this working group, we explore the possibility of computing schematic set visualizations. That is, we are given a set of points in a geographic space, each associated with one or more sets. We want to shift the points to new locations such that we can provide a clear representation for each of the sets; this representation is a geometry connecting (e.g. a tree, cycle or path) or encompassing (e.g. a simple polygon) exactly the points that belong to the set. The main considerations are the extent of the changes we allow to the points, such that we can control for geographic distortion, and the criteria and measures to assess the quality of the resulting set representations.

7.1.1 Sample of Related Work

Set visualization (also known as hypergraph drawing) has received attention in both the visualization and the graph-drawing communities. A recent survey [3] shows variety of visualization techniques; various of these methods target spatial data, e.g. [2, 6, 7, 10]. In the graph-drawing community, most attention has been afforded to hypergraph supports [9] for both fixed and free vertex locations, e.g. [1, 4, 5, 8].

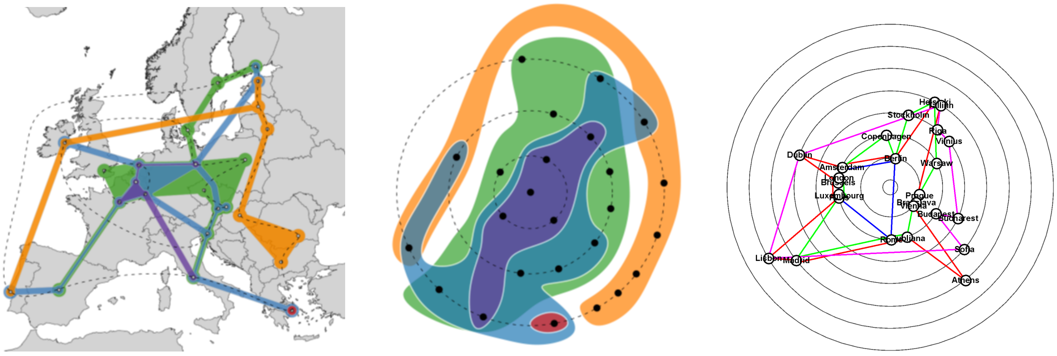

Figure 1 Left: KelpFusion set visualization [10] on a geographically accurate base map. Middle: Manual sketch of a concentric schematic representation of the sets. Right: Preliminary result of our prototype implementation.

7.1.2 Concentric Set Schematization

We study a model where we are given, in addition to the geospatial set system, a set of concentric circles. The goal is to place the given points on the circles such that a clear representation of the set system is given, without distorting the geospatial situation too severely. Through various models of allowed distortion can be considered, we focus on allowing points to move only along the ray through it, originating from the circles' center. For every point an interval of possible circle locations is given with the input. The sets are to be represented as connecting geometries (trees, paths, or cycles). That is, we aim to compute a point placement together with a support of the hypergraph (set system), such that their combination has good quality.

The first question we studied was on computing a good support, measured by the number of intersections the support induces. We found relations to existing work on layered graph drawing and book embeddings. However, the problem as modeled provides different challenges. We were able to show that, given only two circles and a set system of non-intersecting sets, testing whether a planar support exists can be tested easily as follows. We found two configurations, one that implies that a support edge much cross from one circle to the other, and one that implies that a support edge cannot cross. If there are no contradicting configurations, then a planar support exists, by appropriately choosing sides for each point. However, we conjecture that the general version of this problem (intersecting sets, multiple circles) is NP-hard.

The second question we studied assumes that we have a support given (designed or computed by an algorithm), and want to decide for each point on which circle is should be placed. We show that, if we want to minimize the radial change between any points connected by the support, a simple (integer) linear program suffices. We prove that the relaxation has an optimal integer solution and thus the problem can be solved efficiently. In particular, for every LP solution we can remove a set of constraints that are not tight and obtain a LP with the same solution, for which the underlying matrix is totally unimodular. In fact, all vertices of the feasible region induced by the original LP are integral and in bijection to the layer assignments. As a consequence the optimization problem can be solved by greedily improving a layer assignment.

7.1.3 Outlook

These initial findings leave us with a host of interesting questions, both algorithmic in nature and on visual design of concentric set schematization. In particular, we plan to further investigate different models of spatial distortion and criteria for layout quality. We also need to determine how to best route the connecting geometries to obtain an effective design of schematic set representations.

References

1 H. A. Akitaya, M. Löffler, and C. D. Tóth. Multi-colored spanning graphs. In *Graph Drawing and Network Visualization (GD'16)*, LNCS 9801, pages 81–93, 2016.

2 B. Alper, N. Henry Riche, G. Ramos, and M. Czerwinski. Design study of LineSets, a novel set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2259–2267, 2011.

3 B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers. The state of the art of set visualization. *Computer Graphics Forum*, 35(1):234–260, 2016.

4 U. Brandes, S. Cornelsen, B. Pampel, and A. Sallaberry. Path-based supports for hypergraphs. *Journal of Discrete Algorithms*, 14:248–261, 2012.

5 T. Castermans, M. van Garderen, W. Meulemans, M. Nöllenburg, and X. Yuan. Short plane supports for spatial hypergraphs. In *Graph Drawing and Network Visualization (GD'18)*, LNCS 11282, pages 1–14, 2018.

6 C. Collins, G. Penn, and S. Carpendale. Bubble Sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, 2009.

7 K. Dinkla, M. van Kreveld, B. Speckmann, and M. Westenberg. Kelp Diagrams: Point set membership visualization. *Computer Graphics Forum*, 31(3pt1):875–884, 2012.

8 F. Hurtado, M. Korman, M. van Kreveld, M. Löffler, V. Sacristán, A. Shioura, R. I. Silveira, B. Speckmann, and T. Tokuyama. Colored spanning graphs for set visualization. *Computational Geometry: Theory and Applications*, 68:262–276, 2018.

9 D. S. Johnson and H. O. Pollak. Hypergraph planarity and the complexity of drawing Venn diagrams. *Journal of Graph Theory*, 11(3):309–325, 1987.

10 W. Meulemans, N. Henry Riche, B. Speckmann, B. Alper, and T. Dwyer. KelpFusion: A hybrid set visualization technique. *IEEE Transactions on Visualization and Computer Graphics*, 19(11):1846–1858, 2013.

7.2 From Linear Diagrams to Interval Graphs

Steven Chaplick (Universität Würzburg, DE), Michael Kaufmann (Universität Tübingen, DE), and Stephen G. Kobourov (University of Arizona – Tucson, US)

7.2.1 Introduction

There are many different approaches to visualizing sets. Here we are interested in a particular type of visualization called *linear diagram* [7, 11]. In such visualizations the sets are represented by line segments and the membership of an element in multiple sets is represented by an overlap between the segments in the corresponding sets; see Fig. 2. There is some evidence that linear diagrams are better than Euler diagrams [9].

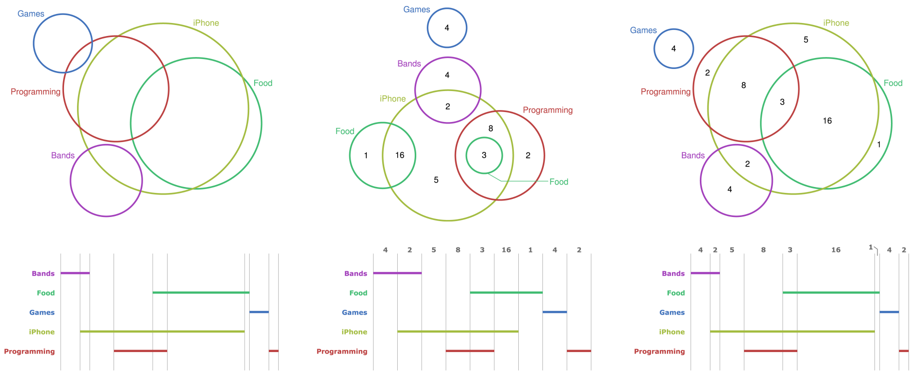

Figure 2 Example of linear diagrams from [9].

We note that linear diagrams are related to the classical interval representation of graphs, originating in combinatorics [6] and genetics [1], and we want to study the connections between the linear diagrams and interval representations. For example, a natural question to ask is given a particular set system, is it possible to represent it so that each set has exactly one segment? The equivalent question is whether a given graph is an interval graph. This problem can be decided in linear time [2, 3]. If such a solution does not exist, then it would be nice to minimize the number of segments per set. However, this problem is NP-hard, even when asking whether 2 segments per set suffice [12].

If we want to visualize linear diagrams over time, there are several parameters to consider:

- do we consider the input given offline (all information given at once) or online (information given incrementally)
- weighted/unweighted (proportional/not) intervals
- vertices have fixed positions over time, or they move
- no splits of intervals are allowed, or we want to minimize the number of splits
- if splits are allowed, we minimized the total number of attributes that get split, or the total number of segments in the representation.

7.2.2 Summary of Discussion

We formalize a *linear diagram* of a set C of n characters $\{c_1, \ldots, c_n\}$ and a set A of k attributes $\{a_1, \ldots, a_k\}$ as a hypergraph $H = (V, E)$ as follows. Let M be the incidence matrix where each row corresponds to a character and each column to an attribute such that the entry $M_{i,j}$ is 1 when character c_i has attribute a_j.

Remarks: across all variants, we should have all characters with the same attribute set consecutive (i.e., consecutive rows). Note that for a single time step we can consider “condensing” twins, e.g., showing the count of the twins instead of each individual.

- Two natural measures of the quality of a linear diagram are the number of segments per attribute and the total number of segments in the whole diagram. It is NP-complete to test if a hypergraph can be realized with 2 segments per attribute (as this is recognition of 2-interval graphs [12]), but it remains open for approximation. On the other hand the status of minimizing the total number of segments seems open.
- As these problems are likely to be computationally difficult, one idea is to employ a simple greedy heuristic to produce some split interval representation, e.g., by iteratively introducing vertices (using the PQ-tree approach) so as to minimize the number of segments for the new vertex. A limitation here is that the first two vertices are certainly never split.

We considered two different models of linear diagrams.

- Model 1: rows are fixed
  - Option 1: suppose we obtain an interval model for each time step (somehow, e.g„ by running the greedy heuristic). If the combined graph is an interval graph, then there is a permutation of the rows consistent with all time steps.
  - Option 2: use heuristic/approximation approach (black-box) to obtain "splits" on all attributes over all time steps. This would provide an interval model for the entire time frame, but it seems unlikely that such diagrams would provide good visualizations (due to having many splits).
- Model 2: rows can move. (when Model 1 fails)
  - (as in Option 1 above): suppose we obtain an interval model for each time step (somehow, eg, by running a greedy heuristic).
  - Limitation: Minimizing the difference between a permutation of time steps $i, i+1$ the problem is NP-hard (as this is related to the tanglegram problem [4]).
  - Potential upside: an efficient algorithm for the related tanglegram problem where one permutation is fixed and the other is flexible (in their case, from a binary tree) [4], seems likely to be able to be generalized to our case when similarly one permutation is fixed and the other should be obtained from an appropriate PQ-tree.

Some further considerations regarding these problems and how to solve them include:

- characters appearing and disappearing over time.
- order of columns, e.g., to maximize overlaps on consecutive attributes or optimize the intervals of the characters (e.g., gaps on their attributes).
- exact algorithms via ILP/SAT formulations, or dynamic programming.

Finally, a further direction in the context of linear diagrams is how they relate to Euler diagrams. In an Euler diagram (e.g., see the top row of Figure 2) the *regions* are the faces of the planarization of the Euler Diagram (made up of closed curves, one for each attribute). If we trace all regions of an Euler Diagram with a closed curve so that this curve visits each region exactly once, then this describes a permutation on the regions. In particular, the number of times the curve visits a each attribute, is the equal to the number of segments in the linear diagram corresponding to the permutation on the regions.

There are some connections to traversing a path in the Euler diagram. But not all sets can be represented by Euler diagrams [8]. There is more background about "well formed Euler diagrams" and what can and cannot be done in these papers [5, 10]. Such connections seem to be an interesting topic for further consideration.

7.2.3 Specific Directions to Study

1. Describe and analyze the greedy heuristic for computing an initial linear diagram, based on inserting the first 2 segments in the PQ-tree without any splits and then splitting as little as possible, given the current PQ-tree. Does this provide any approximation guarantees? (e.g„ to the problem of minimizing the total number of intervals in the linear diagram).
2. In the second model we have a tanglegram variation where the two trees that we want to align are not binary trees but PQ-trees. This means that there are some nodes of high degree and we might need to compute all permutations for their children. This could lead to an efficient algorithm (possibly parameterized by the maximum degree of the PQ-tree).
3. It remains to formalize the precise algorithm to obtain a split interval representation using the PQ-tree approach and to prove that it is *incrementally* optimal.

4. What can we say about the connection to Euler diagrams? For example, if the input set system is nice (well-formed, or whatever property we need), is it true that the best linear diagram corresponds to a Hamiltonian path in the dual of the Euler diagram?

References

1 Seymour Benzer. On the topology of the genetic fine structure. *Proceedings of the National Academy of Sciences of the United States of America*, 45(11):1607–1620, 1959.

2 Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, 1976.

3 Derek G. Corneil, Stephan Olariu, and Lorna Stewart. The LBFS structure and recognition of interval graphs. *SIAM J. Discrete Math.*, 23(4):1905–1953, 2009.

4 Henning Fernau, Michael Kaufmann, and Mathias Poths. Comparing trees via crossing minimization. *J. Comput. Syst. Sci.*, 76(7):593–608, 2010.

5 Jean Flower and John Howse. Generating Euler diagrams. In *International Conference on Theory and Application of Diagrams*, pages 61–75. Springer, 2002.

6 G. Hajós. Über eine Art von Graphen. *Int. Math. Nachr.*, 11:65, 1957.

7 Masood Masoodian and Laura Koivunen. Temporal visualization of sets and their relationships using time-sets. In *22nd International Conference Information Visualisation (IV)*, pages 85–90. IEEE, 2018.

8 Paolo Simonetto and David Auber. Visualise undrawable Euler diagrams. In *12th International Conference Information Visualisation*, pages 594–599. IEEE, 2008. doi:10.1109/iv.2008.78

9 Gem Stapleton, Peter Chapman, Peter Rodgers, Anestis Touloumis, Andrew Blake, and Aidan Delaney. The efficacy of Euler diagrams and linear diagrams for visualizing set cardinality using proportions and numbers. *PloS one*, 14(3):e0211234, 2019. doi:10.1371/journal.pone.0211234

10 Gem Stapleton, John Howse, and Peter Rodgers. A graph theoretic approach to general Euler diagram drawing. *Theoretical Computer Science*, 411(1):91–112, 2010.

11 Paola Valdivia, Paolo Buono, and Jean-Daniel Fekete. Hypenet: visualizing dynamic hypergraphs. In *EuroVis 2017 – 19th EG/VGC Conference on Visualization*, pages 1–3, 2017.

12 Douglas B. West and David B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discrete Applied Mathematics*, 8(3):295–305, 1984.

7.3 Thread Visualization

William Evans (University of British Columbia – Vancouver, CA), Somayeh Dodge (University of Minnesota – Minneapolis, US), Fabian Frank (University of Arizona – Tucson, US), Panos Giannopoulos (City – University of London, GB), and Giuseppe Liotta (University of Perugia, IT)

We considered the problem of visualizing the communication pattern between concurrent threads in a distributed computation. One particular visualization (see https://bestchai.bitbucket.io/shiviz/) draws vertical lines in 2d to represent individual threads. The y-dimension represents time, increasing from top to bottom. If one thread A sends a message to another thread B at time s, and thread B receives it at time t, then there is a segment connecting the vertical line representing A at y-coordinate s to the vertical line representing

B at y-coordinate t. For complicated communication patterns, these *communication segments* may be hard to distinguish: they may intersect each other and the *thread lines*, nearly overlap, or be nearly vertical. Our working group considered the problem of choosing the position and x-order of the vertical thread lines in order to minimize the number of crossings of the communication segments. We showed that this problem is NP-complete.

Another possible visualization of thread communication is to draw the communication segments vertically between the send and receive time y-coordinates. The thread lines then become y-monotone polygonal chains that connect the endpoints of these communication segments that represent send or receive events experienced by the thread. We considered the problem of choosing the position and x-order of the vertical communication segments in order to minimize the number of crossings in the drawing. We conjecture that this problem in NP-complete as well, but we made some progress on finding an efficient algorithm to test if a plane drawing exists.

7.4 Clustering Colored Points in the Plane

Jan-Henrik Haunert (Universität Bonn, DE), Hugo Akitaya (Tufts University – Medford, US), Sabine Cornelsen (Universität Konstanz, DE), Philipp Kindermann (Universität Würzburg, DE), Tamara Mchedlidze (KIT – Karlsruher Institut für Technologie, DE), Martin Nöllenburg (TU Wien, AT), Yoshio Okamoto (The University of Electro-Communications – Tokyo, JP), and Alexander Wolff (Universität Würzburg, DE)

Analyzing large sets of geographic objects requires visualizations that present the most important spatial patterns in a legible way. Therefore, one often aims to aggregate the given objects to form larger and more abstract entities that can be displayed with low visual complexity. Aggregation is often performed with clustering algorithms that are based on similarity and proximity. Consider, for example, a set of points of interest (POIs) that belong to different categories (e.g., restaurants, shops, etc.). A reasonable clustering approach is to compute a proximity graph of all points (e.g., the Delaunay triangulation), to discard all edges that connect two points of different categories, and to consider each connected component of the remaining graph $G = (V, E)$ as a cluster. Figure 3 shows the result of this procedure for a set of POIs from OpenStreetMap. Similar triangulation-based methods for point-set clustering have been suggested to find groups of animals, e.g., flocks of birds that are relatively near to each other and have the same heading with respect to a few cardinal directions [4].

Although the Delaunay triangulation yields reasonably defined clusters, the obtained solution may not be satisfactory, for example, because the obtained clusters may be considered too small or not compact enough. Therefore, in this report, we introduce more flexibility into the clustering procedure by using substantially denser proximity graphs. In particular, we do not require the graph G used for clustering to be planar and, therefore, need to deal with edge crossings. We still require that each cluster c is connected in G, i.e., G contains a tree T_c spanning the nodes in c. However, we shall not simply report the connected components of G as clusters since, due to the edge crossings in G, it may be difficult to visualize such clusters in a legible way. Instead, we introduce the following basic optimization problem to find large non-overlapping clusters.

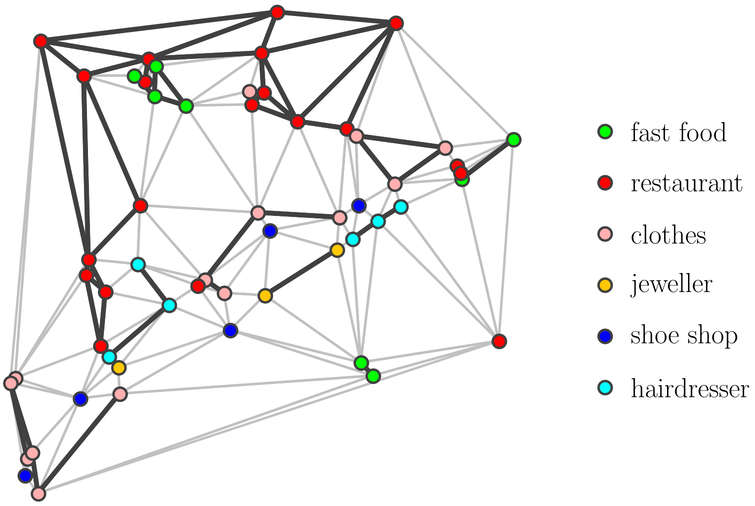

Figure 3 A set of points of interest from OpenStreetMap with its Delaunay triangulation. Edges connecting two points of the same category are displayed fat. These edges induce a subgraph $G = (V, E)$ of the Delaunay triangulation whose connected components can be considered as clusters.

▶ **Cluster Minimization.** Let $G = (V, E)$ be a geometric graph that may contain edge crossings. Find a subgraph H of G with node set V, i.e., a graph $H = (V, E')$ with $E' \subseteq E$, such that no two edges of E' cross and the number of connected components of H is as small as possible.

Note that we do not require the nodes of G to belong to any category. We will, however, focus on special cases of the problem that require categorized points as input. Generally, we refer to the categories as *colors*. Unless stated otherwise, we require that each point is assigned to exactly one of k colors. Consequently, we refer to G as *k-partitioned*. An edge connecting two points of the same color is called *colored* and its color equals the common color of its two end points; other edges are called *uncolored*. A crossing of two colored edges is called *monochromatic* if both edges have the same color and, otherwise, *bichromatic*. *Planarizing* a monochromatic crossing of two edges $e = \{u, v\}$ and $f = \{p, q\}$ means introducing a new node x of the same color as the four involved nodes at the intersection of e and f and replacing these edges with edges $\{u, x\}$, $\{v, x\}$, $\{p, x\}$, and $\{q, x\}$. A graph is *1-planar* if each edge is crossed by at most one other edge. With these definitions, we are ready to define special cases of Cluster Minimization for our further investigations:

▶ **Cluster Minimization in 1-Planar Graphs.** G is obtained from a k-partitioned, 1-planar graph that contains only colored edges, by planarizing monochromatic crossings. The problem is to solve Cluster Minimization for G.

▶ **Cluster Minimization in Complete Graphs.** G is k-partitioned and complete in the sense that every two points of equal color are connected with an edge and there is no uncolored edge. The problem is to solve Cluster Minimization for G.

Cluster Minimization in 1-Planar Graphs arises in the situation that G is constructed by computing a 1-planar proximity graph of the given points, reducing it to its colored edges, and planarizing its monochromatic crossings. For example, one may define the edge set of the proximity graph as the union of the edge sets of all order-1 Delaunay triangulations of the point set. The resulting proximity graph is 1-planar [3] and, due to its close relationship with the Delaunay triangulation, reflects the proximity relationships of the points reasonably well. On the other hand, since this graph has substantially more edges than the Delaunay triangulation, an optimal solution of Cluster Minimization may contain fewer and larger clusters than the approach based on the Delaunay triangulation illustrated in Figure 3. The idea behind Cluster Minimization in Complete Graphs is to define the graph G as large as possible and, thereby, to place even more emphasis on creating few and large clusters.

Finally, we introduce a problem that is of relevance if one wishes to place more emphasis on creating compact clusters.

▶ **Edge Maximization.** Let $G = (V, E)$ be a k-partitioned geometric graph that contains only colored edges and that may contain monochromatic as well as bichromatic edge crossings. Find a subgraph H of G with node set V, i.e., a graph $H = (V, E')$ with $E' \subseteq E$, such that $|E'|$ is as large as possible and H contains no bichromatic edge crossing.

Again, G may be obtained by computing some proximity graph and reducing it to its colored edges. The idea behind maximizing the number of edges of H is that a set of nodes that is densely connected in G constitutes a compact cluster. Bichromatic edge crossings in the output graph H are forbidden to avoid overlapping clusters of different colors. Monochromatic edge crossings, on the other hand, are allowed since they could be planarized before computing the output clusters as the connected components of H.

7.4.1 Results

We have obtained the following main results for the problems defined above:

- Cluster Minimization in 1-Planar Graphs can be solved efficiently with a simple greedy algorithm.
- Cluster Minimization in Complete Graphs can be solved efficiently if there are only $k = 2$ different colors, by adapting an algorithm by Bereg et al. [1]. For $k = 3$, however, it is NP-hard, which we showed by reduction from Tree-Residue Vertex-Breaking [2].
- For general Cluster Minimization, we devised an integer linear program and a heuristic based on the greedy algorithm for the 1-planar case.
- Edge Maximization can be solved efficiently for $k = 2$ using an algorithm for minimum vertex cover in bipartite graphs. For arbitrary k and 1-planar graphs, the problem is trivial since for every edge crossing one can arbitrarily decide which of the two involved edges to select.

7.4.2 Future Work

We plan to continue our research with respect to both (a) a further theoretical study of open problems and (b) an evaluation of our clustering approach based on implementations of our algorithms and experiments with real-world data. We already started to develop ideas for the case that each point has more than one color and we were able to generalize some of our algorithms for this case. Moreover, we have first ideas on how to deal with a dynamic situation in which the points move or change their colors over time and the aim is to compute a dynamic visualization while considering the stability of the visualization as an additional optimization criterion.

References

1 S. Bereg, M. Jiang, B. Yang, and B. Zhu. On the red/blue spanning tree problem. *Theoret. Comput. Sci.*, 412(23):2459–2467, 2011. doi:10.1016/j.tcs.2010.10.038.

2 E. D. Demaine and M. Rudoy. Tree-Residue Vertex-Breaking: A new tool for proving hardness. In D. Eppstein, editor, *Proc. 16th Scandinavian Symp. Algorithm Theory (SWAT'18)*,

volume 101 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.SWAT.2018.32.

3 J. Gudmundsson, M. Hammar, and M. van Kreveld. Higher order Delaunay triangulations. *Comput. Geom. Theory Appl.*, 23:85–98, 2002. doi:10.1016/S0925-7721(01)00027-X.

4 P. Laube, M. van Kreveld, and S. Imfeld. Finding REMO – Detecting relative motion patterns in geospatial lifelines. In *Developments in Spatial Data Handling – Proc. 11th Int. Symp. Spatial Data Handling (SDH'05)*, pages 201–215. Springer-Verlag, Berlin, Germany, 2005. doi:10.1007/3-540-26772-7_16.

7.5 Flexible Visualization of Sets over Time and Space

Christian Tominski (Universität Rostock, DE), Gennady Andrienko (Fraunhofer IAIS – Sankt Augustin, DE), Natalia V. Andrienko (Fraunhofer IAIS – Sankt Augustin, DE), Susanne Bleisch (FH Nordwestschweiz – Muttenz, CH), Sara Irina Fabrikant (Universität Zürich, CH), Eva Mayr (Donau-Universität Krems, AT), Silvia Miksch (TU Wien, AT), Margit Pohl (TU Wien, AT), and André Skupin (San Diego State University, US)

Sets with references to time and space are difficult to visualize. The reason is that multiple aspects of the data must be communicated to the user. In the first place, the set characteristics need to be encoded visually. Moreover, data attributes associated with the set members are important information to be visualized. On top of that, the temporal and spatial frames of reference need to be displayed. Uncertainty of the data might also play a role. Encoding all of these aspects into a single visual representation is typically impractical because the result would likely be complicated to interpret. An alternative is to use multiple views where each view focuses on selected aspects of the data. Yet, connecting findings made in one view to findings made in another view can be a considerable effort. On an elementary level, the user needs to understand how a data object in one view, i.e., in one reference system, relates to another perspective in another view or reference system.

