DAGSTUHL
REPORTS

Volume 9, Issue 2, February 2019

Dagstuhl Reports, Vol. 9, Issue 2 **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
August, 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.9.2.i

Aims and Scope
The periodical *Dagstuhl Reports* documents the program and the results of Dagstuhl Seminars and Dagstuhl Perspectives Workshops.
In principal, for each Dagstuhl Seminar or Dagstuhl Perspectives Workshop a report is published that contains the following:
- an executive summary of the seminar program and the fundamental results,
- an overview of the talks given during the seminar (summarized as talk abstracts), and
- summaries from working groups (if applicable).

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

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

Report from Dagstuhl Seminar 19061

Visual Analytics of Multilayer Networks Across Disciplines

Edited by

Mikko Kivelä[1], Fintan McGee[2], Guy Melançon[3], Nathalie Henry Riche[4], and Tatiana von Landesberger[5]

1 **Aalto University, FI,** mikko.kivela@aalto.fi
2 **Luxembourg Inst. of Science & Technology, LU,** fintan.mcgee@list.lu
3 **University of Bordeaux, FR,** guy.melancon@u-bordeaux.fr
4 **Microsoft, USA,** nathalie.Henry@microsoft.com
5 **TU Darmstadt, DE,** tatiana.von.landesberger@gris.tu-darmstadt.de

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19061 'Visual Analytics of Multilayer Networks Across Disciplines'.

Networks, used to understand systems, often contain multiple types of nodes and/or edges. They are often flattened to a single network, even though real-world systems are more accurately modelled as a set of interacting networks, or layers, with different node and edge types. These are so-called multilayer networks. These networks are studied by researchers both in network visualization and in complex systems – the domain from which the concept of multilayer networks has recently emerged. Moreover, researchers in various application domains study these systems, e.g. biology, digital humanities, sociology and journalism. These research areas have shown parallel individual developments. Therefore, one of the aims of the seminar was to bring together an interdisciplinary community of researchers and practitioners of different disciplines. This interdisciplinary community discussed existing solutions, open challenges and future research directions for visual analytics of multilayer networks across disciplines.

The seminar was attended by researchers from information visualization, visual analytics, complex systems and application domains. The application domains covered digital humanities, social sciences, biological sciences, and in public health research (25% of attendees were from these fields). The seminar not only provided multiple application domains for the visualization experts, but also also provided the domains experts with different groups of visualization experts in breakouts sessions, to expose them to multiple approaches to solving their problems. Building on this close working relationship between the visualization and domain experts, working groups were defined to determine which are the important challenges for multilayer network visualization. A number of sub-topics were identified that require further research: *A unifying visualization framework*, *Novel Visual Encodings*, *Analytic* and *Attributes*, *Interaction*, *Evaluation*, *Use Cases* and *Human Factors*. The outcomes of the seminar should stimulate collaborative research on these topics between our community, complex networks, and wide range of application domains for the visual analytics of multilayer networks

Seminar February 3–8, 2019 – http://www.dagstuhl.de/19061

2012 ACM Subject Classification Human-centered computing → Graph drawings, Human-centered computing → Information visualization, Human-centered computing → Visualization theory, concepts and paradigms, Human-centered computing → Visualization design and evaluation methods

Keywords and phrases biological networks, complex systems, geographic networks, graph visualization, multilayer network visualization, social network analysis, visual analytics

Digital Object Identifier 10.4230/DagRep.9.2.1

Edited in cooperation with Björn Zimmer

Visual Analytics of Multilayer Networks Across Disciplines, *Dagstuhl Reports*, Vol. 9, Issue 02, pp. 1–26
Editors: Mikko Kivelä, Fintan McGee, Guy Melançon, Nathalie Henry Riche, and Tatiana von Landesberger
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Tatiana von Landesberger (Technische Universität Darmstadt, Germany and Karsruher Institut für Technologie, Karlsruhe, Germany, tatiana.von.landesberger@gris.tu-darmstadt.de)
Fintan McGee (Luxembourg Institute of Science & Technology (LIST), LU, fintan.mcgee@list.lu)