Given the two extremes of fully integrating all aspects in a single representation and separating selected aspects into multiple representations, the working group discussed an alternative option in between, which we call *flexible visualization.* The idea is to bring the two extremes of integration and separation closer together by means of animated transitions. The starting point is to have visual representations that show the data with selected aspects being prioritized and other aspects being attenuated or omitted. For example, a set of movement trajectories is shown on a 2D map to prioritize the spatial aspect of the data. Another view might show the same data as stacked 3D bands above a map to better reveal the data attributes along individual trajectories. Yet another view might show the data as horizontal 2D bands to emphasize the temporal aspect of the data. Now, the core idea of flexible visualization is to have smooth transitions that transform one view into another, rather than showing them as multiple views. That said, flexible visualization aims to balance visual complexity and interaction while providing opportunities to see data and patterns from multiple perspectives.

There are already existing approaches that implement smooth transitions to flexibly animate between views. We recognized the need for a systematic approach to categorizing flexible visualizations in order to gain a better understanding of the potential and limitations

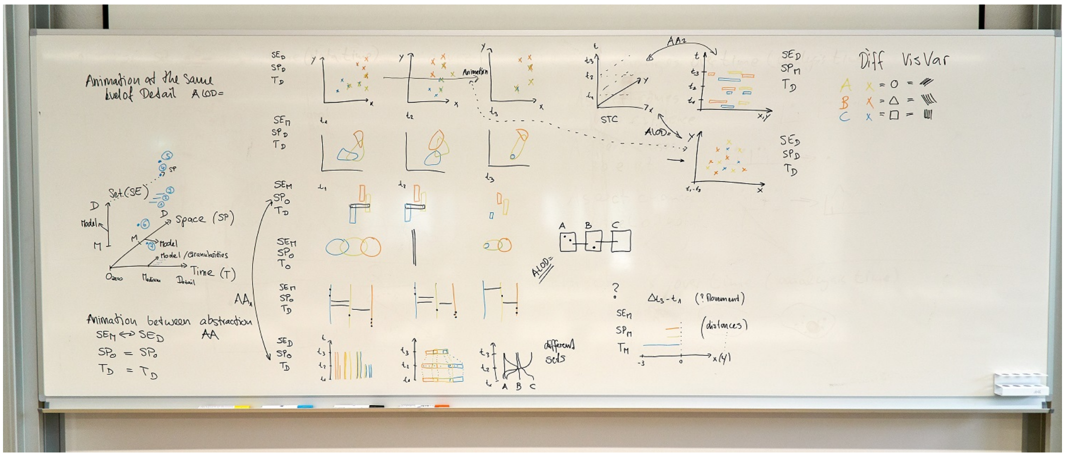
Animation at the same level of Detail ALOD=
Set (SE)
Space (SP)
Time (T)
Animation between abstraction
AA
SEΠ ⇔ SED
SPO = SPO
TD = TD
STC
Diff VisVar
A B C

Figure 4 Sketches to systematize flexible visualization.

of augmenting the visual analysis by transitions between discrete visual states. The working group split up into two subgroups to discuss in detail several questions related to flexible visualization, including:

1. Conceptual and technical aspects
   - What are the requirements for flexible visualization?
   - What principle transitions between views are possible and make sense?
   - What topology might flexible visualizations exhibit?
   - Where can smooth transitions operate, in data space or in view space?
   - How can flexible visualization be implemented?
2. Human aspects
   - What are perceptual and cognitive constraints?
   - How should an animated transition be designed?
   - What is the role of interactive user control?
   - Where is the sweet spot between abrupt change and very smooth transitions?
   - Does flexible visualization scale to very large data?

The working group developed first sketches to systematize flexible visualization. Figure 4 shows an example. A first draft of a research publication has been prepared. The goal of this publication is to characterize flexible visualization comprehensively as a viable approach to enhance the visual analysis of complex multi-aspect sets.

Participants

- Hugo Akitaya
Tufts University – Medford, US
- Gennady Andrienko
Fraunhofer IAIS –
Sankt Augustin, DE
- Natalia V. Andrienko
Fraunhofer IAIS –
Sankt Augustin, DE
- Michael Bekos
Universität Tübingen, DE
- Susanne Bleisch
FH Nordwestschweiz –
Muttenz, CH
- Steven Chaplick
Universität Würzburg, DE
- Sabine Cornelsen
Universität Konstanz, DE
- Somayeh Dodge
University of Minnesota –
Minneapolis, US
- William Evans
University of British Columbia –
Vancouver, CA
- Sara Irina Fabrikant
Universität Zürich, CH
- Fabian Frank
University of Arizona –
Tucson, US
- Panos Giannopoulos
City – University of London, GB
- Jan-Henrik Haunert
Universität Bonn, DE
- Michael Kaufmann
Universität Tübingen, DE
- Philipp Kindermann
Universität Würzburg, DE
- Stephen G. Kobourov
University of Arizona –
Tucson, US
- Giuseppe Liotta
University of Perugia, IT
- Eva Mayr
Donau-Universität Krems, AT
- Tamara Mchedlidze
KIT – Karlsruher Institut für
Technologie, DE
- Wouter Meulemans
TU Eindhoven, NL
- Silvia Miksch
TU Wien, AT
- Martin Nöllenburg
TU Wien, AT
- Yoshio Okamoto
The University of
Electro-Communications –
Tokyo, JP
- Margit Pohl
TU Wien, AT
- Peter Rodgers
University of Kent –
Canterbury, GB
- André Schulz
FernUniversität in Hagen, DE
- André Skupin
San Diego State University, US
- Christian Tominski
Universität Rostock, DE
- Alexander Wolff
Universität Würzburg, DE

Report from Dagstuhl Seminar 19202

Approaches and Applications of Inductive Programming

Edited by

Luc De Raedt[1], Richard Evans[2], Stephen H. Muggleton[3], and Ute Schmid[4]

1 **KU Leuven, BE,** luc.deraedt@cs.kuleuven.be
2 **Google DeepMind – London, GB,** richardevans@google.com
3 **Imperial College London, GB,** s.muggleton@imperial.ac.uk
4 **Universität Bamberg, DE,** ute.schmid@uni-bamberg.de

Abstract

In this report the program and the outcomes of Dagstuhl Seminar 19202 "Approaches and Applications of Inductive Programming" is documented. After a short introduction to the state of the art to inductive programming research, an overview of the introductory tutorials, the talks, program demonstrations, and the outcomes of discussion groups is given.

Seminar May 12–17, 2019 – http://www.dagstuhl.de/19202

2012 ACM Subject Classification Computing methodologies → Artificial intelligence, Computer systems organization → Robotics, Computing methodologies → Parallel programming languages, Software and its engineering → Compilers, Human-centered computing → Human computer interaction (HCI)

Keywords and phrases Enduser programming, Explainable AI, Human-like computing, Inductive logic programming, Probabilistic programming

Digital Object Identifier 10.4230/DagRep.9.5.58

Edited in cooperation with Lidia Contreras-Ochando (Technical University of Valencia, ES), liconoc@upv.es

1 Executive Summary

Ute Schmid
Luc De Raedt

Inductive programming addresses the automated or semi-automated generation of computer programs from incomplete information such as input-output examples, constraints, computation traces, demonstrations, or problem-solving experience [11]. The generated – typically declarative – program has the status of a hypothesis which has been generalized by induction. That is, inductive programming can be seen as a special approach to machine learning. In contrast to standard machine learning, only a small number of training examples is necessary. Furthermore, learned hypotheses are represented as logic or functional programs, that is, they are represented on symbol level and therefore are inspectable and comprehensible [36, 15, 37, 29]. On the other hand, inductive programming is a special approach to program synthesis. It complements deductive and transformational approaches [39, 25, 4]. In cases where synthesis of specific algorithm details that are hard to figure out by humans inductive reasoning can be used to generate program candidates from either user-provided data such as test cases or from data automatically derived from a formal specification [35]. Finally, symbolic approaches can be combined with probabilistic methods [8, 9].

Approaches and Applications of Inductive Programming, *Dagstuhl Reports*, Vol. 9, Issue 5, pp. 58–88
Editors: Luc De Raedt, Richard Evans, Stephen H. Muggleton, and Ute Schmid
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Inductive program synthesis is of interest for researchers in artificial intelligence since the late sixties [2]. On the one hand, the complex intellectual cognitive processes involved in producing program code which satisfies some specification are investigated, on the other hand methodologies and techniques for automating parts of the program development process are explored. One of the most relevant areas of application of inductive programming techniques is end-user programming [5, 22, 6]. For example, the Microsoft Excel plug-in Flashfill synthesizes programs from a small set of observations of user behavior [15, 14, 13]. Related applications are in process mining and in data wrangling [19, 21]. Inductive programming in general offers powerful approaches to learning from relational data [30, 23] and to learning from observations in the context of autonomous intelligent agents [28, 20, 36]. Furthermore, inductive programming can be applied in the context of teaching programming [38, 41].

A recent new domain of interest is how to combine inductive programming with blackbox approaches, especially in the context of (deep) neural networks [10] and in data science.

Relation to Previous Dagstuhl-Seminars

The seminar is a continuation Dagstuhl-Seminars 13502, 15442, and 17382. In the first seminar, the focus was on establishing the research community by exploring the different areas of basic research and applications of inductive programming and identifying commonalities and differences in methods and goals. In the second seminar, more in-depth coverage of algorithmic methods was provided and the relation of inductive programming to cognitive modeling was explored. The third seminar had a main focus on applications in data cleansing, teaching programming, and interactive training. Furthermore, first proposals for neural approaches to learning for inductive programming were presented.

Besides many new insights from many discussions, visible outcomes from the previous seminars are:

- Muggleton, S.H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A. and T. Besold (2019). Ultra-strong machine learning – comprehensibility of programs learned with ILP. Machine Learning, 107(7), 1119–1140.
- Schmid, U., Zeller, C., Besold, T., Tamaddoni-Nezhad, A., & Muggleton, S.H. (2017). How does predicate invention affect human comprehensibility?. In Alessandra Russo and James Cussens, editors, Proceedings of the 26th International Conference on Inductive Logic Programming (ILP 2016), pp. 52-67, Springer.
- Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., & Dowe, D. L. (2016). Computer models solving intelligence test problems: Progress and implications. Artificial Intelligence, 230, 74-107.
- Gulwani, S., Hernández-Orallo, J., Kitzelmann, E., Muggleton, S. H., Schmid, U., & Zorn, B. (2015). Inductive programming meets the real world. *Communications of the ACM*, 58(11), 90-99.
- https://en.wikipedia.org/wiki/Inductive_programming
- NIPS'2016 workshop on Neural Nets and Program Induction involving Stephen Muggleton.
- A collaboration in the context of the EPSRC funded Human-Like Computing Network+ headed by Muggleton (see http://hlc.doc.ic.ac.uk/).

For the fourth seminar, we extend our invitation to researchers from deep learning and to researchers addressing statistical machine learning and probabilistic reasoning. **The focus of the fourth seminar has been on the potential of inductive programming for explainable AI, especially in combination with (deep) neural networks and with data science.**

Inductive Programming as a Approach to Explainable AI

Recently it has been recognized that explainability is crucial for machine learning to be usable in real world domains – especially such where erroneous decisions might be harmful to humans. Consequently, interfaces which explain (aspects) of classifier decisions have been proposed – especially in the context of deep learning [32, 3]. The notion of explainability has already been addressed in the early days of machine learning: During the 1980s Michie defined Machine Learning in terms of two orthogonal axes of performance: predictive accuracy and comprehensibility of generated hypotheses. Since predictive accuracy was readily measurable and comprehensibility not so, later definitions in the 1990s, such as that of Mitchell [27], tended to use a one-dimensional approach to Machine Learning based solely on predictive accuracy, ultimately favouring statistical over symbolic Machine Learning approaches.

In [29] a definition was provided of comprehensibility of hypotheses which can be estimated using human participant trials. Experiments were conducted testing human comprehensibility of logic programs. Results show that participants were not able to learn the relational concept on their own from a set of examples but they were able to apply the relational definition provided by the ILP Metagol system correctly. That is, the results demonstrate that ILP systems can fulfill Michie's criterion of operational effectiveness. The findings also imply the existence of a class of relational concepts which are hard to acquire for humans, though easy to understand given an abstract explanation. We believe improved understanding of this class could have potential relevance to contexts involving human learning, teaching and verbal interaction.

Finally, while research in explanations in the context of neural networks is focusing on visualization, ILP learned classifiers allow natural language explanations. In the Dagstuhl seminar we plan to discuss possibilities for combining deep learning approaches and inductive programming such that both modes of explanations can be generated.

Inductive Programming for Support in Data Science

The success of the inductive programming system FlashFill has motivated the developed of several approaches to using inductive programming in the context of data science, more specifically data wrangling. It is well known that in data science and data mining processes, about 80 per cent of the time goes to selecting the right data, and further pre-processing it so that it be input into data mining software. One important step in that process is data wrangling, which is concerned with cleaning up the data and transforming it across different formats. For this step, inductive programs can be used; various approaches are moving in that direction, e.g. FlashRelate [1], Tacle [19] and SYNTH [7]. Furthermore, workshops are being organised on the topic of automating data wrangling (e.g. at ICDM 2016, http://dmip.webs.upv.es/DWA2016/, at ECMLPKDD 2019, https://sites.google.com/view/autods and at Dagstuhl Seminar 18401), and tools such as MagicHaskeller and JailBreakR being used in this context. In the Dagstuhl seminar, we also want to deepen the link between data wrangling and inductive programming.

Inductive Programming and Neural Computation

The deep learning community has been interested in taking on challenging tasks from the artificial intelligence community. It is therefore no surprise that they have also started to look into inductive and automatic programming. In particular, they have contributed several mixtures of traditional computational models with those of neural networks. For instance, the neural Turing machine [12] integrates a neural network with an external memory and it is able to learn simple algorithms such as copy and sort, the neural program interpreter [31] is a recurrent neural network that learns to represent and execute programs from program traces, while [33] present an end-to-end differentiable interpreter for the programming language Forth and [34, 24] for a declarative Prolog like language. The central goal in these approaches is to obtain an end-to-end differentiable model. While initial results are promising, the approaches still require a lot of data to be trained or need to scale up. This contrasts with the more traditional symbolic approaches to inductive programming and thus a key opportunity is to further cross-fertalize these two approaches.

Inductive Programming and Human-like Computing

The human ability to master complex demands is to a large extend based on the ability to exploit previous experiences. Based on our experience, we are able to predict characteristics or reactions of (natural or man-made, inanimate or animate) objects, we can reason about possible outcomes of actions, and we can apply previously successful routines and strategies to new tasks and problems. In philosophy, psychology and artificial intelligence, researchers proposed that the core process to expand knowledge, that is, construct hypotheses, in such a way that we can transfer knowledge from previous experience to new situations is *inductive* inference [18, 16, 40].

One special aspect of induction is that humans are able to acquire complex, productive rule sets from experience. Following Chomsky, rules are productive when they can be applied in situations of various complexity. Typical examples of such rule sets are knowledge about natural language grammar, recursive concepts such as ancestor and recursive problem solving strategies. For example, if humans have learned how to solve Tower of Hanoi problems with three and four discs, at least some of them are able to generalize the underlying strategy for solving problems with an arbitrary number of discs.

Inductive programming provides mechanisms to generate such productive, recursive rule sets. Examples of recent work on using inductive programming to model this learning-capability of human cognition are [26, 20, 36, 17, 23]. Therefore, it might be fruitful for cognitive scientists to get acquainted with inductive programming as one approach to model the acquisition of complex knowledge structures. On the other hand, knowledge gained from experiments in human problem solving, concept learning and language acquisition can be a source of inspiration for new algorithmic approaches to inductive programming.

Objectives and Expected Outcomes of the Seminar

A long-term objective of the seminar series is to establish inductive programming as a self-contained research topic in artificial intelligence, especially as a field of machine learning and of cognitive modeling. The seminar serves as community building event by bringing together researchers from different areas of inductive programming – especially inductive logic programming and inductive functional programming –, from different application areas such as end-user programming and tutoring, and from cognitive science research, especially from cognitive models of inductive (concept) learning. For successful community building we seek to balance junior and senior researchers and to mix researchers from universities and from industry.

The previous seminars resulted in new collaborations between researchers from different backgrounds as documented in joint publications and we expect that the collaborations will continue, deepen and extend, resulting not only in further joint publications but also in joint research projects.

In the fourth seminar, we continued and extended previous discussions addressing the following aspects:

- Identifying the specific contributions of inductive programming to machine learning research and applications of machine learning, especially identifying problems for which inductive programming approaches are more suited than standard machine learning approaches, including deep learning and probabilistic programming. Focus here is on possibilities of combining (deep) neural approaches or probabilistic programming with (symbolic) inductive programming, especially with respect to new approaches to comprehensibility of machine learned models and on explainable AI.
- Establishing criteria for evaluating inductive programming approaches in comparison to each other and in comparison to other approaches of machine learning and providing a set of benchmark problems.
- Discussing current applications of inductive programming in end-user programming and programming education and identifying further relevant areas of application.
- Establishing stronger relations between cognitive science research on inductive learning and inductive programming under the label of human-like computation.
- Strengthening the relation of inductive programming and data science, especially with respect to data cleansing and data wrangling.

Concluding Remarks and Future Plans

In the wrapping-up section, we decided to move the IP webpage[1] to a Wiki and encouraged all participants to make available their systems, tutorial/lecture slides and publications there.

As the grand IP challenge we came up with 2017 is still up:

> An IP program should invent an algorithm publishable in a serious journal (e.g., an integer factorization algorithm) or win a programming competition!

References

1 Daniel W. Barowy, Sumit Gulwani, Ted Hart, and Benjamin Zorn. Flashrelate: Extracting relational data from semi-structured spreadsheets using examples. *SIGPLAN Not.*, 50(6):218–228, June 2015.

2 A. W. Biermann, G. Guiho, and Y. Kodratoff, editors. *Automatic Program Construction Techniques*. Macmillan, New York, 1984.

3 Alexander Binder, Sebastian Bach, Gregoire Montavon, Klaus-Robert Müller, and Wojciech Samek. Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, pages 913–922. Springer, 2016.

4 Rastislav Bodik and Emina Torlak. Synthesizing programs with constraint solvers. In *CAV*, page 3, 2012.

5 A. Cypher, editor. *Watch What I Do: Programming by Demonstration.* MIT Press, Cambridge, MA, 1993.

[1] www.inductive-programming.org

6 Allen Cypher, Mira Dontcheva, Tessa Lau, and Jeffrey Nichols, editors. *No Code Required: Giving Users Tools to Transform the Web.* Elsevier, 2010.

7 Luc De Raedt, Hendrik Blockeel, Samuel Kolb, Stefano Teso, and Gust Verbruggen. Elements of an automatic data scientist. In *International Symposium on Intelligent Data Analysis*, pages 3–14. Springer, 2018.

8 Luc De Raedt and Angelika Kimmig. Probabilistic (logic) programming concepts. *Machine Learning*, 100(1):5–47, 2015.

9 Richard Evans and Edward Grefenstette. Learning explanatory rules from noisy data. *Journal of Artificial Intelligence Research*, 61:1–64, 2018.

10 Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. Can neural networks understand logical entailment? *arXiv preprint arXiv:1802.08535*, 2018.

11 P. Flener and U. Schmid. Inductive programming. In C. Sammut and G. Webb, editors, *Encyclopedia of Machine Learning*, pages 537–544. Springer, 2010.

12 Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

13 Sumit Gulwani. Automating string processing in spreadsheets using input-output examples. In *38th Symposium on Principles of Programming Languages*. ACM, 2011.

14 Sumit Gulwani, William R. Harris, and Rishabh Singh. Spreadsheet data manipulation using examples. *Communications of the ACM*, 55(8):97–105, 2012.

15 Sumit Gulwani, José Hernández-Orallo, Emanuel Kitzelmann, Stephen H. Muggleton, Ute Schmid, and Benjamin G. Zorn. Inductive programming meets the real world. *Communications of the ACM*, 58(11):90–99, 2015.

16 Ulrike Hahn, Todd M Bailey, and Lucy BC Elvin. Effects of category diversity on learning, memory, and generalization. *Memory & Cognition*, 33(2):289–302, 2005.

17 J. Hernández-Orallo, D. L. Dowe, and M. V. Hernández-Lloreda. Universal psychometrics: Measuring cognitive abilities in the machine kingdom. *Cognitive Systems Research*, 27(0):50–74, 2014.

18 J.H. Holland, K.J. Holyoak, R.E. Nisbett, and P.R. Thagard. *Induction – Processes of Inference, Learning, and Discovery.* MIT Press, Cambridge, MA, 1986.

19 Samuel Kolb, Sergey Paramonov, Tias Guns, and Luc De Raedt. Learning constraints in spreadsheets and tabular data. *Machine Learning*, 106(9-10):1441–1468, 2017.

20 P. Langley and D. Choi. A unified cognitive architecture for physical agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, Boston, MA, 2006. AAAI Press.

21 Vu Le and Sumit Gulwani. Flashextract: A framework for data extraction by examples. *ACM SIGPLAN Notices*, 49(6):542–553, 2014.

22 Henry Lieberman, editor. *Your Wish is My Command: Programming by Example.* Morgan Kaufmann, San Francisco, 2001.

23 D. Lin, E. Dechter, K. Ellis, J.B. Tenenbaum, and S.H. Muggleton. Bias reformulation for one-shot function induction. In *Proceedings of the 23rd European Conference on Artificial Intelligence (ECAI 2014)*, pages 525–530, Amsterdam, 2014. IOS Press.

24 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 3749–3759. Curran Associates, Inc., 2018.

25 Zohar Manna and Richard Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems*, 2(1):90–121, 1980.

26 G. F. Marcus. *The Algebraic Mind. Integrating Conncetionism and Cognitive Science.* Bradford, Cambridge, MA, 2001.

27 Tom Mitchell. *Machine learning*. McGraw Hill, 1997.

28 Bogdan Moldovan, Plinio Moreno, Davide Nitti, José Santos-Victor, and Luc De Raedt. Relational affordances for multiple-object manipulation. *Auton. Robots*, 42(1):19–44, 2018.

29 S.H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold. Ultra-strong machine learning – comprehensibility of programs learned with ILP. *Machine Learning*, 2018.

30 Stephen H. Muggleton, Dianhuan Lin, and Alireza Tamaddoni-Nezhad. Meta-interpretive learning of higher-order dyadic datalog: predicate invention revisited. *Machine Learning*, 100(1):49–73, 2015.

31 Scott E. Reed and Nando de Freitas. Neural programmer-interpreters. *CoRR*, abs/1511.06279, 2015.

32 Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

33 Sebastian Riedel, Matko Bosnjak, and Tim Rocktäschel. Programming with a differentiable forth interpreter. *CoRR*, abs/1605.06640, 2016.

34 Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. *CoRR*, abs/1705.11040, 2017.

35 Reudismam Rolim, Gustavo Soares, Loris D'Antoni, Oleksandr Polozov, Sumit Gulwani, Rohit Gheyi, Ryo Suzuki, and Bjoern Hartmann. Learning syntactic program transformations from examples. *arXiv preprint arXiv:1608.09000*, 2016.

36 Ute Schmid and Emanuel Kitzelmann. Inductive rule learning on the knowledge level. *Cognitive Systems Research*, 12(3):237–248, 2011.

37 Ute Schmid, Christina Zeller, Tarek Besold, Alireza Tamaddoni-Nezhad, and Stephen Muggleton. How does predicate invention affect human comprehensibility? In *International Conference on Inductive Logic Programming*, pages 52–67. Springer, 2016.

38 Rishabh Singh, Sumit Gulwani, and Armando Solar-Lezama. Automated feedback generation for introductory programming assignments. *ACM SIGPLAN Notices*, 48(6):15–26, 2013.

39 Douglas R. Smith. The synthesis of LISP programs from examples: A survey. In *Automatic Program Construction Techniques*, pages 307–324. Macmillan, 1984.

40 J. Tenenbaum, T.L. Griffiths, and C. Kemp. Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, 10(7):309–318, 2006.

41 Christina Zeller and Ute Schmid. Automatic generation of analogous problems to help resolving misconceptions in an intelligent tutor system for written subtraction. In *Proceedings of the Workshop on Computational Analogy at the 24th International Conference on Case Based Reasoning (ICCBR 2016, Atlanta, GA, 31th October to 2nd November 2016)*, 2016.

2 Table of Contents

Executive Summary

Introductory Talks

Overview of Talks

Discussion groups

3 Introductory Talks

3.1 Introduction to Inductive Functional Programming

Ute Schmid (Universität Bamberg, DE)

Inductive programming (IP) is an area of research concerned with learning recursive programs, typically in some declarative language, from small sets of input/output examples. Since a general program is induced from examples, IP can be seen as a special case of machine learning. In contrast to standard machine learning, induced program hypotheses are required to cover all given examples correctly. In the talk I will give a general introduction to the research area of IP and then focus on induction of functional programs. First, I will present Thesys – the classical approach for learning Lisp programs by regularity detection. Afterwards, I will present the more recent approach IgorII. IgorII has been the first approach realizing necessary function invention on the fly. It integrates the concept of structure-guided program induction from Thesys, concepts of contemporary programming languages, such as pattern matching, as well as concepts proposed in Inductive Logic Programming (ILP), especially allowing to take into account background knowledge. I will point out relations of IP to cognitive models of learning, demonstrating that IgorII can be applied out of the box to learning the recursive rule sets for problems such as Tower of Hanoi, language parsing, and number series induction. Finally, I will present arguments for IP as a highly expressive approach to interpretable ML.