Introduction

The topic of multilayer networks has recently emerged from the field of complex systems, however many of the of the fundamental concepts and ideas have existed for some time, in fields such as sociology, and often under different nomenclature, such as multimodal, heterogeneous or multiplex networks. The multilayer network framework of Kivelä *et al* [1] has collected many of these concepts and different labels, along with example data sets, allowing us to recognize the multi-disciplinary importance of multilayer networks as a topic. Despite the importance of this topic, it is only recently that the visualization community is beginning to consider approaches for the visual analytics of multilayer networks. This seminar was the first to bring together practitioners from multiple domains to discuss the visual analytics of multilayer networks. These fields included data visualization, complex systems, digital humanities, biological sciences, health informatics, and sociology. The primary goal of this seminar was to bring together these thinkers and practitioners from different disciplines to drive forward new advances on the topic. The seminar was designed to foster discussions between researchers and designers of visual analytics tools, those who define the underlying theory, and the the end-users of these tools. To push research further and produce significant impact in industry and general public practices, the research community needs to establish a deeper collaboration between data scientists and researchers from applications domains (e.g. biologists, social scientists, business analysts, journalists, physicists), who collect and analyze the data; and researchers in maths, physics and computer science who push the state-of-the-art, producing visualization and analysis models, algorithms and tools. This deeper collaboration starts with building an understanding of the needs and tasks of network analysts. This seminar was an important first step, leveraging cross domain synergies with the goal of identifying the shared underlying problems and helping to solve them. The domain experts presented their domain problems early on in the seminar, and then interacted with two different sets of visualization experts in two separate breakout sessions. The motivation for this was to expose the visualization experts to many different domain problems and to expose the domain experts to multiple approaches to their problems. Our goal was to not only to advance research in the field of visualization, but also to provide techniques to help the domain experts to advance research in their own field. Interdisciplinary intersection was a key part of the methodology of our seminar.

Seminar Topics

The seminar featured talks and working groups that discussed topics on visualization, analysis, theory and applications of multilayer networks (see Sections 3 and 4). The application domain focus was maintained throughout the seminar. Experts from application domains gave talks in the first day and a half highlighting the problems they encountered. Then there were two breakout sessions where each experts was assigned a different group of visualization experts, allowing the domain experts to brainstorm solutions to their problems with different sets of visualization experts.

Talks

The talks brought the interdisciplinary participants initial information on a) current application problems dealt with in the area of multilayer networks and b) current visualization, analysis and systems solutions.

The purpose of talks by application experts was to make sure that the potential solutions provided by the interactive visualizations and analytics fully meet the requirements of those who actually use them, i.e. the system biologists, social network analysts, historians, etc. Therefore, the talks provided understanding of the data and problems/tasks/goals when analyzing multilayer graphs by the domain experts. The talks covered application areas of social networks by A. Cottica and by M. Magnani, information circulation in an international organization by M. Grandjean, digital humanities by M. Düring , multi-omics data by S. Legay, population health by M. McCann, digital ethnography by A. Munk(see sections 3). These talks allowed the visualizations and complex systems theory experts to gain some insight into the domain experts problems. As we also wanted the domain experts to be exposed to multiple approaches to their problems, we had two breakout sessions after all of the domain experts presented their personal topics. In these breakout sessions, each domain experts was assigned a small group of visualization researchers to further brainstorm, mapping visualization problems to domain problems. Different researchers were assigned to each domain expert for each session. This exposed the domain experts to multiple visualization approaches, and allowed for synergies between application domain problems to be identified by the visualization researchers. At the end of each breakout session, each groups gave a short report back to other participants, allowing for further discussion and cross fertilization of ideas. This approach ensured that that both the domain experts and the visualization experts had a wide range of ideas to explore as part of the working groups in the later half of the seminar.

The purpose of talks by visualization, analysis and systems experts was to present currently available solutions to multilayer network visualization and analysis (see also Section 3). These talks were dispersed throughout out the week. The talk topics were: an introduction to multilayer networks by F. McGee, a complex systems perspective on the concept of multilayer network by M. Kivelä, survey of multilayer visualizations by G. Melançon, Py3plex library for visualization by B. Škrlj, interaction with multilayer network visualization by B. Renoust. This allowed application experts to get to know the advantages and limitations of existing solutions. The talk schedule was flexible, for example, due to a high level of interest form all attendees M. Kivelä gave a second question and answers session to his talk the following day.

Working Groups

At the midpoint of the week we defined the working groups. The breakout session stimulated a large amount of discussion and ideas across participants of all disciplines. Following on the breakout sessions discussions, all seminar participants wrote down topics and ideas that were that were of interest to them of pieces of paper, which were affixed to a board. Similar topics were re-positioned closer together on the board, until all participants reached a consensus of five topic areas for discussion within working groups. The resulting working groups were as follows:

- *Unifying Terminology and visual analytic approaches*: One open problem of multilayer network analysis and visualization is the inconsistent terminology across disciplines. There are many different names given to networks with such characteristics, outlining the current lack of consistent definitions between disciplines, such as heterogeneous,

multi-faceted, multi-modal, or multi-relational networks, amongst others (see [1]) and in the vast majority of cases it is possible to model them as multilayer networks. The discussion group assessed various types of networks from visualization, application and systems perspective. It discussed possible unification of these perspectives in one visual analytics framework and identified open challenges (see Section 4.3).
- *Analytics, Communities Comparison and attributes*: Visual analysis of multilayer networks is also concerned with the exhibition of salient properties and patterns in data. Salience in networks is often captured through metrics (networks statistics) while patterns most often correspond to particular subsets of entities (nodes and edges). Layers bring additional complexity to the computation of these metrics and patterns, as metrics and patterns may need to be computed across several layers. The visualization of the computed metrics and patterns needs to consider also these layers, thus, posing challenges to the data presentation. This working group analyzed the current network metrics and proposed novel metrics specifically for multilayer networks (see Section 4.4).
- *Interaction (and Layer Creation)*: (see Section 4.2) This topic concentrated on interactive creation of layers in networks. While the input multilayer network may have predefined layers, in many use cases, the layers need to be adapted to the analytical task during network exploration. This working group has gathered requirements for interaction with layers, surveyed current solutions and their limitations. They have proposed novel approaches that will be pursued after the seminar.
- *Visual Encodings* The complex relationships between complex structures mean that traditional interactive visualizations need to be enhanced. Researchers from the various domains can exchange their ideas and thus start novel avenues in interactive visualization. The discussion of this working group focused on the visualization design – encodings. The group identified main requirements for visualization: aggregations, interactive layer editing, overview of all layers, details of an individual layer and exploration paths – top-down versus bottom up (see Section 4.5). These requirements are used to derive a design space of possible visualization approaches in future.
- *Human Factors and Multilayer Networks* This topic focused on the user's point of view in the design of multilayer network visualization. This is a challenge as the complexity of multilayer networks results in a significant amount of cognitive load on the users. The group collected results from related work that can be used as guidelines for designing multilayer visualizations. It also identified gaps in literature for future research (see Section 4.1).

Seminar Outcomes

During the seminar, a number of sub-topics were identified that require further research: *A unifying visualization framework*, *Novel Visual Encodings*, *Analytics* and *Attributes*, *Interaction*, *Evaluation*, *Use Cases* and *Human Factors*.

- **A unifying visualization framework for multilayer networks:** Currently, multilayer networks are referred to across communities using various names and concepts. A novel unified conceptual framework for multilayer network is needed that would be used for visualization, interaction and analytics purposes. It should extend the underlying mathematical framework [1] to meet the needs of the data and tasks associated with the various use cases, as well as existing visualization and interaction concepts.

- **Novel visual encodings:** The existing visualization techniques have limited scope for the broad range of data and tasks in the applications of multilayer networks. Therefore, novel visual encodings need to be researched that to enable data exploration across layers.
- **Interaction:** Visual exploration and analysis of multilayer networks requires novel interaction techniques that would allow to browse across layers and also to create new layers during the exploration process.
- **Interdisciplinarity:** The wide range of application domain problems sets novel problems that may be best addressed by new visualization approaches. The development of novel solutions for visual analysis of multilayer networks requires joined forces of application, visualization and analysis experts.
- **Multiple layers and attributes:** The complexity of multilayer networks often includes an additional dimension: The multivariate nature of node and edge attributes. This information needs to be encoded in the visualization and supported in analytical functions. This raises novel challenges.
- **Network Analytics:** Visual network analysis also covers the understanding the analytical relationship between layers (with respect to structure and/or attributes) and the layer comparison. The limitations of current analytical approaches and network metrics raises many interesting challenges and opportunities for developing new metrics for the multilayer use case.
- **Evaluation & Human Factors:** The human perspective on the complexity of the network structure and its visualization needs to be assessed. It covers a) the perceptual and cognitive aspects when interactively exploring the networks and b) a thorough empirical evaluation of the analytical paths and insights. The existing methodologies for such research should be adapted for the multilayer network case.

These topics will be discussed in the follow-up VIS 2019 Workshop 'Challenges in Multilayer Network Visualization and Analysis'. The workshop is co-organized by Dagstuhl Seminar organizers and participants: Fintan McGee, Tatiana von Landesberger, Daniel Archambault and Mohammad Ghoniem. The seminar will feature keynote, paper and poster sessions as well as discussion rounds on the above-mentioned topics.

References

1 Mikko Kivelä and Alex Arenas and Marc Barthelemy and James P. Gleeson and Yamir Moreno and Mason A. Porter (1998). *Multilayer networks*. Journal of Complex Networks.