References

1 Gulwani, S., Hernández-Orallo, J., Kitzelmann, E., Muggleton, S., Schmid, U., and Zorn, B. (2015). Inductive Programming Meets the Real World, *Communications of the ACM, 58(11)*, 90-99.

2 Hernández-Orallo, J., Martínez-Plumed, F., Schmid, U., Siebers, M., Dowe, D.L. (2016). Computer Models Solving Intelligence Test Problems: Progress and Implications. *Artificial Intelligence, Vol. 230*, 74–107.

4 Overview of Talks

4.1 Declarative Compression of Imperative Programs

Eli Bingham (Uber AI Labs – San Francisco, US)

Joint work of Eli Bingham, Fritz Obermeyer, Noah D. Goodman

Standard approaches to program synthesis in general-purpose imperative languages like Python often struggle with ambiguous or inconsistent examples or produce large deterministic programs which are difficult to interpret. These limitations can be overcome in principle by adding randomness or other forms of nondeterminism to the language, but support for nondeterminism is limited in existing synthesis tools and the resulting programs may only approximately solve the tasks at hand.

To bridge this gap, we introduce program merging, a simple algorithm for finding short nondeterministic programs that approximate the behavior of a deterministic imperative program. We define a family of compressive program transformations that refactor exactly duplicated program fragments into new subroutines or replace approximately duplicated program fragments with declarative approximations such as random variables, constraints, or learnable parameters. Program merging searches for sequences of such transformations applied to an initial program that maximize the posterior probability of the final program given the original examples or specification.

The resulting programs may mix multiple intractable declarative computations (in the form of nested sums over multiple semirings). As a step towards principled and efficient approximate evaluation of such programs, we introduce Funsor, a Python-embedded domain-specific language for mixed-mode declarative programming with an initial focus on approximate probabilistic inference and gradient-based optimization.

4.2 Difficulty of problems for humans

Ivan Bratko (University of Ljubljana, SI)

Joint work of Ivan Bratko, Matej Guid, Dayana Hristova, Simon Stoiljkovikj

Some questions of interest for Explainable AI are: (1) How difficult a given problem is for humans, and how humans would typically tackle the problem? (2) How can we measure the comprehensibility of a theory or of an explanation? In this talk we discuss one approach to automatic prediction of difficulty for humans of problems that are solved through informed search. Although there are many applications that require prediction of difficulty, for example intelligent tutoring systems, in our case the motivation came from a computer analysis of the quality of play in chess games played by human world chess champions. This analysis aimed at answering questions like who was the best chess player of all time. The quality of play was estimated on the basis of the differences between moves played by humans and moves played in the same positions by a chess playing engine taken as the golden standard. Different chess champions tended to play in different styles: some tended towards simple, quiet and safe positions, while others preferred complicated, aggressive and risky game. As it is easier to play correctly in simple positions than in complex positions, our evaluation method, to be fair, also had to take into account the difficulty of individual positions that occurred in champions games. Our method of assessing the difficulty of chess positions was based on the amount of search needed by the chess engine to find the best move. This enabled a fair comparison of players with different playing styles, and to answer questions like: How well would a player like Capablanca, known for his tendency to simplify the game, do if he played in the style of Tal, known for his tendency toward extremely complicated positions. Although this approach was generally successful in the comparison of world champions, it was later found inappropriate for estimating the difficulty of particular type of positions, called tactical chess positions. In solving tactical positions, humans use pattern-based knowledge to guide their search extremely effectively. As this knowledge has never been sufficiently formalized for use in a chess program, a refinement was needed to simulate human search by a chess program without access to humans' pattern-based tactical knowledge.

References

1 Ivan Bratko, Dayana Hristova, Matej Guid. *Search vs. Knowledge in Human Problem Solving: A Case Study in Chess.* Model-Based Reasoning in Science and Technology: Logical, Epistemological and Cognitive Issues, pp. 569-584. Springer 2016. SAPERE Book Series (Studies in Applied Philosophy, Epistemology and Rational Ethics).

2 Simon Stoiljkovikj, Ivan Bratko, Matej Guid. *A Computational Model for Estimating the Difficulty of Chess Problems.* Proceedings of the Third Annual Conference on Advances in Cognitive Systems, ACS-2015 (Article 7), Cognitive Systems Foundations.

4.3 Trace-Based Programming Method

Maurice Chandoo (Leibniz Universität Hannover, DE)

Main reference Maurice Chandoo: “A Systematic Approach to Programming”, CoRR, Vol. abs/1808.08989, 2018.
URL https://arxiv.org/abs/1808.08989

We describe a programming method for systematically implementing sequential algorithms in any imperative or functional programming language. The method is based on the premise that it is easy to write down how an algorithm proceeds on a concrete input. This information–which we call execution trace–is used as a starting point to derive the desired program. In contrast to test-driven development the program is directly constructed from the test cases instead of written separately. The program's operations, control flow and predicates guiding the control flow are worked out separately, which saves the programmer from having to think about them simultaneously. This reduces the likelihood of introducing errors. We demonstrate our method for two examples and discuss its utility. The method only requires pen and paper. However, a computer can facilitate its usage by taking care of certain trivial tasks.

Outline of the method. The execution trace of an algorithm can be seen as a table where each column corresponds to a variable. The i-th row contains the values of the variables after i execution steps of the algorithm for a certain input. The first step is to determine the set of variables used by the algorithm. Then an input is chosen and an execution trace for this input is written down. The next step is to generalize the literals in this table. More specifically, one has to determine how each value in row $i+1$ can be expressed in terms of the values in row i. After removing the literals from the table, the sequence of expressions that is left in each row corresponds to an operation executed by the algorithm. Certain rows correspond to the same operation. This information can be used to derive the control flow graph of the desired program. Assume the i-th row corresponds to the operation $\alpha(i)$. Then the graph has a vertex for each operation and for each i there is an edge from $\alpha(i)$ to $\alpha(i+1)$. Stated differently, the sequence of operations in the table describes a path through the control flow graph. By repeating the previous steps for various inputs one will eventually arrive at the complete control flow graph of the program. Finally, it remains to determine the edge predicates, i.e. under what circumstances does the program move from one program state to the next.

The method works in such a way that the constructed program is consistent with all execution traces that were used to build it. A one-to-one correspondence between program states and operations is assumed. Intuitively, this means each line of code in a program must be unique. While this is not necessarily the case, this can always be achieved by adding a state variable.

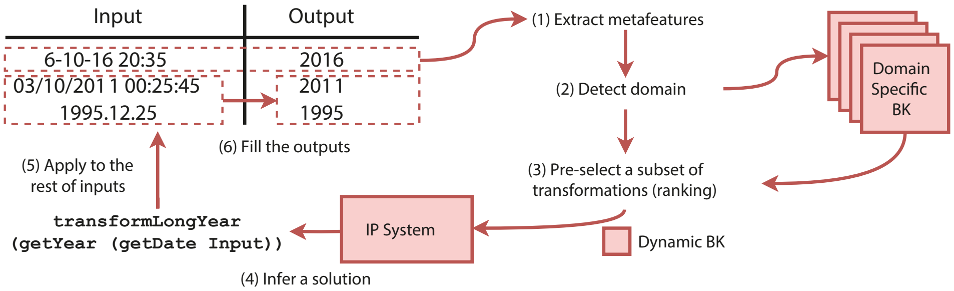

Figure 1 Automating data wrangling with IP: process example. The first row (Input and Output) is used as an input example for the IP system. The function returned is applied to the rest of the instances to obtain the outputs.

4.4 Automated Data Transformation with Inductive Programming and Dynamic Background Knowledge

Lidia Contreras-Ochando (Technical University of Valencia, ES)

Joint work of Lidia Contreras-Ochando, Cesar Ferri Ramirez, José Hernández-Orallo, Fernando Martínez-Plumed, María-José Ramírez-Quintana, Susumu Katayama

Main reference Lidia Contreras-Ochando, Cesar Ferri Ramirez, José Hernández-Orallo, Fernando Martínez-Plumed, María-José Ramírez-Quintana, Susumu Katayama: "Automated data transformation with inductive programming and dynamic background knowledge". In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2019 (to appear). ECML-PKDD'19 (2019)

Data quality is vital for machine learning and data science. Despite the growing amount of data preparation tools, the most tedious data wrangling activities and feature manipulation are still partially resistant to automation because they depend heavily on domain information. For example, if the strings "17th of August of 2017" and "2017-08-17" are to be formatted into "08/17/2017" to be correctly recognised by a data science tool, this is generally processed in two phases: (1) they are recognised as dates and (2) some conversions are applied specific to the date domain. The dates manipulating processes, however, are very distinct from those for manipulating addresses or phone numbers. This needs enormous quantities of background knowledge, which generally becomes a bottleneck as domain and format diversity grows. We assist to relieve this issue by using inductive programming (IP) with dynamic background knowledge (BK) fuelled by a machine learning meta-model that chooses the domain, the primitives (or both) from some descriptive features of the data wrangling problem (see Figure 1). We demonstrate this for the automation of data transformation and we evaluate the approach on an integrated benchmark for data wrangling, which we share publicly for the community.

Acknowledgments. This research was supported by the EU (FEDER) and the Spanish MINECO (RTI2018-094403-B-C32), Universitat Politècnica de València (PAID-06-18), and the Generalitat Valenciana (PROMETEO/2019/098 and BEST/2018/027). L. Contreras-Ochando was also supported by the Spanish MECD (FPU15/03219). J. Hernández-Orallo is also funded by FLI (RFP2-152).

References

1 Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., Ramírez-Quintana, M.J., Katayama, S.: Automated data transformation with inductive programming and dynamic background knowledge. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2019 (to appear). ECML-PKDD '19 (2019)

2 Contreras-Ochando, L.: DataWrangling-DSI: BETA – Extended Results (2019). 10.5281/zenodo.2557385

3 Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., Ramírez-Quintana, M.J., Katayama, S.: General-purpose declarative inductive programming with domain-specific background knowledge for data wrangling automation. arXiv preprint arXiv:1809.10054 (2018)

4.5 Inductive general game playing

Andrew Cropper (University of Oxford, GB), Richard Evans (Google DeepMind – London, GB), and Mark Law (Imperial College London, GB)

Main reference Andrew Cropper, Richard Evans, Mark Law: "Inductive general game playing", CoRR, Vol. abs/1906.09627, 2019.
URL https://arxiv.org/abs/1906.09627

General game playing (GGP) is a framework for evaluating an agent's general intelligence across a wide range of tasks. In the GGP competition, an agent is given the rules of a game (described as a logic program) that it has never seen before. The task is for the agent to play the game, thus generating game traces. The winner of the GGP competition is the agent that gets the best total score over all the games. In this paper, we invert this task: a learner is given game traces and the task is to learn the rules that could produce the traces. This problem is central to *inductive general game playing* (IGGP). We introduce a technique that automatically generates IGGP tasks from GGP games. We introduce an IGGP dataset which contains traces from 50 diverse games, such as *Sudoku*, *Sokoban*, and *Checkers*. We claim that IGGP is difficult for existing inductive logic programming (ILP) approaches. To support this claim, we evaluate existing ILP systems on our dataset. Our empirical results show that most of the games cannot be correctly learned by existing systems. The best performing system solves only 40% of the tasks perfectly. Our results suggest that IGGP poses many challenges to existing approaches. Furthermore, because we can automatically generate IGGP tasks from GGP games, our dataset will continue to grow with the GGP competition, as new games are added every year. We therefore think that the IGGP problem and dataset will be valuable for motivating and evaluating future research.

4.6 Playgol: learning programs through play

Andrew Cropper (University of Oxford, GB)

Main reference Andrew Cropper: "Playgol: learning programs through play", CoRR, Vol. abs/1904.08993, 2019.
URL https://arxiv.org/abs/1904.08993

Children learn though play. We introduce the analogous idea of *learning programs through play*. In this approach, a program induction system (the learner) is given a set of tasks and initial background knowledge. Before solving the tasks, the learner enters an *unsupervised playing* stage where it creates its own tasks to solve, tries to solve them, and saves any solutions (programs) to the background knowledge. After the playing stage is finished, the learner enters the *supervised building* stage where it tries to solve the user-supplied tasks and can reuse solutions learnt whilst playing. The idea is that playing allows the learner to discover reusable general programs on its own which can then help solve the user-supplied tasks. We claim that playing can improve learning performance. We show that playing can reduce the textual complexity of target concepts which in turn reduces the sample complexity of a learner. We implement our idea in *Playgol*, a new inductive logic programming system. We experimentally test our claim on two domains: robot planning and real-world string transformations. Our experimental results suggest that playing can substantially improve learning performance. We think that the idea of playing (or, more verbosely, *unsupervised bootstrapping for supervised program induction*) is an important contribution to the problem of developing program induction approaches that self-discover BK.

4.7 Inductive Programming and Constraint Learning for Automated Data Science

Luc De Raedt (KU Leuven, BE)

Joint work of Luc De Raedt, Hendrik Blockeel, Samuel Kolb, Andrea Passerini, Stefano Teso, Gust Verbruggen
Main reference Luc De Raedt, Hendrik Blockeel, Samuel Kolb, Stefano Teso, Gust Verbruggen: "Elements of an Automatic Data Scientist", in Proc. of the Advances in Intelligent Data Analysis XVII – 17th International Symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, October 24-26, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 11191, pp. 3–14, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-01768-2_1

The SYNTH approach on automated data science was presented. It is centred around a simple but non-trivial "autocompletion" setting for automating data science. Given are a set of worksheets in a spreadsheet and the goal is to automatically complete some values. The SYNTH approach has several components that are based on inductive programming. This includes the data wrangling work (that was presented by Gust Verbruggen, this volume) and the work on constraint learning. This talk focussed especially on inductively learning constraints from examples. Although constraints are ubiquitous in artificial intelligence, there are only few approaches that learn them.

References

1 De Raedt, Luc, Hendrik Blockeel, Samuel Kolb, Stefano Teso, and Gust Verbruggen. "Elements of an Automatic Data Scientist." In International Symposium on Intelligent Data Analysis, LNCS 11191, pp. 3-14. Springer, Cham, 2018.

2 Verbruggen, Gust, and De Raedt, Luc. Automatically wrangling spreadsheets into machine learning data formats. In International Symposium on Intelligent Data Analysis, LNCS 11191, pp. 367-379. Springer, Cham, 2018.

3 De Raedt, Luc, Passerini, Andrea and Teso, Stefano. Learning constraints from examples. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018

4.8 Contrastive Explanations: Status and Future

Amit Dhurandhar (IBM TJ Watson Research Center – Yorktown Heights, US)

Main reference Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, Payel Das: "Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives", CoRR, Vol. abs/1802.07623, 2018.
URL https://arxiv.org/abs/1802.07623

With the widespread adoption of AI technologies across society, explainability as a tool to understand and build trustworthy systems has become an endeavor of utmost importance. This has implications in building systems that are not only robust but also fair and accountable. In my talk I will present our recent work on contrastive explanations and along with future directions. An explanation being contrastive is considered to be the most important aspect of an explanation [1] followed by it being selective. Our method is able to achieve both where we generate contrastive explanations that are also sparse. We show how to generate these explanations for images as well as tabular data for complex models such as deep neural networks. We then generalize our approach to be also applicable in model agnostic settings. Generating such explanations for text is part of future work given that we want to change a piece of text by minimal amount, yet create semantically correct text that lies in a different class. An extremely interesting and ambitious direction is to learn boolean features using our explanations and create a simple model using them that can potentially replicate a complex models performance.

References

1 Christoph Molnar. *Interpretable Machine Learning*. Creative Commons License, USA, 2019.

2 Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam, Payel Das. *Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives*. Advances of Neural Inf. Proc. Systems, 2018.

3 Ronny Luss, Pin-Yu Chen, Amit Dhurandhar, Prasanna Sattigeri, Chun-Chen Tu and Karthikeyan Shanmugam. *Generating Contrastive Explanations with Monotonic Attribute Functions*. arxiv, 2019.

4 Amit Dhurandhar, Tejaswini Pedapati, Avinash Balakrishnan, Pin-Yu Chen, Karthikeyan Shanmugam and Ruchir Puri. *Model Agnostic Contrastive Explanations Method for Structured Data*. arxiv, 2019.

4.9 Apperception

Richard Evans (Google DeepMind – London, GB) and José Hernández-Orallo (Technical University of Valencia, ES)

This work is an attempt to answer a central question in unsupervised learning: what does it even mean to "make sense" of a sensory sequence? Imagine a machine, equipped with sensors, receiving a stream of sensory information. It must, somehow, make sense of this stream of sensory data. But what does it mean, exactly, to "make sense" of sensory data? We have an intuitive understanding of what is involved in making sense of the sensory stream – but can we specify precisely what is involved? Can this intuitive notion be formalized?

In this paper, we make two contributions. First, we provide a precise formalization of what it means to "make sense" of a sensory sequence. According to our definition, making sense of a sensory sequence involves constructing a symbolic causal theory that explains the sensory sequence and satisfies a set of unity conditions that were inspired by Kant's discussion of the "synthetic unity of apperception" in the Critique of Pure Reason. According to our interpretation, making sense of sensory input is a type of program synthesis, but it is unsupervised program synthesis.

Our second contribution is a computer implementation, the Apperception Engine, that was designed to satisfy our requirements for making sense of a sensory sequence. Our system is able to produce interpretable human-readable causal theories from very small amounts of data, because of the strong inductive bias provided by the Kantian unity constraints. A causal theory produced by our system is able to predict future sensor readings, as well as retrodict earlier readings, and "impute" (fill in the blanks of) missing sensory readings. In fact, it is able to do all three tasks simultaneously.

We tested the engine in a diverse variety of domains, including cellular automata, rhythms and simple nursery tunes, multi-modal binding problems, occlusion tasks, and sequence induction IQ tests. In each domain, we test our engine's ability to predict future sensor values, retrodict earlier sensor values, and impute missing sensory data.The Apperception Engine performs well in all these domains, significantly out-performing neural net baselines. These results are significant because neural nets typically struggle to solve the binding problem (where information from different modalities must somehow be combined together into different aspects of one unified object) and fail to solve occlusion tasks (in which objects are sometimes visible and sometimes obscured from view). We note in particular that in the sequence induction IQ tasks, our system achieved human-level performance. This is notable because the Apperception Engine was not designed to solve these IQ tasks; it is not a bespoke hand-engineered solution to this particular domain. Rather, it is a general purpose apperception system for making sense of *any* sensory sequence, that just happens to be able to solve these IQ tasks out of the box.

4.10 Machine Teaching with P3

Cesar Ferri Ramirez (Technical University of Valencia, ES) and José Hernández-Orallo (Technical University of Valencia, ES)

Joint work of Jan Arne Telle, José Hernández-Orallo, Cèsar Ferri
Main reference Jan Arne Telle, José Hernández-Orallo, Cèsar Ferri: "The teaching size: computable teachers and learners for universal languages", Machine Learning, Vol. 108(8-9), pp. 1653–1675, 2019.
URL http://dx.doi.org/10.1007/s10994-019-05821-2

Machine Teaching is a problem based on finding a minimal subset of examples (or witness set) from which a learner can induce a concept. Traditionally, machine teaching techniques employ the notion of "teaching dimension". This dimension is linked to a concept and is defined as the minimum cardinality of a witness set for learning the concept. In this work, we present some limitations of the notion of teaching dimension, and, motivated by these drawbacks, we introduce the definition of "teaching size" that expresses the the size of the shortest witness set needed to identify the concept. We study some theoretical properties of this definition in the general machine teaching framework. Experiments over as on a simple Turing-complete language are used to show important differences between the teaching dimension and the teaching size. Concretely, we use P3, a version of P", a primitive programming language introduced in 1964 by Corrado Böhm. P3 is Turing complete and uses just 7 instructions. We present some interesting results about the relationship between the teaching dimension and the teaching size when inducing simple concepts in the P3 language.

Acknowledgments. This research was supported by the EU (FEDER) and the Spanish MINECO (RTI2018-094403-B-C32), Universitat Politècnica de València (PAID-06-18), and the Generalitat Valenciana (PROMETEO/2019/098 and BEST/2018/027). J. Hernández-Orallo is also funded by FLI (RFP2-152).

4.11 A potpourri of things we've worked on that at some point seemed to be relevant for this workshop.

Johannes Fürnkranz (TU Darmstadt, DE)

Joint work of Johannes Fürnkranz, Eneldo Loza Mencía, Tomáš Kliegr

Having not been active in inductive programming for several years, I arrived at this workshop with the predisposition to listen and learn. I was surprised to find that several discussion directly related to work that we have been doing in the past, and gave a spontaneous talk that summarized some of these results. The issues that we touched upon included inductive programming, interpretability and rule learning, learning in games, and multi-label rule learning.

To inductive programming, I could not add much, but was surprised that an ancient, long-forgotten paper of mine popped up during dinner conversations [1].

Interpretability is currently a very fashionable topic in machine learning. Our work particularly focuses on the aspect that although rules are inherently comprehensible, in the sense that they can be read and interpreted, their interpretability should nevertheless not be taken for granted and requires a deeper investigation. In particular, we have challenged the view that shorter rules should be preferred and argued instead that in many cases, longer rules may yield better explanations, even in cases when the discriminative power of both rules is approximately the same [8]. We have shown in a crowd-sourcing study that humans do not necessarily prefer shorter rules [3], and have argued for an interpretability bias in rule learning algorithms [2].

In multi-label rule learning, the key challenge is that labels can occur both in the head of the rules, but also as potential inputs in the body of the rules [5]. This gives rise to several interesting challenges that appear in similar form in ILP, such as multi-predicate learning and the challenge of learning mutually recursive rules. Our work so far includes adaptations of the covering strategy [4] as well as pruning techniques for efficiently determining the best multi-label head to a rule body [7].

Finally, a significant portion of the discussion revolved about chess endgame and strategy learning as a test bed for inductive programming, which reminded me of (unpublished) work I did in the mid-90s. Since then, we have continued to work on strategy learning in chess, albeit not relying on IP. For example, we have attempted to learn evaluation functions for players of different strengths, where some of the qualitative results seemed reasonable (important positional features received more recognition by stronger players, whereas material values were approximately evaluated the same), but the overall playing strength was found to be dominated by the search algorithm [6]. The adaptation of search strategies to different playing strengths is still an open problem. In another work, we used preference learning to learn evaluation functions from annotated chess games [9].

References

1 Johannes Fürnkranz: Dimensionality reduction in ILP: A call to arms. In L. De Raedt and S. Muggleton (eds.) Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming, pp. 81–86, Nagoya, Japan, 1997.

2 Johannes Fürnkranz, Tomáš Kliegr: The Need for Interpretability Biases. Proc. IDA 2018: 15-27

3 Johannes Fürnkranz, Tomáš Kliegr, Heiko Paulheim: On Cognitive Preferences and the Interpretability of Rule-based Models. CoRR abs/1803.01316 (2018)

4 Eneldo Loza Mencía, Frederik Janssen: Learning rules for multi-label classification: a stacking and a separate-and-conquer approach. Machine Learning 105(1): 77-126 (2016)

5 Eneldo Loza Mencía, Johannes Fürnkranz, Eyke Hüllermeier, Michael Rapp: Learning Interpretable Rules for Multi-label Classification. In Escalante H. J., Escalera S., Guyon I., Baró X., Güclütürk Y., Güclü U., van Gerven M. A. J., van Lier R. (eds.) Explainable and Interpretable Models in Computer Vision and Machine Learning, Springer Series on Challenges in Machine Learning, Springer-Verlag, 2018

6 Philipp Paulsen and Johannes Fürnkranz: A moderately successful attempt to train chess evaluation functions of different strengths. In Proceedings of the ICML-10 Workshop on Machine Learning in Games, Haifa, Israel, 2010.

7 Michael Rapp, Eneldo Loza Mencía, Johannes Fürnkranz: Exploiting Anti-monotonicity of Multi-label Evaluation Measures for Inducing Multi-label Rules. Proc. PAKDD (1) 2018: 29-42

8 Julius Stecher, Frederik Janssen, Johannes Fürnkranz: Shorter Rules Are Better, Aren't They? Proc. DS 2016: 279-294

9 Christian Wirth, Johannes Fürnkranz: On Learning From Game Annotations. IEEE Trans. Comput. Intellig. and AI in Games 7(3): 304-316 (2015)

4.12 What is (Machine) Teaching with Universal Languages? PbE, IP or more?

José Hernández-Orallo (Technical University of Valencia, ES)

Joint work of José Hernández-Orallo, Jan Arne Telle
Main reference José Hernández-Orallo, Jan Arne Telle: "Finite Biased Teaching with Infinite Concept Classes", CoRR, Vol. abs/1804.07121, 2018.
URL http://arxiv.org/abs/1804.07121

One classical view of Inductive Logic Programming (ILP) places it at the intersection between Logic and Learning, while Inductive Programming (IP) would be at the intersection between Programming and Learning. In both cases, there are two remarkable advantages over some other inductive techniques and general machine learning methods: learning can work from very few examples and the models are interpretable. However, when we look at the range of learning scenarios, they go from the common situation where there is no control of the data by the learner, as in many machine learning applications, to situations where the learner has some control (active learning and reinforcement learning), but also to situations where there is a teacher that can select these examples. (Note that learning by demonstration can fall in any of these three categories). Interestingly, when we have a teacher (or user) that wants to demonstrate a concept or a procedure (to a machine), we have a teaching situation, and if coupled with a representation based on programs, we end up in the area of Programming by Example (PbE), the learner is able to write the program that has the behaviour that is shown by a few examples carefully selected by a teacher (or user). In the presentation I gave a short introduction to the area of machine teaching, which has derived theoretical results about the efficiency of teaching in terms of the teaching dimension, defined as the minimum number of examples that are required so that the learner identifies the concept unequivocally. I argued that in the machine teaching field there were no results for rich languages, defined as those that manipulate structured objects and recursions, which are those that are common in PbE, ILP and IP. I then presented the key ideas of a recent paper [1] that shows results for Turing Machines and Finite State Automata for the expected value of a variant of the teaching dimension that assumes a strong learning prior based on simplicity and a similar sampling prior for the teacher. We make the priors explicit and show why they are important, arguing that we will not teach computers efficiently if we cannot align priors. As potential applications, we do not need to consider both learner and teacher as machines. So we could look at teaching from a more general perspective: theoretical or computational teaching rather than machine teaching. If the learner is a machine and the teacher is a human, we have a common situation where humans have to teach computers, and in the case of inductive programming this is fully identified with PbE, where we can now better understand what examples the human should select and why. If the learner is a human and the teacher is a machine, we are in an explanation situation, and this can be used to understand how key examples must be chosen so that the human can identify the concept or behaviour that is hidden behind the machine. Finally, some of the problems that we found appear because we ignore the size of the examples, which finally leads us to the new notion of teaching size, introduced in a different paper [2] and presentation.

Acknowledgments. This research was supported by the EU (FEDER) and the Spanish MINECO (RTI2018-094403-B-C32), Universitat Politècnica de València (PAID-06-18), and the Generalitat Valenciana (PROMETEO/2019/098 and BEST/2018/027). J. Hernández-Orallo is also funded by FLI (RFP2-152).

References

1 José Hernández-Orallo and Jan Arne Telle: "Finite Teaching in Expectation with Infinite Concept Classes", 2019

2 Jan Arne Telle, José Hernández-Orallo and Cesar Ferri: "The Teaching Size: Computable Teachers and Learners for Universal Languages", 2019

4.13 Can Meta-Interpretive Learning outperform Deep Reinforcement Learning of Evaluable Game Strategies?

Céline Hocquette (Imperial College London, GB) and Stephen H. Muggleton (Imperial College London, GB)

Main reference Stephen H. Muggleton, Céline Hocquette: "Machine Discovery of Comprehensible Strategies for Simple Games Using Meta-interpretive Learning", New Generation Comput., Vol. 37(2), pp. 203–217, 2019.
URL https://doi.org/10.1007/s00354-019-00054-2

World-class human players have been outperformed in a number of complex two person games (Go, Chess, Checkers) by Deep Reinforcement Learning systems. However, owing to tractability considerations minimax regret of a learning system cannot be evaluated in such games. In this work we consider simple games (Noughts-and-Crosses and Hexapawn) in which minimax regret can be efficiently evaluated. We use these games to compare Cumulative Minimax Regret for variants of both standard and deep reinforcement learning against two variants of a new Meta-Interpretive Learning system called MIGO. In our experiments all tested variants of both normal and deep reinforcement learning have worse performance (higher cumulative minimax regret) than both variants of MIGO on Noughts-and-Crosses and Hexapawn. Additionally, MIGO's learned rules are relatively easy to comprehend, and are demonstrated to achieve significant transfer learning in both directions between Noughts-and-Crosses and Hexapawn.

4.14 DreamCoder: Growing Deep Domain Expertise with Wake/Sleep Program Learning

Kevin Ellis (MIT – Cambridge, US)

Joint work of Catherine Wong, Max Nye, Mathias Sablé Meyer, Lucas Morales, Luke Hewitt, Joshua B. Tenenbaum, Armando Solar-Lezama

Human domain expertise hinges upon both explicit, declarative concepts and implicit, procedural skill in deploying those concepts to solve new problems. We present a computational model that jointly acquires both these kinds of domain expertise. The model represents solutions to problems as programs, meaning that its domain-expertise takes the form of knowledge about how to write code. It learns this domain expertise through a wake/sleep algorithm, alternating between writing code (during waking) and improving its declarative knowledge (during sleep) by growing out a library of reused library routines. During sleep the model also trains a neural network on randomly generated programs, or "dreams", where the network is trained to guide program search, capturing features of implicit procedural skill in writing code. The learned library routines build on one another, forming a deep hierarchy of explicit declarative knowledge.

4.15 Facilitating research on explainability of (RDF) rule learning: Interactive software systems and crowdsourcing studies

Tomáš Kliegr (University of Economics – Prague, CZ)

Joint work of TomášKliegr, Václav Zeman, Stanislav Vojir, Vojtech Svátek, Jaroslav Kuchar, Stepan Bahnik, Heiko Paulheim
Main reference Stanislav Vojír, Vaclav Zeman, Jaroslav Kuchar, Tomás Kliegr: “EasyMiner.eu: Web framework for interpretable machine learning based on rules and frequent itemsets”, Knowl.-Based Syst., Vol. 150, pp. 111–115, 2018.
URL https://doi.org/10.1016/j.knosys.2018.03.006

This talk covered several interactive academic machine learning software systems based on association rule learning, which is an algorithmic approach for generating exhaustive sets of rules from data that meet user-specified quality requirements. We also report on algorithmic research aimed at reduction of size of rule models, and on cognitive experiments focused on understanding interpretability of rules discovered with association rule learning. Finally, subject of ongoing work is finalization of a rule editor aimed at cognitive science experiments with rules. This will allow to transition from questionnaire-based data elicitation setup to a more realistic setting.

EasyMiner [6] is an interactive rule learning system for discovering association rules. EasyMiner uses the Classification-based on Associations algorithm (CBA) [1] to generate rule-based classification models from tabular data. A comparison of CBA with follow-up association rule classification algorithms shows that CBA provides very good balance between speed of learning, predictive performance, and understandability of the resulting models. Possibly the biggest limitation of CBA is that it tends to generate larger models in terms of rule count than some related algorithms. We presented the Quantitative CBA algorithm (QCBA) [3], which makes CBA models smaller by recovering some of the information lost during discretization of numeric attributes.

RDF Rules [7] is a framework for extracting horn clauses from RDF-style knowledge bases that is based on the the AMIE+ [2] algorithm. Experiments presented in [2] show that AMIE+, an association rule learning approach, can be orders of magnitude faster than ALEPH, which is based on principles of Inductive Logic Programming (ILP). The main enhancements in RDF Rules compared to AMIE+ include support for preprocessing of numerical data, top-k approach and a new pattern language for limiting the search space. These new features make the framework more practical to use and even faster on some types of tasks. The RDF Rules reference implementation also contains a web-based user interface.

The second part of the presentation was devoted to on-going experimental research on explainability of rule models. We briefly summarized two working papers. The first paper [5] reviews possible effects of about twenty cognitive biases on interpretation of rule-based machine learning models. The second paper [4] reports on several user studies aimed at assessing effect of selected psychological phenomena on plausibility of learnt rules. There was also a brief demo of currently developed functionality of EasyMiner, which is a rule editor with features specifically designed for crowdsourced experiments on explainability.

A discussion followed on the possibilities for adapting presented association rule learning frameworks for current challenges facing ILP.

References

1 Ma, Bing Liu Wynne Hsu Yiming, Bing Liu, and Yiming Hsu. “Integrating classification and association rule mining.” Proceedings of the fourth international conference on knowledge discovery and data mining. 1998.

2 Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.M.: Fast rule mining in ontological knowledge bases with AMIE+. The VLDB Journal **24**(6), 707–730 (2015)

3 Kliegr, Tomáš. "Quantitative CBA: Small and Comprehensible Association Rule Classification Models." arXiv preprint arXiv:1711.10166 (2017).

4 Fürnkranz, Johannes, Tomáš Kliegr, and Heiko Paulheim. "On Cognitive Preferences and the Plausibility of Rule-based Models." arXiv preprint arXiv:1803.01316 (2018).

5 Kliegr, Tomáš, Štěpán Bahník, and Johannes Fürnkranz. "A review of possible effects of cognitive biases on interpretation of rule-based machine learning models." arXiv preprint arXiv:1804.02969 (2018).

6 Vojir, Stanislav, Zeman, Vaclav, Kuchar, Jaroslav, Kliegr, Tomáš: Easyminer.eu: Web framework for interpretable machine learning based on rules and frequent itemsets. Knowledge-Based Systems **150**, 111–115 (2018)

7 Zeman, Václav, Tomáš Kliegr, and Vojtech Svátek.: RdfRules Preview: Towards an Analytics Engine for Rule Mining in RDF Knowledge Graphs. No. 478. CEUR-WS, 2018.

4.16 Inductive Learning of Answer Set Programs

Mark Law (Imperial College London, GB)

Joint work of Mark Law, Alessandra Russo, Krysia Broda
Main reference Mark Law: "Inductive learning of answer set programs", 2018.
URL https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.762179

In recent years, non-monotonic Inductive Logic Programming (ILP) has received growing interest. Specifically, several new learning frameworks and algorithms have been introduced for learning under the answer set semantics, allowing the learning of common-sense knowledge involving defaults and exceptions, which are essential aspects of human reasoning.

The first part of this talk will present our recent advances which have extended the theory of ILP and yielded a new collection of algorithms, called ILASP (Inductive Learning of Answer Set Programs), which are able to learn ASP programs consisting of normal rules, choice rules and both hard and weak constraints. Learning such programs allows ILASP to be applied in settings which had previously been outside the scope of ILP. In particular, weak constraints represent preference orderings, and so learning weak constraints allows ILASP to be used for preference learning.

In the second part, we will present a recent noise-tolerant version of ILASP, which has been successful at learning both from synthetic and from real data sets. In particular, we have shown that on many of the data sets ILASP achieves a higher accuracy than other ILP systems that have previously been applied to those same data sets.

4.17 Representing and Learning Grammars in Answer Set Programming

Mark Law (Imperial College London, GB)

Joint work of Mark Law, Alessandra Russo, Elisa Bertino, Krysia Broda, Jorge Lobo
Main reference Mark Law, Alessandra Russo, Elisa Bertino, Krysia Broda, Jorge Lobo: "Representing and Learning Grammars in Answer Set Programming", in Proc. of the The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 – February 1, 2019., pp. 2919–2928, AAAI Press, 2019.
URL https://aaai.org/ojs/index.php/AAAI/article/view/4147

In this paper, we introduce an extension of context-free grammars called answer set grammars (ASGs). These grammars allow annotations on production rules, written in the language of Answer Set Programming (ASP), which can express context-sensitive constraints. We investigate the complexity of various classes of ASG with respect to two decision problems: deciding whether a given string belongs to the language of an ASG and deciding whether the language of an ASG is non-empty. Specifically, we show that the complexity of these decision problems can be lowered by restricting the subset of the ASP language used in the annotations. To aid the applicability of these grammars to computational problems that require context-sensitive parsers for partially known languages, we propose a learning task for inducing the annotations of an ASG. We characterise the complexity of this task and present an algorithm for solving it. An evaluation of a (prototype) implementation is also discussed.

4.18 Using Association Rule Learning for Verification of Configuration Files

Ruzica Piskac (Yale University – New Haven, US)

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

Traditionally software synthesis is trying to automatically derive code that corresponds to some specification. The specification can be given explicitly, or it is stated in the form of given input-output examples illustrating the intentional code behavior. However, sometimes the main challenge is actually to derive the specification itself.

Using verification for configuration files as our main motivation, we learn specification from a given set of configuration files. This set might also contain faulty configuration files. Software failures resulting from configuration errors have become commonplace as modern software systems grow increasingly large and more complex.

We describe a framework which analyzes data sets of correct configuration files and learns rules for building a language model from the given data set. Our framework is based on a generalized association rule learning.

4.19 Generating Explanations for IP Learned Models

Ute Schmid (Universität Bamberg, DE)

Inductive (logic) programming provides interpretable, human-inspectable, explicitly represented models. However, a symbolic model per se might not be easily comprehensible for humans, especially such humans without a background in programming. To make models comprehensible to domain experts and other end-users, it is therefore necessary to explain model decisions. I will argue that a variety of explanation modes – verbal, visual, and example-based – should be made available to accomodate different personal and situational needs. A current challenge is to combine verbal explanations with visual information. Here we explore different strategies to automatically infer perceptual/spatial features and relations which can be used in the context of symbolic models.

References

1 Schmid, Ute (2018). Inductive Programming as Approach to Comprehensible Machine Learning. Proceedings of the 6th Workshop KI & Kognition (KIK-2018), co-located with 41st German Conference on Artificial Intelligence (KI 2018), Berlin, Germany, September 25, 2018. http://ceur-ws.org/Vol-2194/schmid.pdf

4.20 Semantically Aware Data Wrangling

Gust Verbruggen (KU Leuven, BE) and Luc De Raedt (KU Leuven, BE)

Joint work of Gust Verbruggen, Luc De Raedt, Lidia Contreras-Ochando, Cesar Ferri, José Hernández-Orallo
Main reference Gust Verbruggen, Luc De Raedt: "Automatically Wrangling Spreadsheets into Machine Learning Data Formats", in Proc. of the Advances in Intelligent Data Analysis XVII – 17th International Symposium, IDA 2018, 's-Hertogenbosch, The Netherlands, October 24-26, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 11191, pp. 367–379, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-030-01768-2_30

We show how to use predictive synthesis for automatically wrangling spreadsheets into machine learning formats, introduce semantic spreadsheet segmentation as a way to provide feedback during wrangling and explore how such a predictive synthesizer can be trained from scratch.

5 Discussion groups

5.1 Neuro-symbolic integration

Richard Evans (Google DeepMind – London, GB), Eli Bingham (Uber AI Labs – San Francisco, US), Eneldo Loza Mencía (TU Darmstadt, DE), Harald Ruess (fortiss GmbH – München, DE), Johannes Rabold (Universität Bamberg, DE), Kevin Ellis (MIT – Cambridge, US), Ute Schmid (Universität Bamberg, DE)

We discussed two different approaches to neuro-symbolic integration. In the first, we try to extract interpretable features from standard neural networks (MLPs, CNNs, RNNs). In the second, we implement a hybrid neuro-symbolic architecture that was designed in advance to allow extraction of interpretable features.

When considering the first approach, standard neural networks tend to suffer from what McCarthy called the "propositional fixation": insofar as they learn rules at all, they tend to be rules of propositional logic, not first-order logic. This means the rules do not generalize well. E.g. you train an RNN to reverse a list of length 5, and test it on lists of length 7, and it fails. This means that if we want to extract logical information from a standard neural network, it should be propositional logic, not first-order logic. So we should look at extracting propositional rules, decision-trees, or state-machines – not full first-order rules.

Related work on this approach includes: LIME, LRP (Layerwise Relevance Propagation), www.heatmapping.org, prototype-based neural network layers: https://arxiv.org/abs/1812.01214, neural stethoscopes: https://arxiv.org/abs/1806.05502, interpretable CNNs: https://arxiv.org/abs/1901.02413, and extracting decision trees from NNs: https://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf.

In the second approach, the key question is: what form should the hybrid architecture take? One homogenous system for both low-level features and high-level rules, or two distinct systems that communicate? We considered various options. (1) End-to-end differentiable. Create a differentiable ILP system and learn the high-level rules and the low-level classifiers jointly using SGD. Example: https://arxiv.org/abs/1711.04574. (2) End-to-end SMT. Model the neural classifiers in SMT and learn the high-level rules and low-level classifiers jointly using SMT. Example: https://homes.cs.washington.edu/~bornholt/post/nnsmt.html. (3) End-to-end SAT solver. Binarize the neural net classifier and learn the high-level rules and low-level classifiers jointly using SAT. Example of using SAT for binarized networks: https://arxiv.org/pdf/1710.03107.pdf. (4) Two distinct systems: we have a neural net for classifiers and a separate symbolic ILP system for learning rules. We hope the classifications learned by the net are useful for the ILP system.

Related work for the second approach includes:

- DeepProbLog: https://arxiv.org/pdf/1805.10872.pdf
- Learning Explanatory Rules from Noisy Data: https://arxiv.org/abs/1711.04574
- The Neural Theorem Prover: https://arxiv.org/pdf/1705.11040.pdf
- Neural-symbolic integration: https://arxiv.org/abs/1711.03902
- NeSy: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=74066&copyownerid=108715
- Neural Logic Machines: https://openreview.net/forum?id=B1xY-hRctX

- Combining LIME with ILP for relational explanations: https://link.springer.com/chapter/10.10072F978-3-319-99960-9_7
- Investigating human priors for general game playing: https://arxiv.org/abs/1802.10217

5.2 Language primitive bias engineering for generality

José Hernández-Orallo (Technical University of Valencia, ES), Eli Bingham (Uber AI Labs – San Francisco, US), Lidia Contreras-Ochando (Technical University of Valencia, ES), Andrew Cropper (University of Oxford, GB), Kevin Ellis (MIT – Cambridge, US), Tomáš Kliegr (University of Economics – Prague, CZ), Michael Siebers (Universität Bamberg, DE), and Gust Verbruggen (KU Leuven, BE)

In this working group we explored different ways in which a knowledge base or library of predicates or functions (we will use the term primitive) can be improved so that we get results for a general range of problems, by the introduction or invention of new ones, their abstraction or by the selection (or forgetting) of those that are least useful. Note that this is different from the common situation in which we choose the appropriate domain knowledge for a particular example, or the most appropriate schemas or meta-rules for one or more problems. Here we want to explore the way in which the primitives that can be used to reduce the depth of the search space in a learning problem, while keeping breadth at a reasonable level. All this is related to some ideas in psychology or AI (empowerment, curiosity, intrinsic motivation, meta-learning, self-play, etc.) and compositionality (and of course learning to learn).

We started with a discussion on some novel ways in which the use of self-generation or playing with tasks and examples can make these libraries improve. Note that by sampling tasks and "playing" with them we can improve the bias, even if we are not given domain knowledge. Basically we can create new more abstract primitives that we can reuse and the system is better learning new (and especially more complex) problems. This is possible because there are many redundant programs and a few components appear in many programs, especially those that compress the examples, and also because search cost has breadth as the base and depth in the exponent.

For the approach of generating a sample and use learning as a way of improving the library, we addressed a few issues during the discussion:

- What we sample. There are two main alternatives: to sample on the data (problem space) or by sampling programs (solution space) and using them to get the data. If we sample the data, we can use a uniform distribution on limited-size inputs and outputs (e.g., Andrew's paper [1]), or some other distribution for more complex cases (interactive settings). If we sample the programs, we can use a distribution that is inspired by the domain (e.g., a stochastic grammar, Kevin's paper [2]), or we can use a different distribution, but still the system can learn to learn (or we can use a language for generating programs that is different from the language we are using for learning them). In both cases if we adjust the bias to be more efficient for the generated data, we have a system that gets better incrementally. But would this converge?
- How can we do this sampling better than random? This is motivated because even playing has a cost, and we don't want to play (or mind play) with millions of examples.

The distribution may generate very similar cases again and again: this is the same as in sampling theory where we want to cover the space with very few instances. One idea is a diversity of tasks. How can diversity be measured?

- What are good examples for trying this? There are potential problem domains suitable for playing, and building new Lego designs seemed especially promising. Rebrickable (https://rebrickable.com/) is a web site with lots of existing Lego build instructions. A similar (simpler) setting is Build Battle in Malmo, where different configurations of bricks have to be reproduced by the agent (see, e.g., http://microsoft.github.io/malmo/blog/BuildBattle/Introduction/)

Then we explored the actual ways in which we can improve the (use of) the background knowledge (or the library)? We discussed whether forgetting (dropping least useful primitives or predicates) could be useful. Where in humans and other previous studies (e.g., gErl, [3]) it works. In other systems (e.g., Metagol, [1]) it doesn't help much. There may be some reasons for this, such as noise, or perhaps the fact that some of these systems are simply robust to overfitting. Also, we remember that breadth is on the base and not the exponent of the search cost.

So perhaps this is why ranking or selecting primitives by relevance is more useful and general than forgetting ([2], [4]). One interesting question in this approach is whether we need two "modules": one for solving the learning problem and another one for calculating the relevance? Could we think of an integrated system instead? Or something behind, such as causal model of the world that could exclude many primitives or knowledge rules?

In any case, the choice of relevance for this process is key. Do we agree on the metrics for the second module, i.e., for relevance? There are dependencies (e.g., primitive A is useless if you introduce primitive B) so a single score (as the one used for the rankings) is a simple option, but some other structures could be more powerful (a relevance graph with dependencies). Apart from the relevance metrics, we want metrics to analyse if the knowledge we extract gets more coherent or abstract, or more explainable.

As an action from this group we suggested to explore metrics for relevance (e.g., a primitive is relevant if removing it makes search deeper, a primitive is relevant if adding it makes average depth shorter) and generality (e.g., a primitive is generally useful = aggregation of relevance for a wide range of problems).

So solving this optimisation problem depends on finding the size (and composition) of this set of primitives in terms of the time (or effort, measured in bits of the solution) for solving a range of problems (see Figure 2).

Acknowledgments. This research was supported by the EU (FEDER) and the Spanish MINECO (RTI2018-094403-B-C32), Universitat Politècnica de València (PAID-06-18), and the Generalitat Valenciana (PROMETEO/2019/098 and BEST/2018/027). L. Contreras-Ochando was also supported by the Spanish MECD (FPU15/03219). J. Hernández-Orallo is also funded by FLI (RFP2-152).

References

1 Cropper, Andrew. "Playgol: learning programs through play." arXiv preprint arXiv:1904.08993 (2019).

2 Ellis, Kevin, et al. "Dreamcoder: Bootstrapping domain-specific languages for neurally-guided bayesian program learning." Proceedings of the 2nd Workshop on Neural Abstract Machines and Program Induction (2018).

3 Martínez-Plumed, Fernando, et al. "Knowledge acquisition with forgetting: an incremental and developmental setting." Adaptive Behavior 23.5 (2015): 283-299.

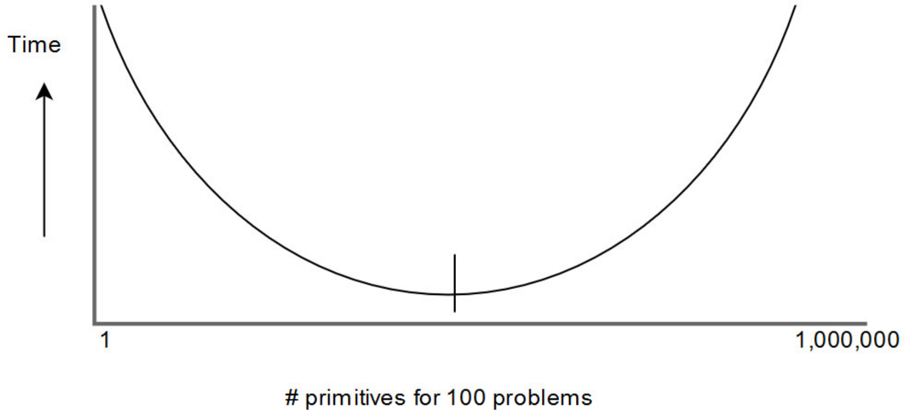

Figure 2 The plot shows that the optimal curve has a minimum, but we may only discover/approximate a curve above (through sampling or assuming a strong prior) that may have a different minimum.

4 Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Martínez-Plumed, F., Ramírez-Quintana, M. J., Katayama, S.: Automated data transformation with inductive programming and dynamic background knowledge. In: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2019 (to appear). ECML-PKDD '19 (2019).

5.3 Game Strategy Discussion Group Report

Stephen H. Muggleton (Imperial College London, GB), Ivan Bratko (University of Ljubljana, SI), Johannes Fürnkranz (TU Darmstadt, DE), Céline Hocquette (Imperial College London, GB), Mark Law (Imperial College London, GB), and Stassa Patsantzis (Imperial College London, GB)

Main reference Stephen H. Muggleton, Céline Hocquette: "Machine Discovery of Comprehensible Strategies for Simple Games Using Meta-interpretive Learning", New Generation Comput., Vol. 37(2), pp. 203–217, 2019.
URL http://dx.doi.org/10.1007/s00354-019-00054-2

We are interested into learning strategies for a variety of games, that can be played with one or several players. Chess endgames are rich and challenging enough which allow to demonstrate convincing results. Another advantage is that minimax databases can be computed and provide a basis for learning, and a baseline for comparison. Examples of such endgames are KRK or KPRKR. Games are equivalent to fully quantified boolean formula.

Determining a strategy is proving the truth of such sentences, for instance with SAT-solving approaches.Incomplete information games such as card games can be solved following an extension of the minimax algorithm. A minimax game tree can be build for each possible world. The strategy is then to minimize the expectation of the regret over several worlds. Another categories of games interesting for the AI community is video games, for which a standing example is the Atari games.

Participants

- Eli Bingham
Uber AI Labs –
San Francisco, US
- Ivan Bratko
University of Ljubljana, SI
- Maurice Chandoo
Leibniz Universität
Hannover, DE
- Lidia Contreras-Ochando
Technical University of
Valencia, ES
- Andrew Cropper
University of Oxford, GB
- Luc De Raedt
KU Leuven, BE
- Amit Dhurandhar
IBM TJ Watson Research Center
– Yorktown Heights, US
- Kevin Ellis
MIT – Cambridge, US
- Richard Evans
Google DeepMind – London, GB
- Cesar Ferri Ramirez
Technical University of
Valencia, ES
- Johannes Fürnkranz
TU Darmstadt, DE
- Elena Leah Glassman
Harvard University –
Cambridge, US
- José Hernández-Orallo
Technical University of
Valencia, ES
- Céline Hocquette
Imperial College London, GB
- Tomáš Kliegr
University of Economics –
Prague, CZ
- Mark Law
Imperial College London, GB
- Eneldo Loza Mencía
TU Darmstadt, DE
- Stephen H. Muggleton
Imperial College London, GB
- Stassa Patsantzis
Imperial College London, GB
- Ruzica Piskac
Yale University – New Haven, US
- Johannes Rabold
Universität Bamberg, DE
- Harald Ruess
fortiss GmbH – München, DE
- Ute Schmid
Universität Bamberg, DE
- Michael Siebers
Universität Bamberg, DE
- Armando Solar-Lezama
MIT – Cambridge, US
- Gust Verbruggen
KU Leuven, BE

Report from Dagstuhl Seminar 19211

Enumeration in Data Management

Edited by

Endre Boros[1], Benny Kimelfeld[2], Reinhard Pichler[3], and Nicole Schweikardt[4]

1 **Rutgers University – Piscataway, US,** endre.boros@rutgers.edu
2 **Technion – Haifa, IL,** bennyk@cs.technion.ac.il
3 **TU Wien, AT,** pichler@dbai.tuwien.ac.at
4 **HU Berlin, DE,** schweikn@informatik.hu-berlin.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19211 "Enumeration in Data Management". The goal of the seminar was to bring together researchers from various fields of computer science, including the Databases, Computational Logic, and Algorithms communities, and establish the means of collaboration towards considerable progress on the topic. Specifically, we aimed at understanding the recent developments, identifying the important open problems, and initiating collaborative efforts towards solutions thereof. In addition, we aimed to build and disseminate a toolkit for data-centric enumeration problems, including algorithmic techniques, proof techniques, and important indicator problems. Towards the objectives, the seminar included tutorials on the topic, invited talks, presentations of open problems, working groups on the open problems, discussions on platforms to compile the community knowledge, and the construction of various skeletons of such compilations.

Seminar May 19–24, 2019 – http://www.dagstuhl.de/19211

2012 ACM Subject Classification Mathematics of computing → Combinatorial algorithms, Mathematics of computing → Enumeration, Theory of computation → Complexity classes, Theory of computation → Data structures and algorithms for data management, Theory of computation → Database query processing and optimization (theory)

Keywords and phrases constant delay, databases, dynamic complexity, enumeration, polynomial delay, query evaluation

Digital Object Identifier 10.4230/DagRep.9.5.89

Edited in cooperation with Nofar Carmeli

1 Executive Summary

Endre Boros
Benny Kimelfeld
Reinhard Pichler
Nicole Schweikardt

In recent years, various concepts of enumeration have arisen in the fields of Databases, Computational Logic, and Algorithms, motivated by applications of data analysis and query evaluation. Common to all concepts is the desire to compute a stream of items with as small as possible waiting time between consecutive items, referred to as the "delay." Alongside each concept, there evolved algorithmic techniques for developing solvers, and proof techniques for establishing complexity bounds. In addition to the traditional guarantees of "polynomial delay" and "incremental polynomial," researchers have been pursuing stronger guarantees such as "constant delay" in the context of logical query evaluation, "dynamic complexity" of incremental maintenance, and "factorized databases." The growing interest and rapid evolution of the associated research brings up opportunities of significantly accelerating the computation of big results, by devising and adopting general-purpose methodologies.

In Dagstuhl Seminar 19211 on "Enumeration in Data Management," key researchers from relevant communities have gathered to gain a better understanding the recent developments, lay out the important open problems, and join forces towards solutions thereof. These communities include researchers who explore enumeration problems in the fields of *databases*, *logic*, *algorithms* and *computational complexity*. We have had invited tutorials by

- Luc Segoufin on *Constant-delay enumeration*
- Takeaki Uno on *Enumeration algorithms*
- Yann Strozecki on *Enumeration complexity – defining tractability*
- Markus Kröll on *Enumeration complexity – a complexity theory for hard enumeration problems*
- Endre Boros on *Monotone generation problems.*

We also had presentations by most of the other participants. Moreover, the participants have prepared in advance a list of open problems in a document that we shared and jointly maintained. We have discussed the open problems during designated times of the seminar.

The organizers are highly satisfied with the seminar. We have got a very high acceptance rate for our invitations. In fact, there were further researchers whom we would have liked to invite after the first invitation round but, unfortunately, no room was left. The participants were exceptionally involved and engaged. Some considerable progress has been made on the open problems prepared in advance, as will be reported in future publications that will acknowledge the seminar. The seminar has also initiated joint efforts to disseminate toolkits for data-centric enumeration problems, including algorithmic techniques, proof techniques, and important indicator problems. To this end, we have had sessions of working groups for the different types of toolkit components. In particular, we have initiated a Wikipedia page on enumeration algorithms:

https://en.wikipedia.org/wiki/Enumeration_algorithm

This page will evolve to contain a thorough picture of the principles and techniques of enumeration problems.

2 Table of Contents

Executive Summary

Overview of Talks

Open problems

3 Overview of Talks

3.1 Hardness for Polynomial Time Problems

Amir Abboud (IBM Almaden Center – San Jose, US)

This will be an overview of recent results in Fine-Grained Complexity. A small set of conjectures about the exact time complexity of certain core problems (such as SAT and 3SUM) are used to derive strong conditional lower bounds for many many other problems. The focus of the talk will be on the conjectures and techniques that are (or may be) of interest to the enumeration algorithms community. In particular, we will highlight the k-Clique conjectures.

3.2 A Circuit-Based Approach to Efficient Enumeration: Enumerating MSO Query Results on Trees and Words

Antoine Amarilli (Telecom ParisTech, FR)

Joint work of Antoine Amarilli, Pierre Bourhis, Louis Jahiet, Stefan Mengel, Matthias Niewerth

This talk presents our circuit-based approach to enumerate the results of monadic second-order (MSO) queries on trees. Specifically, we explain how a deterministic tree automaton encoding an MSO query can be translated to a d-DNNF set circuit representing its answers on an input tree. These answers can then be enumerated with linear-time preprocessing and constant-delay using our algorithm in [1]. The talk also explains how our methods can be extended to work with nondeterministic automata and ensure combined tractability, i.e., ensure that the preprocessing and delay are polynomial in the automaton, following our upcoming results in [4]. We last explain how these methods apply to the problem of enumerating tractably the matches of regular expressions (possibly with captures) on a text document, using our algorithm from [3], and show a demo of a preliminary implementation for this task.

References

1 Antoine Amarilli, Pierre Bourhis, Louis Jachiet, Stefan Mengel. A Circuit-Based Approach to Efficient Enumeration. ICALP 2017. https://arxiv.org/abs/1702.05589

2 Antoine Amarilli, Pierre Bourhis, Stefan Mengel. Enumeration on Trees under Relabelings. ICDT 2018. https://arxiv.org/abs/1709.06185

3 Antoine Amarilli, Pierre Bourhis, Stefan Mengel, Matthias Niewerth. Constant-Delay Enumeration for Nondeterministic Document Spanners. ICDT 2019. https://arxiv.org/abs/1807.09320

4 Antoine Amarilli, Pierre Bourhis, Stefan Mengel, Matthias Niewerth. Enumeration on Trees with Tractable Combined Complexity and Efficient Updates. PODS 2019. https://arxiv.org/abs/1812.09519

3.3 Constant delay enumeration of CQs with FPT-preprocessing

Christoph Berkholz (HU Berlin, DE)

Joint work of Christoph Berkholz, Nicole Schweikardt

In this talk I discuss for which classes of conjunctive queries (CQs) it is possible to enumerate the query result with constant delay after FPT-preprocessing (where the CQ is the parameter). While a general classification theorem is still missing, I present a dichotomy for classes of self-join-free and quantifier-free CQs.

The talk is based on an ongoing joint work with Nicole Schweikardt.

3.4 Monotone Generation Problems

Endre Boros (Rutgers University – Piscataway, US)

This tutorial talk provides an overview of monotone generation problems, the core problem of generating hypergraph transversals, and related techniques and results.

The first part introduces general techniques and results on hypergraph (monotone Boolean) dualization, the crucial task of finding a next output element, and known results about its complexity and hardness in special cases.

The second part surveys some of the standard techniques (that are actualloy useful even for non-monotone generation problems). These techniques include the Flashlight Principle (promising polynomial delay generation), the Supergraph Approach (promising incremental polynomial generation or faster), the Projection Method (that generally guarantees total polynomial time generation) and the method of Joint Generation (that provides a framework for incrementally quasi- polynomial generation). Each of these techniques can be made to work of course only if certain conditions are fulfilled.

The third part illustrates some of these techniques on problems related to/derived from the problem of generating vertices of polyhedra, leading to a special case that turns out to be NP-hard.

The talk closes with a section on open problems formulated in the very general setting of generating simplices and bodies for a given point set in a euclidian space.

3.5 Enumeration Complexity of UCQs

Nofar Carmeli (Technion – Haifa, IL) and Markus Kröll (TU Wien, AT)

Main reference Nofar Carmeli, Markus Kröll: "On the Enumeration Complexity of Unions of Conjunctive Queries", in Proc. of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2019, Amsterdam, The Netherlands, June 30 – July 5, 2019., pp. 134–148, ACM, 2019.
URL https://doi.org/10.1145/3294052.3319700

We study the enumeration complexity of Unions of Conjunctive Queries (UCQs). We aim to identify the UCQs that are tractable in the sense that the answer tuples can be enumerated with linear preprocessing time and constant delay. A union of tractable CQs is always tractable. We show that some non-redundant unions containing intractable CQs are tractable. Interestingly, some unions consisting of only intractable CQs are tractable too. The question of finding a full characterization of the tractability of UCQs remains open. We end the talk with open problems, and describe some examples of specific queries for which we have no classification.

3.6 An Optimization-based Primer on Flag Algebras

Aritanan Gruber (University of ABC – Santo André, BR)

Joint work of Aritanan Gruber, Marcel K. de Carli Silva, Fernando Mário de Oliveira Filho, Cristiane Maria Sato

Despite all the progress made in the past two decades, a unifying theory of enumeration algorithms seems, by any reasonable matter, a distance milestone – if achievable at all. In the mean time, we search for tools and inspiration in the slightly more developed settings of counting and density estimation. The theory of flag algebras offers a systematic approach to derive computer-assisted proofs of density estimation results in asymptotic extremal combinatorics. We briefly survey the theory in an optimization-based approach and then focus on a conic programming strong duality relation lying at its core, providing simpler proofs along the way. At the end, we mention possible extensions and ramifications of our approach.

3.7 Parameterized Enumeration

Heribert Vollmer (Leibniz Universität Hannover, DE)

Joint work of Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, Heribert Vollmer
Main reference Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, Heribert Vollmer: "Paradigms for Parameterized Enumeration", Theory Comput. Syst., Vol. 60(4), pp. 737–758, 2017.
URL https://doi.org/10.1007/s00224-016-9702-4

We introduce parameterized classes for enumeration problems. Particular interest is devoted to the class DelayFPT. We prove some structural results about this class, in particular we obtain a characterization in terms of kernelization. We also give a number of upper and lower bounds for specific problems from graph theory and propositional logic.

References

1 Nadia Creignou, Arne Meier, Julian-Steffen Müller, Johannes Schmidt, and Heribert Vollmer. Paradigms for parameterized enumeration. *Theory Comput. Syst.*, 60(4):737–758, 2017.

3.8 Some Generalizations of the Monotone Boolean Duality Testing (Hypergraph Transversal) Problem

Khaled Elbassioni (Khalifa University – Abu Dhabi, AE)

Joint work of Khaled Elbassioni, Leonid Khachiyan, Endre Boros, Vladimir Gurvich, Kazuhisa Makino

Given two sets A and B of binary vectors such that no vector in A is dominated by a vector in B, the well-known monotone Boolean duality testing problem calls for checking if A and B monotonically cover the entire binary cube, that is, if every vertex of the cube dominates some vector in A or is dominated by some vector in B. In this talk, we consider two generalizations of monotone Boolean duality testing problem. In the first generalization, we are given a hypergraph where each hyperedge intersects all but a small number of other hyperedges, and a list of colors for each vertex, and the requirement is to check if each vertex can be assigned a color from its list such that no hyperedge is monochromatic. In the second generalization, the binary cube and families A and B are replaced by a box and vectors over an integer lattice, and the requirement again is to check if A and B monotonically cover all the lattice points inside the box. We illustrate that the first generalization can be solved in quasi-polynomial time and give an application of the second generalization to finding sparse regions in multi-dimensional data.

References

1 Khaled M. Elbassioni. Quasi-polynomial algorithms for list-coloring of nearly intersecting hypergraphs. *CoRR*, abs/1904.02425, 2019.

2 Leonid Khachiyan, Endre Boros, Khaled M. Elbassioni, Vladimir Gurvich, and Kazuhisa Makino. Dual-bounded generating problems: Efficient and inefficient points for discrete probability distributions and sparse boxes for multidimensional data. *Theor. Comput. Sci.*, 379(3):361–376, 2007.

3.9 Listing Maximal Subgraphs Satisfying Strongly Accessible Properties

Andrea Marino (Univerisità degli Studi di Firenze, IT)

Joint work of Alessio Conte, Roberto Grossi, Andrea Marino, Luca Versari
Main reference Alessio Conte, Roberto Grossi, Andrea Marino, Luca Versari: "Listing Maximal Subgraphs Satisfying Strongly Accessible Properties", SIAM J. Discrete Math., Vol. 33(2), pp. 587–613, 2019.
URL https://doi.org/10.1137/17M1152206

Algorithms for listing the subgraphs satisfying a given property (e.g., being a clique, a cut, a cycle, etc.) fall within the general framework of set systems. A set system $(\mathcal{U}, \mathcal{F})$ consists of a ground set $\mathcal{U}$ (e.g., a network's nodes) and a family $\mathcal{F} \subseteq 2^{\mathcal{U}}$ of subsets of $\mathcal{U}$ that have the required property. For the problem of listing all sets in $\mathcal{F}$ maximal under inclusion, the ambitious goal is to cover a large class of set systems, preserving at the same time the efficiency of the enumeration. Among the existing algorithms, the best-known ones list the maximal subsets in time proportional to their number but may require exponential space. In this talk, we show how to improve the state of the art in two directions by introducing an algorithmic framework based on reverse search that, under standard suitable conditions, simultaneously (i) extends the class of problems that can be solved efficiently to *strongly accessible* set systems, and (ii) reduces the additional space usage from exponential in $|\mathcal{U}|$ to *stateless*, i.e., with no additional memory usage other than that proportional to the solution size, thus accounting for just polynomial space.

References

1 Alessio Conte, Roberto Grossi, Andrea Marino, Luca Versari. Listing Maximal Subgraphs Satisfying Strongly Accessible Properties. SIAM Journal Discrete Mathematics. 2019

3.10 Dichotomies for Evaluation and Enumeration Problems for Simple Regular Path Queries

Wim Martens (Universität Bayreuth, DE)

Joint work of Wim Martens, Tina Trautner
Main reference Wim Martens, Tina Trautner: "Evaluation and Enumeration Problems for Regular Path Queries", in Proc. of the 21st International Conference on Database Theory, ICDT 2018, March 26-29, 2018, Vienna, Austria, LIPIcs, Vol. 98, pp. 19:1–19:21, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018.
URL http://dx.doi.org/10.4230/LIPIcs.ICDT.2018.19

Regular path queries (RPQs) are a central component of graph databases. We investigate decision and enumeration problems concerning the evaluation of RPQs under several semantics that have recently been considered: arbitrary paths, shortest paths, paths without node repetitions (simple paths), and paths without edge repetitions (trails). Whereas arbitrary and shortest paths can be dealt with efficiently, simple paths and trails become computationally difficult already for very small RPQs. We study RPQ evaluation for simple paths and trails from a parameterized complexity perspective and define a class of simple transitive expressions that is prominent in practice and for which we can prove dichotomies for the evaluation problem. We observe that, even though simple path and trail semantics are intractable for RPQs in general, they are feasible for the vast majority of RPQs that are used in practice. At the heart of this study is a result of independent interest: the two disjoint paths problem in directed graphs is W[1]-hard if parameterized by the length of one of the two paths.

References

1 Wim Martens, Tina Trautner. *Evaluation and Enumeration Problems for Regular Path Queries*. International Conference on Database Theory (ICDT), 2018: 19:1-19:21.

3.11 A Circuit-Based Approach to Efficient Enumeration

Stefan Mengel (CNRS, CRIL – Lens FR)

Joint work of Antoine Amarilli, Pierre Bourhis, Louis Jachiet, Stefan Mengel, Matthias Niewerth
Main reference Antoine Amarilli, Pierre Bourhis, Louis Jachiet, Stefan Mengel: "A Circuit-Based Approach to Efficient Enumeration", in Proc. of the 44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland, LIPIcs, Vol. 80, pp. 111:1–111:15, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.ICALP.2017.111

In this talk, we present a framework for efficient enumeration that is based on representations by circuits. We show that for a certain type of circuits that we call d-DNNF set circuits we can enumerate the sets computed by a circuit with constant delay and linear preprocessing. Combined with the fact that several query answering problems from database theory allow efficient representations by these set circuits, this yields a general technique for constant delay enumeration in that area.

References

1 Amarilli, A., Bourhis, P., Jachiet, L., and Mengel, S. A Circuit-Based Approach to Efficient Enumeration. ICALP 2017.
2 Amarilli, A., Bourhis, P., and Mengel, S. Enumeration on Trees under Relabelings. ICDT 2018.
3 Niewerth, M. MSO Queries on Trees: Enumerating Answers under Updates Using Forest Algebras. LICS 2018.
4 Amarilli, A., Bourhis, P., Mengel, S., and Niewerth, M. Constant-Delay Enumeration for Nondeterministic Document Spanners. ICDT 2019.
5 Amarilli, A., Bourhis, P., Mengel, S., and Niewerth, M. Enumeration on Trees with Tractable Combined Complexity and Efficient Updates. PODS 2019, to appear

3.12 Complex Event Processing and Efficient Enumeration

Cristian Riveros (Pontificia Universidad Catolica de Chile, CL)

Complex Event Processing (CEP) is a unifying field of technologies for processing and correlating distributed data sources in real-time. CEP finds applications in diverse domains, which has resulted in a large number of proposals for expressing and processing complex events. However, although these technologies have reached a good level of maturity, existing CEP systems still lack of general techniques of query evaluation with strong performance guarantees, regarding the update time per event and enumeration of complex events.

In this talk, I will present our recent proposal for processing complex events [1]. I will start by presenting our framework of a query language and automata model for defining and compiling CEP queries with local predicates, a subset of queries that are in the core of CEP. Then I will explain our evaluation strategy for processing this subset of queries with strong performance guarantees, namely, constant update time per event and constant delay enumeration of complex events.

References

1 Alejandro Grez, Cristian Riveros, and Martín Ugarte. A formal framework for complex event processing. In *ICDT*, volume 127 of *LIPIcs*, pages 5:1–5:18. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2019.

3.13 Tutorial on Enumeration Complexity: Defining Tractability

Yann Strozecki (University of Versailles, FR)

Joint work of Yann Strozecki, Florent Capelli, Arnaud Mary
Main reference Florent Capelli, Yann Strozecki: "Incremental delay enumeration: Space and time", Discrete Applied Mathematics, 2018.
URL https://doi.org/10.1016/j.dam.2018.06.038

We review the different ways tractability is defined in enumeration using as complexity measure total time, incremental time, delay and space. We present the associated complexity classes and show how they relate and can be separated modulo classical complexity hypothesis. The focus is on understanding incremental polynomial time and polynomial delay which are the classes where the majority of problems are. In particular, we present an attempt to classify which saturation problems naturally in incremental polynomial time are in fact in polynomial delay. We also briefly present low complexity classes and further desirable properties for practitioners: constant delay and strong polynomial delay and show related results and conjectures on the enumeration of models of a DNF.

References

1 Florent Capelli and Yann Strozecki. Incremental delay enumeration: Space and time. *Discrete Applied Mathematics*, 2018.

3.14 Pattern mining with MDL

Alexandre Termier (IRISA – University of Rennes, FR)

Joint work of Alexandre Termier, Jilles Vreeken, Matthijs van Leeuwen, Peggy Cellier, Esther Galbrun, Clement Gautrais, Bruno Crémilleux
Main reference Jilles Vreeken, Matthijs van Leeuwen, Arno Siebes: "Krimp: mining itemsets that compress", Data Min. Knowl. Discov., Vol. 23(1), pp. 169–214, 2011.
URL https://doi.org/10.1007/s10618-010-0202-x

Pattern mining is the field of data mining concerned with discovering (local) regularities in data. Pattern mining algorithms solve combinatorial problems. As such, they are built around an enumeration algorithm, combined with an efficient way to check that the "pattern property" holds into the data (which can be seen as an oracle for the enumeration algorithm). While this approach has been extremely popular at the beginning of the century, the exponential size of its output has increasingly turned off its potential users. Actual data scientists expect to get a synthetic representation of the main patterns occurring of the data. A promising line of research, led by Vreeken and van Leeuwen, proposed to exploit compression techniques, namely the Minimum Description Length (MDL) principle, to select a small set of patterns that are especially good representative of the data. In this talk, we introduce the main ideas of mining patterns based on the MDL principle, and briefly sketch questions that may be of interest for the enumeration community.

3.15 Enumeration of Maximum Cliques and Its Application to Coding Theory

Etsuji Tomita (The University of Electro-Communications – Tokyo, JP)

Joint work of Takayuki Nozaki, Akira Mitsutake
URL http://www.etlab.lab.uec.ac.jp/t̃omita/index_e.html

We present an algorithm for enumerating all maximum cliques of a graph and its application to error correcting codes. First, we review our depth-first search algorithm CLIQUES for enumerating all maximal cliques of a graph in which pruning methods are employed as in the Bron-Kerbosch algorithm. Subsequently, we show step by step modification and improvements to the previous algorithm in order to obtain an algorithm for finding a maximum clique. We have employed techniques from our maximum-clique-finding algorithms MCQ, MCR, MCS, MCT, and some others. We note that an algorithm for enumerating all maximum cliques can be easily obtained from an algorithm for finding a maximum clique. Finally, we deal with coding theory, especially with single deletion correcting codes. The mathematically defined VT (Varshamov – Teneholtz) code is well known to be a single deletion correcting code, but many important problems remain unsolved. We show that largest single deletion correcting codes can be enumerated through enumeration of maximum cliques. Then we present new findings in non-binary single deletion correcting codes by way of enumeration of maximum cliques.

References

1 Etsuji Tomita. Efficient algorithms for finding maximum and maximal cliques and their applications. In *WALCOM*, volume 10167 of *Lecture Notes in Computer Science*, pages 3–15. Springer, 2017.

2 Akira Mitsutake, Takayuki Nozaki, and Etsuji Tomita. Construction of best non-binary single deletion correcting codes via maximum clique enumeration. In *41st Symposium on Information Theory and Its Applications*, Iwaki, Fukushima, Japan, 2018.

3.16 OBDD and Interpretability in Machine Learning

György Turan (University of Illinois – Chicago, US)

Joint work of György Turan, Karine Chubarian

The comprehensibility of models produced using machine learning methods is an important requirement in many applications. Comprehensibility, or interpretability, can refer either to a human user, or to a system using the model as a component. One approach is to construct an interpretable representation of a learned model. For naive Bayesian network classifiers, Chan and Darwiche proposed ordered binary decision diagrams (OBDD) as an exact interpretable representation. As this representation may be of exponential size, we consider approximate representations. It is shown that for tree-augmented naive Bayes classifiers, approximate OBDD representations of polynomial size can be computed efficiently. The algorithm generalizes an algorithm of Gopalan, Klivans and Meka for approximate counting of knapsack solutions to a class of quadratic polynomial threshold functions.

3.17 Constant-Delay Enumeration applied to Dynamic Query Processing: The Dynamic Yannakakis Algorithm

Stijn Vansummeren (Free University of Brussels, BE)

Joint work of Muhammad Idris, Martín Ugarte, Stijn Vansummeren, Hannes Voigt, Wolfgang Lehner
Main reference Muhammad Idris, Martín Ugarte, Stijn Vansummeren, Hannes Voigt, Wolfgang Lehner: "Conjunctive Queries with Inequalities Under Updates", PVLDB, Vol. 11(7), pp. 733–745, 2018.
URL https://doi.org/10.14778/3192965.3192966

The ability to efficiently analyze changing data is a key requirement of many real-time analytics applications like Stream Processing, Complex Event Recognition, Business Intelligence, and Machine Learning.

Traditional approaches to this problem are based either on the materialization of subresults (to avoid their recomputation) or on the recomputation of subresults (to avoid the space overhead of materialization). Both techniques have recently been shown suboptimal: instead of fully materializing results and subresults, one can maintain a data structure that supports efficient maintenance under updates and can quickly enumerate the full query output, as well as the changes produced under single updates.

In our work we are concerned with designing a practical family of algorithms for dynamic query evaluation based on this idea, for queries featuring both equi-joins and inequality joins, as well as certain forms of aggregation. Our main insight is that, for acyclic conjunctive queries, such algorithms can naturally be obtained by modifying Yannakakis' seminal algorithm for processing acyclic joins in the static setting.

This talk presents the main ideas behind this modification, offsets it against the traditional ways of doing incremental view maintenance, and discusses recent extensions such as dealing with general theta-joins.

References

1 Muhammad Idris, Martín Ugarte, Stijn Vansummeren, Hannes Voigt, and Wolfgang Lehner. Conjunctive queries with inequalities under updates. *PVLDB*, 11(7):733–745, 2018.

2 Muhammad Idris, Martín Ugarte, and Stijn Vansummeren. The dynamic Yannakakis algorithm: Compact and efficient query processing under updates. In *SIGMOD Conference*, pages 1259–1274. ACM, 2017.

3.18 Dynamic Complexity of Reachability

Thomas Zeume (TU Dortmund, DE)

Joint work of Thomas Zeume, Samir Datta, Anish Mukherjee, Raghav Kulkarni, Thomas Schwentick, Nils Vortmeier
Main reference Samir Datta, Raghav Kulkarni, Anish Mukherjee, Thomas Schwentick, Thomas Zeume: "Reachability Is in DynFO", J. ACM, Vol. 65(5), pp. 33:1–33:24, 2018.
URL https://doi.org/10.1145/3212685

Dynamic descriptive complexity theory studies how query results can be updated in a highly parallel fashion, that is, by constant-depth circuits or, equivalently, by first-order formulas, or by the relational algebra. After gently introducing dynamic complexity theory, I will discuss recent results regarding the dynamic complexity of the reachability query.

4 Open problems

4.1 Conjunctive Queries with FPT Preprocessing

Christoph Berkholz (HU Berlin, DE)

Bagan et al. [1] showed (under some algorithmic assumptions) that the result $q(D)$ of a self-join-free CQ q over a database D of size N can be enumerated with constant delay after $O(f(q) \cdot N)$ preprocessing time if and only if q is free-connex acyclic. Here, a self-join-free CQ is a query of the form $\exists \overline{x}_0 \bigwedge_{i=1}^{k} R_i(\overline{x}_i)$ for pairwise distinct relation symbols R_i. An open problem is to characterise those CQs that allow constant delay after FPT preprocessing, which means $O(f(q) \cdot N^c)$ preprocessing time for some fixed c. More precisely, let Φ be a recursively enumerable class of self-join-free CQs and consider the following problem:

Problem:	*CQ-Enum(Φ)*
Parameters:	A fixed recursively enumerable class Φ of self-join-free CQs.
Input:	A CQ $q \in \Phi$ and a database D.
Goal:	After some preprocessing, enumerate all tuples of $q(D)$ with constant delay.

The open problem is: for which (recursively enumerable classes of self-join-free CQs) Φ exist a constant c and a computable function f such that CQ-Enum($\mathcal{C}$) can be solved with $O(f(q) \cdot N^c)$ preprocessing? Let us call such classes *tractable*.

It is known that a class Φ is tractable, if it has bounded free-connex treewidth [1], bounded free-connex fractional hypertree-width (i.e. in [2]), or bounded free-connex submodular width [Berkholz, Schweikardt; unpublished]. For join queries (i.e. quantifier free CQs) it then follows from Marx' lower bound [3], that, assuming the exponential time hypothesis (ETH), a class Φ is tractable if and only if Φ has bounded submodular width. However, a full classification for classes of CQs with projections is missing. An intriguing example is the *k-star query with a quantified center*, i.e., the query $\psi_k = \exists z \bigwedge_{i=1}^{k} R_i(z, x_i)$. It is open whether the class $\Psi^{\text{star}} = \{\psi_k : k \in \mathbb{N}\}$ is tractable, that is, whether CQ-Enum(Ψ^{star}) can be solved with $O(f(k) \cdot N^c)$ preprocessing and constant delay over databases of size N.

References

1 Guillaume Bagan, Arnaud Durand, and Etienne Grandjean. On acyclic conjunctive queries and constant delay enumeration. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, pages 208–222, 2007.

2 Dan Olteanu and Jakub Závodný. Size bounds for factorised representations of query results. *ACM Trans. Database Syst.*, 40(1):2:1–2:44, 2015.

3 Dániel Marx. Tractable hypergraph properties for constraint satisfaction and conjunctive queries. *J. ACM*, 60(6):42:1–42:51, 2013.

4.2 Complexity of Enumeration versus Complexity of Counting

Etienne Grandjean (Caen University, FR)

It seems that in general, counting the solutions of a problem is at least as difficult as (and often much harder than) enumerating the solutions of the problem.

E.g., the solutions of problems *2-SAT* and *Perfect-Matching* are enumerable with delay $O(n)$ in the size n of each output solution. However, the corresponding counting problems are $\sharp$P-complete.

Question

Is this "assertion" that we have the order Enumeration $\leq$ Counting (for complexity) a general phenomenon? If yes, to what extent and with what explanation (maybe informal)?

(I strongly think that this inequality holds for logical/query problems. E.g., it holds for Boolean CSP problems [1, 2] where affine queries are the only ones for which the counting problem is easy.)

Opposite challenging question

Can one exhibit a "natural" problem (in graph theory, combinatorics, etc.) which would be a counterexample to the previous assertion, that is a problem for which it is easy to count the solutions but hard (or harder) to enumerate them?

References

1 Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Inf. Comput.*, 125(1):1–12, 1996.

2 Nadia Creignou and Jean-Jacques Hébrard. On generating all solutions of generalized satisfiability problems. *ITA*, 31(6):499–511, 1997.

4.3 Maximal Ptime Graph Properties

Mamadou Moustapha Kanté (Université Clermont Auvergne – Aubiere, FR)

Consider the following question:

Problem:	*Maximal Ptime Graph Property*
Parameters:	A Ptime property
Input:	A graph G
Goal:	Enumerate all maximal (induced) subgraphs satisfying the property

If the property is defined with a list of finite induced subgraph obstructions, one can reduce the problem to the enumeration of minimal transversals in hypergraphs of bounded dimension, resulting in an incremental enumeration algorithm. Some natural questions:

1. For properties with finite induced obstructions, can we have a polynomial delay enumeration algorithm?

2. For which Ptime properties the enumeration is incremental? Preferred examples are minor-closed classes of graphs or generally graphs with a short description of obstructions. Short can be: a finite list of (infinite) patterns (like chordal graphs) or finite list wrt a quasi-order (like minor relation).

4.4 Enumerating Minimal Triangulations in Polynomial Delay

Batya Kenig (University of Washington – Seattle, US)

A graph is triangulated, or chordal, if every cycle of four or more vertices has a chord. Edges can be added to any given graph so that the resulting graph, called a triangulation of the input graph, is chordal. Carmeli et al. [1] have shown that the minimal triangulations of a graph (with regard to inclusion) can be enumerated in incremental polynomial delay. It remains open whether the minimal triangulations of a graph can be enumerated in polynomial delay.

Some observations that point to a negative answer to this question are the following. Deciding whether there exists a minimal triangulation that excludes some set of edges is NP-complete by reduction from the chordal graph sandwich problem of Golumbic et al. [2]. This rules out the possibility of a polynomial delay algorithm by naive backtrack search. It can also be shown that using the framework of Carmeli et al. [1] (i.e., enumerating the maximal independent sets for *succinct graph representations*) cannot lead to a polynomial delay algorithm assuming the Strong Exponential Time Hypothesis (SETH).

References

1 Nofar Carmeli, Batya Kenig, and Benny Kimelfeld. Efficiently enumerating minimal triangulations. In *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2017, Chicago, IL, USA, May 14-19, 2017*, pages 273–287, 2017.

2 Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. Graph sandwich problems. *J. Algorithms*, 19(3):449–473, 1995.

4.5 Evaluating Automata with Capture Variables

Benny Kimelfeld (Technion – Haifa, IL)

A *variable-set automaton* (vset-automaton for short) is an automaton that can open and close variables along its run on a string, so that each variable can be opened and later closed at most once. (So, a vset-automaton is a special case of a transducer.) Therefore, a successful run defines an assignment of spans (intervals from the input string) to variables. We refer to such an assignment as a *query answer.* These automata have been studied by Fagin et al. [1] in the framework of document spanners for information extraction, and they required that a run assigns a span to *every* variable. Maturana et al. [2] have studied the variant where every variable is assigned *at most* once, hence we get *partial* query answers.

Problem:	*Maximal Answers for Vset-Automata*
Parameters:	
Input:	A string **s**, a vset-automaton A.
Goal:	Enumerate the *maximal* answers of $A(\mathbf{s})$.

The query answers can be enumerated with polynomial delay under both the complete and incomplete semantics [3, 4, 2]. The above problem is about the *maximal* (rather than *all partial*) answers can also be enumerated with polynomial delay.

References

1 Ronald Fagin, Benny Kimelfeld, Frederick Reiss, and Stijn Vansummeren. Document spanners: A formal approach to information extraction. *J. ACM*, 62(2):12:1–12:51, 2015.

2 Francisco Maturana, Cristian Riveros, and Domagoj Vrgoc. Document spanners for extracting incomplete information: Expressiveness and complexity. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 125–136, 2018.

3 Antoine Amarilli, Pierre Bourhis, Stefan Mengel, and Matthias Niewerth. Constant-delay enumeration for nondeterministic document spanners. In *22nd International Conference on Database Theory, ICDT 2019, March 26-28, 2019, Lisbon, Portugal*, pages 22:1–22:19, 2019.

4 Dominik D. Freydenberger, Benny Kimelfeld, and Liat Peterfreund. Joining extractions of regular expressions. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 137–149, 2018.

4.6 Ordered Query Evaluation with Sublinear Delay

Benny Kimelfeld (Technion – Haifa, IL)

Problem:	*Ordered Query Evaluation*
Parameters:	A query $q(x_1, \dots, x_k)$, a partial order $\succeq$ over the k-tuples.
Input:	A database D.
Goal:	Enumerate the tuples of $q(D)$ sorted by $\succeq$.

We have a good understanding of the complexity of the problem without order guaranteed, at least for Conjunctive Queries (CQs) [1] and unions of CQs [2]. Specifically, the following is known for CQs without self-joins (i.e., where each relation symbol occurs once), under conventional complexity assumptions (within polynomial time). The answers can be enumerated with a constant delay following a linear pre-processing if and only if the CQ is *acyclic and free-connex* [1]. Moreover, the algorithm of Bagan et al. [1] enumerates in an ordered fashion if the linear order $\succeq$ is the lexicographic order under an *elimination order* of the variables, meaning (intuitively) that there exists a join tree such that every prefix of the variable ordering corresponds to a rooted subtree. Nevertheless, not much is known about other orders. Moreover, the proof technique for the lower bound does not seem to be suitable to dismissing any ordering (assuming the CQ is acyclic free-connex). In particular, concrete questions include the following:

- Which variable orderings allow for a lexicographic answer ordering in sublinear delay (following a linear pre-processing)?
- Can we also support non-lexicographic orders such as decreasing *sum* of numeric values?
- Are there techniques from fine-grained complexity theory that allow to dismiss the possibility of some (natural) orders?

References

1 Guillaume Bagan, Arnaud Durand, and Etienne Grandjean. On acyclic conjunctive queries and constant delay enumeration. In *Computer Science Logic, 21st International Workshop, CSL 2007, 16th Annual Conference of the EACSL, Lausanne, Switzerland, September 11-15, 2007, Proceedings*, pages 208–222, 2007.

2 Nofar Carmeli and Markus Kröll. Enumeration complexity of unions of conjunctive queries. *CoRR*, abs/1812.03831, 2018.

4.7 Clause Sequences of a SAT Formula

Markus Kröll (TU Wien, AT)

The following problem arises when enumerating the answers to well-designed pattern trees, see [1]. In this work, Kroell et al. show that enumeration is possible in polynomial delay for several classes of well-designed pattern trees, while enumeration for other classes is intractable. One class of pattern trees however is still unclassified (see [1], Table 1 on page 16). The enumeration problem below can be reduced to this unsolved case, thus any lower bound on the enumeration complexity is especially helpful.

Let $k \geq 3$ be an integer, and let ϕ be a k-SAT formula with clauses $C_1, \ldots, C_m$ and variables $x_1, \ldots, x_n$. A truth assignment $a : \mathsf{var}(\phi) \to \{0,1\}$ of ϕ leads to a *clause sequence* $c(a) = (c_1, \ldots, c_m) \in \{0,1\}^m$ as follows: Every clause satisfied by a leads to a 1, every clause not satisfied by a to a 0, i.e., $c_i = 1$ if the assignment a restricted to clause i equals 1, otherwise $c_i = 0$. (This means that ϕ is satisfyable iff there exists some assignment a' with $c(a') = (1, \ldots, 1)$. Moreover, the problem MAX-SAT can be encoded by asking for the clause sequence with the largest sum of elements).

▶ **Example 1.** Consider the SAT instance $\phi = x_1 \wedge \neg x_2 \wedge (x_1 \vee x_2)$, thus $C_1 = \{x_1\}$, $C_2 = \{\neg x_2\}$ and $C_3 = \{x_1, x_2\}$. The assignment $a_1 = \{x_1 \mapsto 1, x_2 \mapsto 1\}$ leads to the clause sequence $c(a_1) = (1,0,1)$, the truth assignment $a_2 = \{x_1 \mapsto 0, x_2 \mapsto 1\}$ to the clause sequence $a_2 = (0,0,1)$. The set of clause sequences is given by

$$\{(0,1,0),(1,1,1),(0,1,1),(1,0,1)\}$$

The enumeration problem is now given as follows (k can be assumed to be fixed):

Problem:	*Enumerating all Clause Sequences*
Parameters:	k
Input:	A k-Sat formula ϕ.
Goal:	All clause sequences.

References

1 Markus Kröll, Reinhard Pichler, and Sebastian Skritek. On the Complexity of Enumerating the Answers to Well-designed Pattern Trees. In Wim Martens and Thomas Zeume, editors, *19th International Conference on Database Theory (ICDT 2016)*, volume 48 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:18, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

4.8 Unions of Conjunctive Queries

Markus Kröll (TU Wien, AT) and Nofar Carmeli (Technion – Haifa, IL)

Carmeli and Kroell [1] have initiated a systematic study on the enumeration complexity of unions of conjunctive queries (UCQs). The enumeration problem is given as follows:

Problem:	*UCQ Enumeration*
Parameters:	A UCQ $Q = \bigcup Q_i$
Input:	A database D.
Goal:	Enumerate all tuples of $Q(D)$.

As with enumerating the answers to CQs [2], the size of the query is assumed to be fixed. In this setting, the task is to fully characterize which class of UCQs allows for a constant delay enumeration with linear preprocessing. An example of a UCQ, for which the complexity is still unknown, is the following:

▶ **Example 1.** Let $Q = Q_1 \cup Q_2$ with

$$Q_1(x, z, y, v) \leftarrow R_1(x, z, v), R_2(z, y, v), R_3(y, x, v) \text{ and}$$
$$Q_2(x, z, y, v) \leftarrow R_1(x, z, v), R_2(y, t_1, v), R_3(t_2, x, v).$$

Note that Q_1 is cyclic, while Q_2 is acyclic free-connex. A result introduced in [1] shows how an easy query in a union can be used to enumerate the answers to a hard query (Theorem 12). However, for this query, it seems that this approach cannot be used. As of now, it is still an open problem to show either an upper or a lower bound for the enumeration complexity of the query Q.

References

1 Nofar Carmeli, Markus Kröll. *On the Enumeration Complexity of Unions of Conjunctive Queries*. PODS 2019: 134-148.

2 Guillaume Bagan, Arnaud Durand, Etienne Grandjean: On Acyclic Conjunctive Queries and Constant Delay Enumeration. CSL 2007: 208-222

4.9 Maximal Subgraphs with a Forbidden Pattern

Reinhard Pichler (TU Wien, AT)

The following defines a whole family of enumeration problems.

Problem:	*Maximal Subgraphs with Forbidden Pattern*
Parameters:	A graph $G' = (V', E')$, referred to as "pattern."
Input:	A (directed or undirected) graph $G = (V, E)$.
Goal:	Enumerate all maximal, induced subgraphs of G not containing pattern G'.

We say that G contains pattern G' if G contains a subgraph that is isomorphic to G'. This problem generalizes, for instance, the enumeration problem of Maximal Independent Sets, where the forbidden pattern is the graph consisting of a single edge. More generally, also a variant of the enumeration problem of Maximal Independent Sets of k-uniform hypergraphs can be cast as a special case of the Maximal Subgraphs with Forbidden Pattern problem: in case of the Maximal Independent Sets problem of k-uniform hypergraphs, we are given a hypergraph $H = (V, E)$, where each edge in E consists of precisely k vertices from V. Enumerating the Maximal Independent Sets of H corresponds to the Maximal Subgraphs with Forbidden Pattern problem where G is the primal graph of H and G' is a k-clique.

In case of Maximal Independent Sets of graphs (resp. of k-uniform hypergraphs), we know that the enumeration problem is solvable with polynomial delay (resp. in incremental polynomial time) [1, 2]. It would be interesting to come up with sufficient conditions on the graph G and or the pattern G' to make the enumeration problem solvable with polynomial delay or in incremental polynomial time. In which cases does the enumeration problem become intractable? Can one show some kind of dichotomy or trichotomy?

For instance, we have recently come across the following problem: G' is the path of length 2 and G is a tri-partite graph with vertex sets V_1, V_2, and V_3, such that $E \subseteq (V_1 \times V_2) \cup (V_2 \times V_3)$ and each vertex in V_1 and in V_3 is incident to exactly one edge. It is easy to show that this problem can be solved in incremental polynomial time (in fact, it can be cast as a Maximal Independent Sets problem of 3-uniform hypergraphs by representing each path of length 2 in G as a hyperedge in H with these 3 vertices). Can all maximal subgraphs of G not containing G' be enumerated with polynomial delay?

References

1 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. On generating all maximal independent sets. *Inf. Process. Lett.*, 27(3):119–123, 1988.

2 Thomas Eiter and Georg Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Comput.*, 24(6):1278–1304, 1995.

Participants

- Amir Abboud
  IBM Almaden Center –
  San Jose, US
- Kira V. Adaricheva
  Hofstra University –
  Hempstead, US
- Antoine Amarilli
  Telecom ParisTech, FR
- Kristof Berczi
  Eötvös Lorand University –
  Budapest, HU
- Christoph Berkholz
  HU Berlin, DE
- Endre Boros
  Rutgers University –
  Piscataway, US
- Pierre Bourhis
  CNRS – CRIStAL, Lille, FR
- Nofar Carmeli
  Technion – Haifa, IL
- Ondrej Cepek
  Charles University – Prague, CZ
- Nadia Creignou
  Aix-Marseille University, FR
- Arnaud Durand
  University Paris-Diderot, FR
- Khaled M. Elbassioni
  Khalifa University –
  Abu Dhabi, AE
- Etienne Grandjean
  Caen University, FR
- Alejandro J. Grez
  Pontificia Universidad Catolica
  de Chile, CL
- Aritanan Gruber
  University of ABC –
  Santo André, BR
- Mamadou Moustapha Kanté
  Université Clermont Auvergne –
  Aubiere, FR
- Batya Kenig
  University of Washington –
  Seattle, US
- Benny Kimelfeld
  Technion – Haifa, IL
- Christoph Koch
  EPFL – Lausanne, CH
- Phokion G. Kolaitis
  University of California –
  Santa Cruz, US
- Markus Kröll
  TU Wien, AT
- Ester Livshits
  Technion – Haifa, IL
- Kazuhisa Makino
  Kyoto University, JP
- Andrea Marino
  Univerisità degli Studi di
  Firenze, IT
- Wim Martens
  Universität Bayreuth, DE
- Stefan Mengel
  CNRS, CRIL – Lens FR
- Shin-Ichi Nakano
  Guuma University – Kiryu, JP
- Matthias Niewerth
  Universität Bayreuth, DE
- Lhouari Nourine
  University Clermont
  Auvergne, FR
- Liat Peterfreund
  Technion – Haifa, IL
- Reinhard Pichler
  TU Wien, AT
- Cristian Riveros
  Pontificia Universidad Catolica
  de Chile, CL
- Yehoshua Sagiv
  The Hebrew University of
  Jerusalem, IL
- Nicole Schweikardt
  HU Berlin, DE
- Thomas Schwentick
  TU Dortmund, DE
- Luc Segoufin
  INRIA & ENS Paris, FR
- Yann Strozecki
  University of Versailles, FR
- Alexandre Termier
  IRISA – University of
  Rennes, FR
- Etsuji Tomita
  The University of
  Electro-Communications –
  Tokyo, JP
- György Turan
  University of Illinois –
  Chicago, US
- Martin Ugarte
  PUC Chile, CL
- Takeaki Uno
  National Institute of Informatics –
  Tokyo, JP
- Stijn Vansummeren
  Free University of Brussels, BE
- Alexandre Vigny
  University of Warsaw, PL
- Heribert Vollmer
  Leibniz Universität
  Hannover, DE
- Thomas Zeume
  TU Dortmund, DE

Report from Dagstuhl Seminar 19212

Topology, Computation and Data Analysis

Edited by

Michael Kerber[1], Vijay Natarajan[2], and Bei Wang[3]

1 **TU Graz, AT,** kerber@tugraz.at
2 **Indian Institute of Science – Bangalore, IN,** vijayn@iisc.ac.in
3 **University of Utah – Salt Lake City, US,** beiwang@sci.utah.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19212 "Topology, Computation and Data Analysis". The seminar brought together researchers with mathematical and computational backgrounds in addressing emerging directions within computational topology for data analysis in practice. This seminar was designed to be a followup event after a very successful Dagstuhl Seminar (17292; July 2017). The list of topics and participants were updated to keep the discussions diverse, refreshing, and engaging. This seminar facilitated close interactions among the attendees with the aim of accelerating the convergence between mathematical and computational thinking in the development of theories and scalable algorithms for data analysis.

Seminar May 19–24, 2019 – http://www.dagstuhl.de/19212
2012 ACM Subject Classification Human-centered computing → Visualization, Information systems → Data analytics, Mathematics of computing → Algebraic topology, Theory of computation → Computational geometryy
Keywords and phrases computational topology, topological data analysis, Topology based visualization
Digital Object Identifier 10.4230/DagRep.9.5.110

1 Executive Summary

Vijay Natarajan (Indian Institute of Science – Bangalore, IN)
Michael Kerber (TU Graz, AT)
Bei Wang (University of Utah – Salt Lake City, US)

The Dagstuhl Seminar titled "Topology, Computation, and Data Analysis" brought together researchers in mathematics, computer science, and visualization to engage in active discussions on theoretical, computational, practical, and application aspects of topology for data analysis. The seminar has led to stronger ties between the computational topology and TopoInVis (topology based visualization) communities and identification of research challenges and open problems that can be addressed together.

Context

Topology is the study of connectivity of space that abstracts away geometry and provides succinct representations of the space and functions defined on it. Topology-based methods for data analysis have received considerable attention in the recent years given its promise

Topology, Computation and Data Analysis, *Dagstuhl Reports*, Vol. 9, Issue 5, pp. 110–131
Editors: Michael Kerber, Vijay Natarajan, and Bei Wang
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to handle large and feature-rich data that are becoming increasingly common. Computing topological properties in the data domain and/or range is a step in the direction of more abstract, higher-level data analysis and visualization. Such an approach has become more important in the context of automatic and semi-automatic data exploration, analysis, and understanding. The primary attraction for topology-based methods is the ability to generate "summary" qualitative views of large data sets. Such views often require fewer geometrical primitives to be extracted, stored, and to be visualized as compared to views obtained directly from the raw data. Two communities, computational topology and TopoInVis (topology based visualization), have made significant progress during the past two decades on developing topological abstractions and applying them to data analysis. In addition, there are multiple other research programs (relatively fewer in number) on this topic within the statistics and machine learning fields, and within a few application domains. Computational topology grew from within computational geometry and algebraic topology and studies algorithmic questions on topological structures. The focus of topological data analysis and TopoInVis is data – algorithms, methods, and systems for improved and intuitive understanding of data via application of topological structures. Researchers in computational topology typically have a math or theoretical computer science background whereas TopoInVis researchers have a computational, computer engineering, or applied background. There is very little communication between the two communities due to the different origins and the fact that there are no common conferences or symposia where both communities participate.

Goals

The Dagstuhl seminar 17292 (July 2017) successfully brought together researchers with mixed background to talk about problems of mutual interest. Following this seminar, the benefits of the inter-community ties was well appreciated, at least by the attendees of the seminar. The goal of the current seminar was to strengthen existing ties, establish new ones, identify challenges that requires the two communities to work together, and establish mechanisms for increased communication and transfer of results from one to the other. During the previous Dagstuhl seminar, we also noticed significant interaction between researchers within the individual communities, with say theoretical and applied backgrounds. We wanted to continue to encourage such interaction.

Topics

We chose four current and emerging topics that will benefit from an inter-community discussion. Topics are common to both communities, with different aspects studied within an individual community.

Reeb graphs, Reeb Spaces, and Mappers. The Reeb graph, its loop-free version called the contour tree, and the higher-dimensional generalization called the Reeb space are topological structures that capture the connectivity of level sets of univariate or multivariate functions. They are independently well studied within the computational topology and TopoInVis communities. Recent developments define stable distance measures between Reeb graphs, inspired by analogous distance measures in persistent homology. Barring a few exceptions, the theoretical results have no practical realizations. On the practical side, effective visual exploration and visual analysis methods based on Reeb graphs and spaces have been developed for a wide variety of domains including combustion studies, climate science, astronomy, and molecular modeling. These applications often utilize only a simplified version of the topological structure. One such simplification, the mapper algorithm, consists of a discretized version of Reeb graphs and has shown an immense industrial potential. Very recently, the theoretical aspects of the mapper algorithm and its generalizations has moved in the focus of research. Exchange of ideas and results between the two communities will help advancing this progress further.

Topological analysis and visualization of multivariate data. Multivariate datasets arise in many scientific applications. Consider, for example, combustion or climate simulations where multiple physical measurements (say, temperature and pressure) or concentrations of chemical species are computed simultaneously. We model these variables mathematically as multiple continuous, real-valued functions. We are interested in understanding the relationships between these functions, and more generally, in developing efficient and effective tools for their analysis and visualization. Unlike for real-valued functions, very few tools exist for studying multivariate data topologically. Besides the aforementioned Reeb spaces and mappers, notable examples of these tools are the Jacobi sets, Pareto sets, and Joint Contour Nets. Understanding the theoretical properties of these tools and adapting them in analysis and visualization remains a very active research area. In addition, combining these topological tools with multivariate statistical analysis would be of interest. On the other hand, research towards multidimensional persistence would help advance multivariate data analysis both mathematically and computationally. We plan to expand our discussion on multidimensional persistent homology that include topics such as identifying meaningful and computable topological invariants; discussing computability and applicability in the multidimensional setting, comparison of multidimensional data, kernel methods for multidimensional persistence, and adapting multidimensional persistence in visualization.

New opportunities for vector field topology. Vector field topology for visualization pioneered by Helman and Hesselink has inspired much research in topological analysis and visualization of vector fields. A large body of work for time-independent vector field deals with fixed (critical) points, invariant sets, separatrices, periodic orbits, saddle connectors and Morse decomposition as well as vector field simplification that reduces its complexity. Research for time-dependent vector field is concerned with critical point tracking, Finite Time Lyapunov Exponents (FTLE), Lagrangian coherent structure (LCS), streak line topology, as well as unsteady vector field topology. For this workshop, we ask the following questions: can advancements in computational topology help bring new opportunities for the study of vector field topology? In particular, can they help developing novel, scalable and mathematically rigorous ways to rethink vector field data? An example is the topological notion of robustness, a cousin of persistence, introduced via the well diagram and well group theory. Robustness has been shown to be very useful in quantifying feature stability for steady and unsteady vector fields.

Software tools and libraries. How do we make topological data analysis applicable to large datasets? A natural first step is algorithm and software engineering. This refers to developing the best algorithms for a particular problem and to optimize the implementation of these algorithms. The state of affairs within the communities is quite diverse: while scalable algorithms are available for some problems(e.g., computation of Reeb graphs or persistence diagrams in low dimensions), current developments make significant progress on other fronts, for example the computation of approximate persistence diagrams of Vietoris-Rips complexes. On the other extreme, the theory of multi-dimensional persistence is just beginning to be supported by algorithmic contributions. Besides these efforts, parallelizable and distributed algorithms play an important role towards practicality. One further important aspect of software design is interface design, that is, to make those implementations available to non-experts. While this final development step is usually rather neglected in theoretical research, there have been efforts in both communities towards generally applicable and easy-to-use software. Software contributors of both communities will profit from exchanging ideas and experiences.

Participants, Schedule, and Organization

The invitees were identified according to the focus topics of the seminar while ensuring diversity in terms of gender, country / region of workplace, and experience. The aim was to bring together sufficient number of experts interested in each topic and representing the two communities to facilitate an engaging discussion.

We planned for different talk types, longer overviews and shorted contributed research talks, and breakout sessions. We scheduled six overview talks on the first day. These overview talks were aligned with the four topics of the seminar, planned to be accessible to members of both communities, and set the stage for the discussions and shorter research talks on the following days. The speakers Ulrich Bauer (Reeb graphs), Christoph Garth (topology based methods in visualization), Gunther Weber (topological analysis for exascale), Michael Lesnick (computational aspects of 2-parameter persistence) Claudia Landi (multi-parameter persistence), and Vanessa Robins (discrete Morse theory and image analysis) gave a gentle introduction to the area followed by a state-of-the-art report and discussion on open problems.

Participants gave short research talks (16 total) during Tuesday-Friday with a focus on challenges and opportunities. These talks were organized during the morning sessions.

We scheduled breakout sessions on the afternoons of Tuesday and Thursday. On Tuesday, we solicited discussion topics and identified three topics to be of interest – *multivariate data, reconstruction,* and *tensor field topology.* Participants chose to join a group based on their interest. All groups contained participants from both communities. We formed two discussion groups on Thursday. The first group wanted to further discuss *multivariate data* with inputs from experts on multi-parameter persistence who were part of a different group on Tuesday. The second breakout session was on *Multi-parameter persistence computation*, where they discussed and analyzed a recently proposed algorithm. All groups presented a summary of their discussion and plans during a plenary session at the end of the day.

Many participants joined an organized excursion to Bernkastel-Kues on Wednesday afternoon. On Friday morning, we scheduled a discussion and brainstorming session to close the seminar and and to plan for future events.

Results and Reflection

Participants unanimously agreed that the seminar was successful in enabling cross-fertilization and identifying important challenging problems that require both communities to work together. The breakout sessions were instrumental in identifying some of the challenges and topics for further collaboration. At least two such challenges (together with motivating applications) were identified, possibly leading to collaboratory efforts.

The breakout sessions were planned for the entire afternoon after lunch. The longer duration allowed for in-depth and technical discussions that stimulates further work after the seminar. Based on feedback during informal discussions and the brainstorming session on Friday, we expect multiple working groups will be formed to write expository articles and survey articles. Members of the two communities have also shown enthusiasm to participate in workshops and conferences of each other. In conclusion, we believe that the seminar has achieved the goal of bringing together the two communities and charting a path for tackling bigger challenges in the area of topological data analysis.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

3 Overview of Talks

3.1 The Space of Reeb Graphs

Ulrich Bauer (TU München, DE)

Main reference Ulrich Bauer, Claudia Landi, Facundo Mémoli: "The Reeb Graph Edit Distance is Universal", CoRR, Vol. abs/1801.01866, 2018.
URL https://arxiv.org/abs/1801.01866

Reeb graphs are low-dimensional descriptors of continuous real-valued functions, capturing information of the connectivity of level sets. We will discuss different definitions of distances between Reeb graphs that are stable with respect to perturbations of the function. We will motivate the notion of a universal distance, and show that most previously considered distances are not universal. We also show that a universal distance can be constructed as a graph edit distance.

3.2 Toward Objective Finite-Time Flow Topology

Roxana Bujack (Los Alamos National Laboratory, US)

We explore how to extend the definition of classical vector field saddles, sinks, and sources to finite time settings in an objective way, i.e. invariant with respect to Euclidean transformations of the reference frame.

3.3 Software Patterns in Parallel Computational Topology

Hamish Carr (University of Leeds, GB)

Computational topology is now established as a class of data analysis techniques, especially for scientific visualization. The scale of the data, however, combined with the global nature of the analysis, leads to the need for effective parallel techniques, and this has formed a significant element of recent research in the area. Enough experience has now accumulated that we can see patterns emerging in the algorithms at scale, and this talk is a first attempt to tease out some of these patterns for future algorithmic development.

3.4 Generalized Persistence Algorithm for Multi-Parameter Persistence

Tamal K. Dey (Ohio State University – Columbus, US)

Main reference Tamal K. Dey, Cheng Xin: "Generalized Persistence Algorithm for Decomposing Multi-parameter Persistence Modules", CoRR, Vol. abs/1904.03766, 2019.
URL https://arxiv.org/abs/1904.03766

There is no known generalization of the classical persistence algorithm to the multi-parameter setting. In this talk, we present one such generalization that provides a matrix based reduction technique for computing the decomposition of a persistence module given by a multi-parameter simplicial filtration. The time complexity of the algorithm is better than the popular Meataxe algorithm used for the purpose.

3.5 Mapper Algorithm and its Variations

Pawel Dlotko (Swansea University, GB)

Main reference Pawel Dlotko: "Ball mapper: a shape summary for topological data analysis", CoRR, Vol abs/1901.07410, 2019.
URL https://arxiv.org/abs/1901.07410

The classical mapper algorithm by Gunnar Carlson, Facundo Memoli and Gurjeet Singh have brought a major breakthrough to topological data analysis. Sometimes however it is difficult to set up all the parameters, especially the lens function. In this talk I will present a ball mapper – a simple construction that allow to some extend to recover shapes of point clouds and continuous spaces.

3.6 Topology-Based Methods in Visualization – A Very High-Level Overview

Christoph Garth (TU Kaiserslautern, DE)

This talk presents a very high-level overview of the state of the art in the area of topology-based visualization. A typology for topological models used in visualization is used to organize research results and the state of the art. I discuss relations among topological models and for each model describe research results for the computation, simplification, visualization, and application. The talk identifies themes common to subfields, current frontiers, and unexplored territory in this research area.

3.7 Topcat: Computing Multiparameter Persistence – Local and Global Methods

Oliver Gäfvert (KTH Royal Institute of Technology – Stockholm, SE)

Topcat is a software for computing multiparameter persistence modules and their invariants. I will talk about the various algorithms to compute the multiparameter persistence module, starting from the original by Carlsson, Singh and Zomorodian and the subsequent ones by Lesnick, Wright and Skryzalin and the one implemented in Topcat. As shown by Skryzalin, current algorithms to compute the persistence module are exponential in the number of filtration parameters. Can this be improved? Finally, I will give an overview of the invariants implemented in Topcat and also give a demo of the software.

3.8 Topology in Visualization – Mix and Match for Applications

Ingrid Hotz (Linköping University, SE)

The goal of visual data analysis and exploration is to generate an environment for scientific reasoning through interaction with data. The basis for such an effective environment is a multi-scale data abstraction that can serve as a backbone for data navigation. Topological data analysis provides an excellent means for this purpose especially with respect to the rapid development of robust extraction algorithms. Mathematical rigorous guarantees contribute strongly to the acceptance of topological analysis tools. However, despite the increasing success of topological methods every new application still implies new challenges. Not only practical and efficient solutions are required. First of all, a semantic context has to be created. An application specific interpretation and adaptation of topological concepts is needed which can then be embedded in a visual analytics framework. Sometimes this might also mean to give up some of the beauty of the mathematical concepts for approximations and heuristics. In this presentation, these challenges are demonstrated on a few examples from our current research.

3.9 Discrete Morse Theory and Multi-Parameter Persistence

Claudia Landi (University of Modena, IT)

Main reference Claudia Landi, Sara Scaramuccia: "Persistence-perfect discrete gradient vector fields and multi-parameter persistence", CoRR, Vol. abs/1904.05081, 2019.
URL http://arxiv.org/abs/1904.05081

Discrete Morse theory permits reducing a cell complex to the critical cells of a discrete gradient vector field defined on it while maintaining all homological information. In particular, the number of critical cells of the vector field bounds the Betti numbers of the cell complex. A similar reduction procedure based on discrete Morse theory can be used for persistent homology, with any number of parameters. So it is natural to ask what information about persistence we can get form the number of critical cells in this case. In this talk, we see how to derive inequalities involving the number of critical cells of a vector field consistent with a multi-filtration and the Betti tables of its persistence module.

3.10 Computational Aspects of 2-Parameter Persistence

Michael Lesnick (University at Albany, US)

Joint work of Michael Lesnick, Matthew Wright

I will introduce 2-parameter persistent homology, focusing on computational aspects, and in particular the problem of minimal presentation computation.

3.11 Local-global Merge Tree Computation with Local Exchanges

Arnur Nigmetov (TU Graz, AT)

Joint work of Arnur Nigmetov, Dmitriy Morozov

Local-global merge trees were invented for distributed computation of merge trees on a very large array of data. In some situations, in particular, in cosmology, one is interested in a merge tree of a function above a user-given threshold. With this cut-off, it is not necessary for all processors to exchange data; only local exchanges are sufficient. We use triplet merge tree representation and compare two algorithms for computation of a merge tree on functions that were computed with Advanced Mesh Refinement numerical solver.

3.12 Topology and Information Fusion

Emilie Purvine (Pacific Northwest National Lab. – Seattle, US)

In the era of "big data" we are often overloaded with information from a variety of sources. Information fusion is important when different data sources provide information about the same phenomena. In order to discover a consistent world view, or a set of competing world views, we must understand how to aggregate or fuse information from these different sources. In practice much of information fusion is done on an ad hoc basis, when given two or more specific data sources to fuse. It turns out that the mathematics of sheaf theory provides a canonical and provably necessary language and methodology for the general problem of information fusion. In this talk I will motivate the introduction of sheaf theory through the lens of information fusion examples including hypothesis discovery, document clustering, and topic modeling.

3.13 Topologically Accurate Digital Image Analysis using Discrete Morse Theory

Vanessa Robins (Australian National University – Canberra, AU)

Main reference Olaf Delgado-Friedrichs, Vanessa Robins, Adrian P. Sheppard: "Skeletonization and Partitioning of Digital Images Using Discrete Morse Theory", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 37(3), pp. 654–666, 2015.
URL https://doi.org/10.1109/TPAMI.2014.2346172

Algorithms to summarise structure in digital images are fundamental to computer vision, image understanding and quantitative analysis. Traditional methods include the watershed transform for region labelling (partitioning) of grayscale images and topology-preserving thinning of binary images to a medial axis (skeletonisation). My work with x-ray micro-CT images of granular and porous materials required the development of robust topologically accurate and consistent versions of these operations. This talk will describe how discrete Morse theory and persistent homology provide the foundations for simplified skeletonisation and partitioning of grayscale images. I will also discuss some of the open challenges that remain in their application to quantitative analysis of complicated geometries.

3.14 MOG: Mapper on Graphs

Paul Rosen (University of South Florida – Tampa, US)

Main reference Mustafa Hajij, Bei Wang, Paul Rosen: "MOG: Mapper on Graphs for Relationship Preserving Clustering", CoRR, Vol. abs/1804.11242, 2018.
URL http://arxiv.org/abs/1804.11242

The interconnected nature of graphs often results in difficult to interpret clutter. Typically techniques focus on either decluttering by clustering nodes with similar properties or grouping together edges with similar relationship. We propose using mapper, a powerful topological data analysis tool, to summarize the structure of a graph in a way that both clusters data with similar properties and preserves relationships. Typically, mapper operates on a given data by utilizing a scalar function defined on every point in the data and a cover of the scalar function codomain. The output of mapper is a graph that summarizes the shape of the space. In this talk, I outline how to use this mapper construction on an input graph, outline three filter functions that capture important structures of the input graph, and discuss an interface for interactively modifying the cover.

3.15 Topology in Features – Features in Topology

Filip Sadlo (Universität Heidelberg, DE)

Vector field topology in visualization is mainly concerned with concepts related to critical points, periodic orbits, and their invariant manifolds. We will discuss that many of these structures are, however, closely related to features that are not considered topological. While such features can provide additional context in steady vector fields, they enable corresponding definitions in cases where the original concepts are not applicable, such as in time-dependent transport. It is an aim of this talk to trigger discussions to what extent such considerations are applicable in computational topology and beyond.

3.16 Continuous Optimization and Persistence Diagrams

Primoz Skraba (Queen Mary University of London, GB)

I presented recent work showing several applications of gradient descent over topological information. I will give the definition of computing the gradient using an idea inspired by max-pooling from deep networks using Morse theory. Then I will describe three applications: enforcing continuity in the functional map setting, surface reconstruction with prescribed topology, and imposing topological structure on random data.

3.17 An Overview of the Topology ToolKit

Julien Tierny (CNRS-Sorbonne University – Paris, FR)

This talk presents an overview of the Topology ToolKit (TTK), an open-source library for Topological Data Analysis (TDA). TTK implements, in a generic and efficient way, a substantial collection of reference algorithms in TDA. Since its initial public release in 2017, both its user and developer bases have grown, resulting in an increase in the number of supported features. The purpose of this talk is to provide an overview of the features currently supported by TTK, ranging from image segmentation tools to advanced topological analysis of point cloud data, with concrete usage examples available on the TTK website.

3.18 Measuring Point-Clouds with Information-Theoretic Non-Distances

Hubert Wagner (IST Austria – Klosterneuburg, AT)

I will show how computational topology can be used in conjunction with information theory for analyzing certain kinds of high dimensional data. In particular, I will focus on ways to define a dissimilarity measure for data represented by collections of histograms based on intersections of non-metric balls. I will also discuss simpler to compute approximations.

3.19 A Structural Average of Merge Trees and Uncertainty Visualization

Bei Wang (University of Utah – Salt Lake City, US)

Physical phenomena in science and engineering are frequently modeled using scalar fields. In scalar field topology, graph-based topological descriptors such as merge trees, contour trees, and Reeb graphs are commonly used to characterize topological changes in the (sub)level sets of scalar fields. One of the biggest challenges and opportunities to advance topology-based visualization is to understand and incorporate uncertainty into such topological descriptors to effectively reason about their underlying data.

In this work, we study a structural average of a set of labeled merge trees and use it to encode uncertainty in data. Specifically, we compute a 1-center tree that minimizes its maximum distance to any other tree in the set under a well-defined metric called the interleaving distance. We also provide heuristic strategies that compute structural averages of merge trees whose labels do not fully agree. We provide an interactive visualization system that resembles a numerical calculator which takes as input a set of merge trees and outputs a tree as their structural average. We highlight structural similarities between the input and the average and incorporate uncertainty information for visual exploration. We develop a novel measure of uncertainty, referred to as consistency, via a metric-space view of the input trees. Our work is the first to employ interleaving distances and consistency to study a global, mathematically rigorous, structural average of merge trees in the context of uncertainty visualization. This is joint work with Lin Yan, Yusu Wang, Elizabeth Munch, and Ellen Gasparovic.

3.20 Topological Analysis for Exascale Computing: Approaches & Challenges

Gunther H. Weber (Lawrence Berkeley National Laboratory, US)

Simulation has quickly evolved to become the third pillar of science and supercomputing centers provide the computational power needed for accurate simulations. Furthermore, there are concentrated efforts in the Exascale Computing Project to cross the next barrier and build a supercomputer that can run simulations at quintillion calculations per second. This talk provides an overview over how topological data analysis has helped in abstracting and analyzing simulation results. It furthermore outlines the challenges that current developments in supercomputer architecture pose to efficient algorithm design for topological data analysis and presents initial solution approaches.

3.21 Interactive Design and Visualization of Branched Covering Spaces

Eugene Zhang (Oregon State University – Corvallis, US)

Branched covering spaces are a mathematical concept which originates from complex analysis and topology and has applications in tensor field topology and geometry remeshing. Given a manifold surface and an N-way rotational symmetry field, a branched covering space is a manifold surface that has an N-to-1 map to the original surface except at the ramification points, which correspond to the singularities in the rotational symmetry field.

Understanding the notion and mathematical properties of branched covering spaces is important to researchers in tensor field visualization and geometry processing, and their application areas. In this paper, we provide a framework to interactively design and visualize the branched covering space (BCS) of an input mesh surface and a rotational symmetry field defined on it. In our framework, the user can visualize not only the BCSs but also their construction process. In addition, our system allows the user to design the geometric realization of the BCS using mesh deformation techniques as well as connecting tubes. This enables the user to verify important facts about BCSs such as that they are manifold surfaces around singularities, as well as the Riemann-Hurwitz formula which relates the Euler characteristic of the BCS to that of the original mesh. Our system is evaluated by student researchers in scientific visualization and geometry processing as well as faculty members in mathematics at our university who teach topology. We include their evaluations and feedback in the paper.

3.22 Second-order Symmetric Tensor Field Topology: Accomplishments and Challenges

Yue Zhang (Oregon State University – Corvallis, US)

Second-order tensor fields have many applications in medicine, solid mechanics, fluid dynamics, and geometry processing. In this talk, I will review recent advances in 3D symmetric tensor field topology and discuss open research problems on this topic.

4 Working groups

4.1 Reconstruction

Pawel Dlotko (Swansea University, GB), Georges-Pierre Bonneau (INRIA – Grenoble, FR), Roxana Bujack (Los Alamos National Laboratory, US), Michael Kerber (TU Graz, AT), Arnur Nigmetov (TU Graz, AT), Filip Sadlo (Universität Heidelberg, DE), and Julien Tierny (CNRS-Sorbonne University – Paris, FR)

This discussion was motivated by the following practical question coming from the High Performance Computing Community; suppose we are running a large scale simulation in a distributed domain on a supercomputer. Due to the gap between what we can compute and what we can store (often referred to as an Input/Output bottleneck) vast majority of data have to be dismissed when they are computed. Therefore the leading question is, what is the information we can store that is meaningful and useful for the further analysis? Building on a number of successful instances of topological simplification and compression, which preserves some of its topological structure, we raised the question: "How can we reconstruct an approximation of the original data?".

Besides the (1) compression, we collected other applications where a reconstruction of a scalar field from a topological structure are also very helpful: (2) construct a topologically meaningful representative of an ensemble, (3) simplification of the data for an overview, and (4) beautification/abstraction of data for outreach or teaching purposes.

We agreed that conservation of the basic topological structure without adding any additional extrema is the only objective metric that could be used to evaluate the quality of the reconstruction. More concrete ones are not useful because the different applications would demand different metrics. For example, the compression would require a similarity to the original data while the beautification would not require that at all.

As the type of the structure we want to preserve changes from application to application, in this discussion we have brought together a few existing techniques and speculate about other possible techniques that can be used in a number of different cases.

To simplify the discussion initially we have restricted to scalar/vector/tensor valued data on two or three dimensional grids. It was noted that a initial removal of low-importance features have to be performed prior to the compression phase and that quite likely those two processes should be related.

Subsequently we have consider the approximation of the function obtained from a simulation by other simplified function that is close in L^p norm. It was noted however that this type of simplification do not carry over information about the underlying dynamics that is important in a number of practical cases. Spinning out of this discussion we have considered the higher order (spline) approximation of functions as well topologically aware binning of the data for the sake of data compression.

Taking inspiration from computational topology, we have discussed ideas of persistence-aware simplification. Performed locally it allows to trim down a number of low persistence features keeping the high persistence feature intact while performing the computations in the distributed domain.

As a subsequent discussion a few days later we have considered a possibility of reconstructing the function from a decorated version of a contour tree. The appropriately chosen decoration would allow us to decompress, to a certain extend, the original data. The fact that we start from the underlying contour tree would allow us to keep the information about the major changes in topology of the data. As in the previously discussed instances, the initial data have to be, at least locally, simplified before the compression phase. That would be the key factor in the efficiency of the obtained procedure and all the possible variations have to be tested on practical examples.

We have also explored the ideas emerging from Persistence Homology Transform. In that instance, for the input being a cubical grid with grey scale valued, a collection of 4 (in the planar case) and 8 (in 3d case) filtrations would give rise to corresponding collection of persistence diagrams. It is then possible to reconstruct the data from such a collection of persistence diagrams. It was noted however that performing the persistent homology computations in the distributed setting for the amount of data we are dealing with do not seems to be practically possible at the moment.

Later in the discussion we classified reconstruction approaches based on the topological structure it uses as input (persistence diagram vs Morse-Smale Complex vs contour tree) and collected existing work. We discussed advantages and disadvantages and agreed on the following main outcome of our discussion:

The Morse-Smale complex is too heavy in 3D and will therefore not be able to compete with the contour tree or persistence diagram based reconstruction. Even though the relation between the critical points gets lost, this will be the approach that scales.

Further, we have came to the conclusion that the most promising optically pleasing and light-weight reconstruction is based on gaussian functions instead of bezier curves, splines, or the heat equation.

Finally we have decided to keep the discussion going in the smaller groups to further explore the most promising topics summarized above.

4.2 Multivariate Data

Christoph Garth (TU Kaiserslautern, DE), Ranita Biswas (IST Austria – Klosterneuburg, AT), Hamish Carr (University of Leeds, GB), Vijay Natarajan (Indian Institute of Science – Bangalore, IN), Vanessa Robins (Australian National University – Canberra, AU), Paul Rosen (University of South Florida – Tampa, US), and Gunther H. Weber (Lawrence Berkeley National Laboratory, US)

The discussion group started out by taking stock of existing approaches for multivariate data analysis:

- **Jacobi sets**[1] consider the relationship between two (or more) Morse functions. For two functions, the sets consists of the critical points of the restriction of one function to the level sets of the second function, or, alternatively, of the points where the gradients of both functions are linearly dependent.
- **Reeb spaces**[2] generalize Reeb graphs to multiple functions. They identify points of the domain that belong to a common component of the preimage of a point in the range.

[1] https://doi.org/10.1016/j.comgeo.2014.10.009

[2] https://doi.org/10.1145/1377676.1377720

They are closely related to fibers, multivariate analogs of level sets. They are connected to the existing mathematical discipline of fiber topology, and are well-defined when the range dimension is no greater than the domain dimension (the "low dimensional case"). For range dimensions greater than domain dimensions (the "high dimensional case"), the Reeb space is not well-defined.

Mapper[3] creates a graph representation of multivariate relationships based on a cover of a function defined over an arbitrary manifold. For the low dimensional case, this amounts to a quantization of the Reeb space. For the high dimensional case, Mapper is still well-defined.

Joint Contour Nets[4] also generalize contour trees and Reeb graphs to multivariate functions by observing connectivity in a quantized range space. Where Mapper depends on a cover of the manifold, Joint Contour Nets depend on a tessellation of the manifold. For the low dimensional case, they are a quantization of the Reeb space, while for the high dimensional case, they are still well-defined, like the Mapper.

Pareto sets[5] arise when applying dominance relations from multi-criteria optimization to several continuous scalar fields. This yields optimal simplices, which can be used to identify features in the data.

Among these, several connections can be observed. For example, Pareto sets are a natural part of Reeb spaces, and have also been shown to be closely connected to the Jacobi set in the case where the dimension of the range is smaller than that of the domain.

Furthermore, we identified several typical use cases for multivariate analysis:

Ensemble[6] **data** captures uncertainty in models through multiple realizations, and can be viewed as multivariate functions by combining all ensemble members into a single function.

Time-varying data can be interpreted as multivariate data in several ways. Naively, each time step of a sampled time-varying function can be interpreted as a single variable. Alternatively, using time as a second variable of scalar data, Jacobi sets can be used to track critical points.

"Native" multivariate data concerns models or data comprising multiple variables that must be considered jointly to achieve meaningful analysis.

These uses cases appear in combination in applications[7].

We identified a number of open questions that would warrant further discussion or possible further work:

- What is relationship between persistence and Reeb analysis, especially in the multivariate case?
- What are "importance" measures (e.g. persistence) for the multivariate case that enable e.g. simplification? Is multidimensional persistence related to phenomenological "importance" measures in some way?
- Should the low-dimensional case (where the dimension of the range is smaller than that of the domain) and high-dimensional case (where the range can be of very large dimension) be treated using fundamentally different methods? (When range is small dimension, pre-images are non-trivial, and thus allow construction of e.g. Reeb space etc.)

[3] https://doi.org/10.2312/SPBG/SPBG07/091-100

[4] https://doi.org/10.1109/TVCG.2013.269

[5] https://doi.org/10.1111/cgf.12121

[6] https://doi.org/10.17226/13395

[7] https://doi.org/10.1109/TVCG.2009.200

- How to approach multiple vector fields (and multi tensor fields), which are arguably within the purview of “multivariate”? There is very little work in this area.
- How useful are comparison-based approaches that first apply (univariate) topological analysis to each variable and then compare the resulting abstractions? Especially, how far do graph comparisons carry?

It appears that a focused survey collecting and comparing work in this area, possibly using a common example, could be helpful in understanding the connections between the different approaches and yield more coherent insight into that substantial open questions.

4.3 Persistence Meets Tensor Fields

Michael Lesnick (University at Albany, US), Ulrich Bauer (TU München, DE), Oliver Gäfvert (KTH Royal Institute of Technology, SE), Ingrid Hotz (Linköping University, SE), Claudia Landi (University of Modena, IT), Emilie Purvine (Pacific Northwest National Lab. – Seattle, US), Primoz Skraba (Queen Mary University of London, GB), Hubert Wagner (IST Austria – Klosterneuburg, AT), Bei Wang (University of Utah – Salt Lake City, US), Eugene Zhang (Oregon State University – Corvallis, US), and Yue Zhang (Oregon State University – Corvallis, US)

In the discussion group, we discussed different approaches to finding hierarchical structures arising from tensor fields. We particularly focused on methods based on persistence and Morse theory.

The discussion started with a review of recent work by Hotz and Wang bringing together well groups with classical tensor field topology (as studied in the visualization community) with the stability properties of well groups (as studied in the applied topology community).

A discussion followed about appropriate metrics to measure the perturbation of tensor fields, which is required to quantify the uncertainty of measurements in the data acquisition process. We discussed the difficulty of extending the ideas employed in classical tensor field topology to dimension above 2.

Part of the discussion focused on an idea for using persistent homology and Morse theory to find “flow lines” in tensor fields. The discussion was motivated by applications to mechanics and diffusion tensor imaging (DTI). Much of our discussion was framed in terms of the well-known problem of tractography in DTI.

Specifically, any field of symmetric bilinear forms on the tangent bundle can be interpreted as a function on the product of the base space with a projective space of directions in the tangent space. The function assigns to each point of the base space and each tangent vector the value of the quadratic form at that point evaluated at the normalized tangent vector. One possible point of departure is to apply Morse theory and superlevel set persistent homology to extract a network of gradient flow lines, which can subsequently be filtered according to their persistence in the filtered Morse complex. A priori, it is not clear how well the gradient directions would align with the principal directions of the tensor field. One issue that has to be addressed is the construction of a metric, required in the construction of a gradient field. The definition of such a metric needs to achieve a balancing of distance and directionality.

4.4 Multiparameter Persistence Algorithm

Ulrich Bauer (TU München, DE), Tamal K. Dey (Ohio State University – Columbus, US), Oliver Gäfvert (KTH Royal Institute of Technology, SE), Michael Lesnick (University of Albany – SUNY, US), Arnur Nigmetov (TU Graz, AT), Primoz Skraba (Queen Mary University of London, GB), Hubert Wagner (IST Austria – Klosterneuburg, AT), and Bei Wang (University of Utah – Salt Lake City, US)

There was sufficient interest in the recently developed algorithm by Dey and Xin for multiparameter persistence computation[8]. This group discussed the algorithm in detail with the goal of improved understanding.

4.5 Motivations for Multivariate Data Analysis

Julien Tierny (CNRS-Sorbonne University – Paris, FR), Ranita Biswas (IST Austria – Klosterneuburg, AT), Georges-Pierre Bonneau (INRIA – Grenoble, FR), Hamish Carr (University of Leeds, GB), Pawel Dlotko (Swansea University, GB), Christoph Garth (TU Kaiserslautern, DE), Ingrid Hotz (Linköping University, SE), Michael Kerber (TU Graz, AT), Claudia Landi (University of Modena, IT), Vijay Natarajan (Indian Institute of Science – Bangalore, IN), Emilie Purvine (Pacific Northwest National Lab. – Seattle, US), Vanessa Robins (Australian National University – Canberra, AU), Paul Rosen (University of South Florida – Tampa, US), and Gunther H. Weber (Lawrence Berkeley National Laboratory, US)

Multivariate data is becoming an increasingly important type of data in a variety of applications. Without loss of generality, multivariate data is typically defined as a multivalued scalar function (of dimension r) defined on the vertices of a simplicial complex (of dimension d). This kind of data mostly emerge in scientific computing from two sources: (i) multi-physics simulation and (ii) ensemble simulation. In the first case (i), several physical phenomena are jointly simulated (for instance, thermodynamics is coupled with magnetism simulation). From a domain expert's perspective, the challenges consist in understanding how the different quantities (modeled by the r components of the multivalued function) relate to each other and how the geometry of such a relation enables to characterize features of interest. Such relations are usually non trivial as the components of the multivalued function can have drastically different function ranges and dynamics. In the second case (ii), ensemble simulations are typically obtained by collecting a large number (r) of simulation outputs for varying input parameters (related to the environment of the system). In this scenario, domain experts need to identify and analyse the structures of interest which are common to all these simulation runs, to build insights about the invariant of the phenomenon. On the other hand, they also need to appreciate the variability of the structures of interest, to understand which features

[8] https://arxiv.org/abs/1904.03766

appear only in a subset the simulation. In a broader context than scientific computing, multivalued function typically occur with natural images ($d = 2$) represented in RGB space ($r = 3$) instead of grayscale space ($r = 1$). They also occur in the case of point cloud data (for arbitrary dimensions d), where multi-parameter filtrations are often considered to estimate the topology of the manifold sampled by the point cloud (for example, distance functions to a sub-level set filtration is an instance of this process).

Participants

- Ulrich Bauer
TU München, DE
- Ranita Biswas
IST Austria – Klosterneuburg, AT
- Georges-Pierre Bonneau
INRIA – Grenoble, FR
- Roxana Bujack
Los Alamos National Laboratory, US
- Hamish Carr
University of Leeds, GB
- Tamal K. Dey
Ohio State University – Columbus, US
- Pawel Dlotko
Swansea University, GB
- Oliver Gäfvert
KTH Royal Institute of Technology, SE
- Christoph Garth
TU Kaiserslautern, DE
- Hans Hagen
TU Kaiserslautern, DE
- Ingrid Hotz
Linköping University, SE
- Michael Kerber
TU Graz, AT
- Claudia Landi
University of Modena, IT
- Michael Lesnick
University at Albany, US
- Vijay Natarajan
Indian Institute of Science – Bangalore, IN
- Arnur Nigmetov
TU Graz, AT
- Emilie Purvine
Pacific Northwest National Lab. – Seattle, US
- Vanessa Robins
Australian National University – Canberra, AU
- Paul Rosen
University of South Florida – Tampa, US
- Filip Sadlo
Universität Heidelberg, DE
- Primoz Skraba
Queen Mary University of London, GB
- Julien Tierny
CNRS-Sorbonne University – Paris, FR
- Hubert Wagner
IST Austria – Klosterneuburg, AT
- Bei Wang
University of Utah – Salt Lake City, US
- Gunther H. Weber
Lawrence Berkeley National Laboratory, US
- Eugene Zhang
Oregon State University – Corvallis, US
- Yue Zhang
Oregon State University – Corvallis, US

Report from Dagstuhl Seminar 19222

Control of Networked Cyber-Physical Systems

Edited by

John S. Baras[1], Sandra Hirche[2], Kay Römer[3], and Klaus Wehrle[4]

1 **University of Maryland – College Park, US,** baras@isr.umd.edu
2 **TU München, DE,** hirche@tum.de
3 **TU Graz, AT,** roemer@tugraz.at
4 **RWTH Aachen, DE,** klaus@comsys.rwth-aachen.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19222 “Control of Networked Cyber-Physical Systems”. Such systems typically operate under very tight timing constraints and at the same time witness an ever-increasing complexity in both size and the amount of information needed to main controllability. Yet, the development of control systems and of communication/computation infrastructures has traditionally been decoupled, so that valuable insights from the respective other domain could not be used towards the joint goal of keeping cyber-physical systems (CPS) controllable. In order to overcome this “black box” thinking, the seminar brought together researchers from the key communities involved in the development of CPS. In a series of impulse talks and plenary discussions, the seminar reviewed the current start-of-the-art in CPS research and identified promising research directions that may benefit from closer cooperation between the communication and control communities.

Seminar May 26–29, 2019 – http://www.dagstuhl.de/19222

2012 ACM Subject Classification General and reference → Evaluation, General and reference → Metrics, Networks → In-network processing, Networks → Network architectures, Networks → Network performance evaluation, Networks → Network protocols, Software and its engineering → Communications management, Applied computing → Command and control, Computer systems organization → Embedded and cyber-physical systems, Networks → Cyber-physical networks

Keywords and phrases Control Theory, Cyber-Physical Systems, Latency, Network Architecture

Digital Object Identifier 10.4230/DagRep.9.5.132

Edited in cooperation with René Glebke

1 Executive Summary

René Glebke (RWTH Aachen, DE)
John S. Baras (University of Maryland – College Park, US)
Sandra Hirche (TU München, DE)
Kay Römer (TU Graz, AT)
Klaus Wehrle (RWTH Aachen, DE)

Motivation and Purpose of the Seminar

Manufacturing cells and factories, transportation systems and various other parts of critical infrastructure such as energy grids have traditionally been controlled via self-contained, centralized systems continuously monitored and reconfigured by humans. The ever-growing

Control of Networked Cyber-Physical Systems, *Dagstuhl Reports*, Vol. 9, Issue 5, pp. 132–141
Editors: John S. Baras, Sandra Hirche, Kay Römer, and Klaus Wehrle
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

complexity and integration of these Cyber-Physical Systems (CPS) into reconfigurable value chains ("Industrie 4.0"), autonomous cars and other services with high reliability requirements necessitates a radical change in the control strategy: Classic controllers will not be able to handle the massive amounts of data generated by these emerging systems, not only because of restrictions with regard to computational power and complexities that might bar human interventions in the processes, but also due to missing or inadequate methods for the control and the interconnection of the devices comprising such systems. Whilst CPS have moderate bandwidth/throughput requirements, often in the range of a few bytes per control or sensor message, they require high delivery success rates and predictable latency bounds for these messages and the computations performed on the data, often in the order of a few milliseconds. Stable controllers can only be developed if a predictable behavior of the communication and computation infrastructure may be assumed. Otherwise, the systems may not reach the desired states or even become unstable, up to the point where they may cause physical injuries or the loss of human life. Hence, a paradigm shift towards real-time oriented communication and computation in CPSs is necessary.

Such a shift can, however, only be achieved by overcoming the traditionally loose coupling in the design of system components in networks. Currently, both the communication systems community and the control systems community consider the components of the respective other field as a "black box" and abstract from the variations. Valuable insights that the other domain might provide towards the joint goal of keeping a CPS controllable may hence not be available. Although solutions have already been developed that bring communication and control closer together for specific use-cases, the abstraction problem has not been approached from a general, overarching perspective.

The purpose of this seminar was hence to bring together experts working in the key communities relevant for the science of CPSs and Cyber-Physical Networking (CPN) to get a clearer and more detailed picture of the most important issues of the control and networking aspects that CPSs/CPNs bear and to identify the mutual relations and influences of the associated fields, in order to overcome the so-far strict abstractions and boundaries that exist, and to sketch a roadmap for further research in the field. The driving question was how it is possible to derive generalizable co-design methods and metrics that support the development of universal networked CPSs/CPNs.

Prior Dagstuhl Seminars have already addressed CPS aspects such as synthesis (Seminar 17201) and verification methods (Seminar 14122), robustness (Seminar 16362), as well as software engineering for control (Seminar 14382), yet none of these have focused on the interaction, interdependencies and the co-design of communication and control.

Participants and Structure

The seminar brought together a total of 30 participants from various fields within the communication and control domains, ranging from promising young scientists to leading authorities within their respective fields, but also including practitioners from industry with a strong research background, as well as representatives from funding organizations.

The first day of the seminar was dedicated to an in-depth introductory session. Besides as short personal introduction with background and current research interests, **each participant was asked to prepare a personal statement answering the following questions:**

- What are the most important problems to solve in the realm of CPS/CPN?
- What are the main scientific challenges and which fields can contribute to them?
- What have we achieved so far, and what are the pitfalls of past and current research?

Each personal statement was followed by a discussion round on the presented individual statements. The statements and discussions proved highly fruitful, as they allowed the organizers and the participants to gain an understanding of the current state and future challenges in the Control of Networked CPS from the different disciplinary perspectives.

Most often, opinions revolved around the need to understand more about the implications of the dynamic behavior of both the controlled systems themselves and of the communication networks. Research so far seems to have primarily focused on the "steady state", as participant termed it. The uncertainties introduced by controlled systems and (especially wireless) networks in coexistence with other systems, however, seem to call for various improvements in CPS/CPN design. Yet, as other participants expressed it, besides having fostered a better understanding of the basics of the respective other fields in recent years by programs such as DFG's Priority Programme 1914 *Cyber-Physical Networking*, "little" has been achieved be community so far, with a major pitfall being "lopsided" methods which are often attributed to "sticking to domain-specific models". Opening these models to incorporate knowledge from other domains, therefore, seems to be a major challenge for the upcoming time.

A further major topic discussed was the need for more realistic and relevant problem settings in the research efforts, since, as one participant put it, "real problems are more complex than a single inverted pendulum". Hence, to avoid "esoteric" research and thus "ending up as an academic field with zero practical impact", CPS/CPN is in the need of "prov[ing] that what we develop is useful/needed" within the upcoming years. This does not mean that basic research has or needs to be concluded in any way. Yet, further opinions voiced more than once regarded energy efficiency and usable abstraction/decomposition methods (which may at times even sacrifice optimality for applicability and efficiency) as interesting research challenges for the upcoming years, which shows that the community has already begun tackling more practical issues recently. A variety of additional comments showed that few, if any, of the issues of CPS/CPN can be considered as solved by today.

Plenary Discussion: Properties of Cyber-Physical Networks

The unexpected intensity of the discussions following the respective personal introductions revealed the extreme variety of opinions on the nature of CPS/CPN and the major challenges in this interdisciplinary field. To facilitate a common understanding, the personal introductions were thus followed by a plenary discussion on which properties define CPS/CPN and make them interesting for scientific study.

It was agreed that – besides the eponymous intertwining of control, networking and the physically tangible world – CPS/CPN are dominated by *uncertainties* of both the systems and their operational environments, *dynamics* of configuration and load, (usually) *limitations* e.g., with respect to the capacity of the network, computation power and energy, a *control objective* that is sought to achieve through the network (if it is not serving pure monitoring purposes), as well as the associated relative administrative and technical *autonomy* of CPN compared to their traditional counterparts. Regarding typical metrics of timing and scale, it was further agreed that **traditional complexity metrics do not apply to CPN**. There often exist intricate and counterintuitive relationships between timing constraints of control and the network, leading to situations in which certain upper- and lower(!)-bounded delays may even be beneficial for the simplification and stabilization of control. Hence, defining the time-criticality of a system is scenario-dependent. Likewise, scaling effects may lead to situations in which too many local observations may prove counterproductive to controllability so that, depending on the scenario at hand, issues arise regarding the "right" amount of information sharing between local and global players in distributed decision-making processes. As such, **conceiving widely-applicable categories for the complexity of CPS/CPN was identified as an open problem**.

Impulse Talks & Plenary Discussions

For the remaining one and a half days of the seminar, the participants were asked to propose impulse talks on topics related to their respective areas of control of CPS/CPN research. Each talk served as the basis for a subsequent plenary discussion aimed at identifying worthwhile research directions for the community. **Out of a total of 18 proposed impulse talks, six talks were selected by the organizers.** In the following, we present the major insights from the talks and the discussions.

- The development of next-generation wireless communication technologies such as 5G and the increased efficiency of small-scale mobile devices in general, have fueled the interconnection of ever more devices into large-scale CPS. However, as the number of devices generating data and potentially taking action increase, so do the burdens on controllers and the network. In his talk, Carlos Canudas-de-Wit showed first results pointing at the fact that both state estimation and control may provide sufficient results even when only considering a well-chosen aggregating subset of a system's sensing and actuating nodes, as long as the distribution of these nodes follows a specific structure, which can, however, be found for many real-world scenarios. Together with another technique based on partial differential equations, the results of his work showed that when combining both control- and information-/network theoretic models, as well as upcoming techniques such as in-network computation that may provide the necessary aggregation infrastructure, even systems of immense complexity can be controlled without overloading controllers and networks.
- Another challenge of CPS arises when safety guarantees need to be fulfilled, especially when a failure to meet these guarantees can lead to injury or endangerment of human life. Adam Molin presented an industry perspective on the validation and verification of (increasingly) autonomous vehicles, a field in which scenario-based testing approaches represent the state-of-the-art. While the determinism of systems without humans in the loop may aid in the construction of such scenarios, only probabilistic guarantees can be given when humans are involved in the operation of a system. This fact reflects not just on automotives but on multiple other scenarios discussed in the seminar and highlights the importance of joint analysis methods for the control and the communication components of such systems.
- In her talk on 5G Service Automation, Chrysa Papagianni expressed the view that upcoming mobile networks will witness a shift from open-loop to hierarchical closed-loop control as customers shift towards a pay-per-use scheme for the offered services. Whether the control problems (e.g., regarding network slicing) can be considered to exhibit sub-minute or even real-time requirements (as witnessed in most other systems discussed in the seminar) is still an open question. Yet, considering the anticipated, wide-spread application scenarios of 5G also in the area of CPS/CPN, the seminar identified the issue of base station multi-tenancy as an area for future research within the context of CPN.
- A cornerstone for the successful operation of CPS/CPN are easily-calculable metrics to assess the operational status, as well as to guide the generation, transmission and evaluation of signals within the systems. Vahid Mamduhi in his talk showed that simple age-of-information (AoI) – a common metric applied both by the control and the

communications communities in theoretical and practical scenarios – bases on assumptions that can hardly be met by the systems. As a consequence, AoI needs to be augmented by notions of state, timing constraints of the system, and the objective of the control function (all related to a single piece of information) to really provide benefits. Such metrics are arguably hard to conceive for the general case, yet the talk inspired discussions among the participants regarding sensible metrics with broader applicability.

- From a more communication-oriented perspective, James Gross presented his group's efforts towards determining latency bounds in wireless CPS/CPN. Both a queuing-theoretic and a model checking-based approach (the latter concerning a practical implementation of an ultra-reliable low latency protocol) yielded qualitative results that seem promising. Yet, the practical applicability of such approaches is currently hampered by assumptions regarding distortion that may not hold in practice. In the subsequent discussion, topics included *(a)* the question whether making the network completely deterministic (or the ability to make determinism assumptions) is actually needed and achievable, and how possible compromises may look like, *(b)* to which degree techniques such as software-defined networking, in-network processing, time slicing and standards such as 5G can contribute towards such goals, and *(c)* which interfaces, abstractions and design patterns should exist that allow specifying and proving certain guarantees in CPN, especially regarding the interplay of control algorithms and networks.
- The complexity and variety of communication protocols within automation is addressed by the recent Time-Sensitive Networking (TSN) efforts of the IEEE, which seek to offer a vendor-neutral Ethernet-based solution catering both legacy and future real-time applications, including control. Eventually, the automation pyramid will be transformed into an automation pillar at which TSN serves as the (sole) connectivity provider for control loops which will span the whole automation network from virtualized (/centralized) controllers and the field level. In his talk, Tobias Heer provided an overview of the changes that TSN brings with regard to medium access methods to enable real-time capabilities in Ethernet. While TSN brings significant improvements to wired settings, the subsequent discussion round revolved around the difficulties in achieving this in wireless scenarios. Besides the apparent issues of jamming and/or other attack vectors in wireless control systems, the possibility of trading reliability against capacity and the resulting implications on control algorithms was identified as a research issue.

Conclusion

Throughout the presentations and especially the discussions both during the plenary sessions as well as during off-hour activities, the seminar successfully brought together researchers from control and communication from both academia and industry, and undoubtedly fostered a deeper understanding of the intricate interplay of the disciplines in the research area of CPS/CPN. A variety of open problems and promising research areas were identified, with some in dire need of increased cooperation between the involved fields. This underlines the need in CPS/CPN research for formats valuing open and honest discussions, and both the organizers and the participants hope to be able to continue these discussions in the following years through additional summits and – once the insights gained in this first edition have shown visible impact on the scientific community – possibly another Dagstuhl Seminar. As a concrete follow-up, the organizers and participant James Gross are planning to conduct a seminar in Stockholm/Sweden in 2020 on this diverse research area.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Scales Paradigms in Large-scale networks: micro-control / macro-output

Carlos Canudas-de-Wit (GIPSA Lab – Grenoble, FR)

Joint work of ERC Scale-FreeBAck group
Main reference see several publications at http://scale-freeback.eu/publications/conference-journal/
URL http://scale-freeback.eu

In this talk we presents some results from the ERC Scale-FreeBAck (This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement N° 694209)).

The talk deals with the problems of controlling aggregates of large-scale complex systems with a few inputs (micro-control). Aggregates here are "aggregated" variables functions of the systems state-space variables such as mean values (macro-outputs). Examples of such a class of systems are traffic networks, Brain neural networks, heating systems, among others. The basic idea is to devise a "virtual" aggregated model of the original large-scale system using the scale-free (SF) metric, which indicate that the degree distributions of the associated graph follows an exponential decaying law. Then, we discuss different partitioning algorithms leading to aggregated graphs with the SF desired distribution but also with the suited control/observation properties. In the talk, I also present the mathematical properties necessary for the average observability.

In the second part of the talk, I present a different alternative for cutting system complexity, which consist in representing a large traffic network as a continuum. That is, to approximate a large-scale dynamic graph (where each node represent a variable), by a Partial Differential equation. The objective of this second approach is to use the PDE model for designing boundary estimators and control.

3.2 The future of automation networks in the IIoT; The impact of Time Sensitive Networks

Tobias Heer (Hochschule Albstadt-Sigmaringen, DE)

TSN is a technology that will enable real-time scheduling in standard industrial ethernets. This will facilitate the transformation of industrial network architectures from the classical automation pyramid towards the new paradigm: the automation pillar.

3.3 Modeling of Uncertainty vs. Reality: A Dilemma?

James Gross (KTH Royal Institute of Technology – Stockholm, SE)

This talk discusses modeling and interfacing issues in networked CPS from the perspective of the network, and more importantly, from the perspective of wireless systems. Wireless systems are fundamentally subject to random variations. In theory, models exist of these variations which allow a great deal of reasoning with respect to information-theoretic, communication-theoretic or queuing-theoretic perspectives. In the first part of the talk we explore recent contributions of our group with respect to queuing-theoretic aspects. For networked CPS this is interesting, as it allows a reasoning about the likelihood of a wireless system to exceed predetermined latency thresholds, so called delay violation probabilities, which have practical applications with respect to safety layers. However, the achieved results are theoretic in nature, and allow at best qualitative insights into design trade-offs of future systems. From a different perspective, we discuss other efforts of the research group to capture the delay violation probability of an implemented system by probablistic model checking. This effort is more practical in nature, and at least with respect to ex-post analysis, it possible to bound the error behavior of an implemented wireless system, if the channel parameterization is chose correctly.

3.4 Age-of-Information in net-CPS

Mohammad Hossein Mamduhi (KTH Royal Institute of Technology – Stockholm, SE)

In this talk we discuss the advantages and disadvantages of using age-of-information (AoI) metric in networked control systems. AoI has emerged as a concept that models delay from the receiving end point of view and has shown to be beneficial in communication society. Whether AoI can be employed when quality of control is of utmost interest is still not fully understood. Therefore, we try to summarize the results of some of our early works that quantify the quality of control when AoI is used as a metric for delay and show that age, in its original formulation, is not all that matters in networked systems. Comparing the efficiency of using AoI with the other conventional approaches such as vale-of-information (VoI) clearly shows that AoI is under-performant in many cases. We discuss that age can be used in other formats than its original linear formulation to capture more of the requirements of the networked control systems, e.g. nonlinear age functions or state-dependent age functions.

3.5 A Safety Perspective for Future Mobility

Adam Molin (Denso Automotive – Eching, DE)

Complexity of automotive systems is steadily growing witnessed through the increasing level of autonomy, increased amount of real-time data, and increasing number of interconnections. In order to tame the complexity of such networked CPS and to be able to provide safety guarantees under the vast degree of uncertainties, current systems engineering practices need to be reconsidered. This talk displays recent activities within the automotive domain towards this endeavor. Herein, scenario-based verification and validation (V&V) plays a focal point, in which coverage metrics need to be defined within the operational design domain. In accordance with the safety of the intended functionality, the V&V methods shall aim at minimizing the set of unknown critical instances related to the tested automated driving function. Finally, new opportunities and new V&V challenges emerging from connected mobility are outlined in form of infrastructure-supported decision making and the ability of data collection and update mechanisms.

References

1 Leitner, A., Watzenig D., Ibanez-Guzman, J., (Eds.). *Validation and Verification of Automated Systems – Results of the ENABLE-S3 Project.* Springer International Publishing, Switzerland, 2019.

3.6 5G Service Automation

Chrysa Papagianni (Nokia Bell Labs – Antwerp, BE)

5G systems are set out to address the business contexts of 2020 and beyond, by enabling new network and service capabilities, opening up innovation opportunities for vertical markets. Communication service providers should be able to provide tailor-cut solutions to service requests from the verticals over the same network infrastructure. Network slicing provides a solution for realizing 5G's vision of supporting the highly diversified network needs of emerging applications involving Cyber-Physical Systems (e.g., smart manufacturing, smart grids and railway cyber physical systems). However, to support the multiplicity and demands of emerging networking applications, we need to fully automate network slice management and orchestration. Automation should be enabled throughout all phases of the network slice lifecycle through optimized closed-loop control at different levels and time-scales. In this talk, we discuss the envisioned 5G service automation architecture. We focus on the developing challenges related to network slice management and orchestration and data plane programmability. These research topics are investigated in the context of the forthcoming 5G trials, planned in the framework of the 5G Public Private Partnership co-led by the European Commission and European ICT industry.

Participants

- John S. Baras
University of Maryland – College Park, US
- Sankar Basu
NSF – Arlington, US
- Marcel Carsten Baunach
TU Graz, AT
- Carlos Canudas-de-Wit
GIPSA Lab – Grenoble, FR
- Georg Carle
TU München, DE
- Aaron Ding
TU Delft, NL
- Rolf Findeisen
Universität Magdeburg, DE
- Hannes Frey
Universität Koblenz-Landau, DE
- René Glebke
RWTH Aachen, DE
- James Gross
KTH Royal Institute of Technology – Stockholm, SE
- Andrei Gurtov
Linköping University, SE
- Tobias Heer
Hochschule Albstadt-Sigmaringen, DE
- Thorsten Herfet
Universität des Saarlandes, DE
- Sandra Hirche
TU München, DE
- Wolfgang Kellerer
TU München, DE
- Na Li
Harvard University – Cambridge, US
- Mingyan Liu
University of Michigan – Ann Arbor, US
- Mohammad Hossein Mamduhi
KTH Royal Institute of Technology – Stockholm, SE
- Adam Molin
Denso Automotive – Eching, DE
- Ehsan Nekouei
KTH – Stockholm, SE
- Chrysa Papagianni
Nokia Bell Labs – Antwerp, BE
- Daniel Quevedo
Universität Paderborn, DE
- Kay Römer
TU Graz, AT
- Wolfgang Schröder-Preikschat
Universität Erlangen-Nürnberg, DE
- Olaf Stursberg
Universität Kassel, DE
- Sebastian Trimpe
MPI – Stuttgart, DE
- Klaus Wehrle
RWTH Aachen, DE
- Herbert Werner
TU Hamburg-Harburg, DE
- Gerhard Wunder
FU Berlin, DE
- Marco Zimmerling
TU Dresden, DE
- Martina Zitterbart
KIT – Karlsruher Institut für Technologie, DE