2 Table of Contents

3 Overview of Talks

3.1 Visualizing multilayer networks with Py3plex

Blaž Škrlj (Jozef Stefan Institute – Ljubljana, SI)

Joint work of Blaž Škrlj, Jan Kralj, Nada Lavrač
Main reference Blaz Skrlj, Jan Kralj, Nada Lavrac: "Py3plex: A Library for Scalable Multilayer Network Analysis and Visualization", in Proc. of the Complex Networks and Their Applications VII – Volume 1 Proceedings The 7th International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2018, Cambridge, UK, December 11-13, 2018., Studies in Computational Intelligence, Vol. 812, pp. 757–768, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-05411-3_60

Real-world complex networks frequently consist of separate layers, representing either the same entity in different contexts (multiplex), or interacting, entirely different entities (multilayer). In this talk we first describe how multilayer networks are visualized currently along with the main drawbacks of such approaches. Next, we discuss Py3plex, a python library offering a novel multilayer visualization layout. It first projects individual layers across a diagonal. Here, each layer can have a unique layout. Next, inter-layer edges are drawn as curves between layers. We have identified layout algorithms as one of the main bottlenecks in graph drawing. We propose an embedding-based layout computation, where the network is first embedded to a high dimension (e.g., 512) and next compressed to two dimensions. The obtained projections are used as the starting (initial) positions for only a handful of iterations of the Barnes-Hut force-directed layout. The proposed approach scales to networks with hundreds of thousands of nodes and millions of edges on an of-the-shelf laptop.

3.2 Semantic social networks: a multilayer interpretation

Alberto Cottica (Edgeryders – Brüssel, BE)

Joint work of Alberto Cottica, Guy Melançon, Amelia Hassoun, Jason Vallet, Benjamin Renoust
Main reference Alberto Cottica, Amelia Hassoun, Jason Vallet, Guy Melançon: "Semantic Social Networks: A New Approach to Scaling Digital Ethnography", in Proc. of the Internet Science – 4th International Conference, INSCI 2017, Thessaloniki, Greece, November 22-24, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10673, pp. 412–420, Springer, 2017.
URL http://dx.doi.org/10.1007/978-3-319-70284-1_32

Semantic social networks are a method to treat large ethnographic corpora. Starting from a conversational medium, like an online forum, semantic social networks are built by ethnographic coding of the utterances in the conversation. The result is a social network of interaction, where the edges encode meaning.

Semantic social networks can be interpreted as multiplayer networks. A social layer encodes interaction between key informants in the ethnography; a semantic layer encodes co-occurrence between ethnographic codes. I discuss how best to interrogate each layer to address questions relevant to ethnographic research.

3.3 Digital Humanities Data

Marten Düring (University of Luxembourg, LU)

This talk stressed the interpretative nature and purpose of data encountered in the Digital Humanities (DH) sphere, especially data which is collected by humanities researchers from either unstructured text, serial sources, such as membership lists, or existing databases. The DH community has three main expectations when collaborating with Visual Analytics scholars: the joint development of custom-fit visualisation solutions, trustworthy representations of their data as well as insights of actual significance for their own work. Differences arise with regard to the strong interpretative aspect in DH data collection, the common absence of a testable ground truths as well as more interpretative research practices such as close reading as opposed to quantifications. The talk pointed out potential tensions in the relationship between DH and Visual Analytics scholars which arise from the goal to develop highly original contributions in VA and the need for robust tools in DH. These tensions can be overcome by fostering close relations between the fields which are based on the early definition of success criteria, ease of integration into DH workflows, sufficient time for skill development, and the acknowledgment of DH data particularities.

3.4 Mapping information circulation within an international organisation in the 1920s – An example of historical multilayer analysis

Martin Grandjean (Université de Lausanne, CH)

Main reference Grandjean Martin: "Les réseaux de la coopération intellectuelle, la Société des Nations comme actrice des échanges scientifiques et culturels dans l'entre-deux-guerres", University of Lausanne, 2018, 600 p.

URL https://halshs.archives-ouvertes.fr/tel-01853903

We propose to analyse the activity and connectivity of the International Committee on Intellectual Cooperation (ICIC) of the League of Nations through a fine indexing of its correspondance archives. Created in 1922, this committee brought together leading scientists such as Henri Bergson, Albert Einstein, Marie Sklodowska-Curie and Hendrik Lorentz and laid the foundations of UNESCO after the Second World War. The rise and bureaucratization of this organization during the 1920s is accompanied by a rapid increase of the number of documents produced and received as well as a complexification of information flows. By mapping this circulation in a network composed of more than 3.000 individuals involved in intellectual cooperation during its early years (approx. 30.000 documents), this method reveals the main organizational trends while highlighting the situation of actors that are so far little studied in this context. This exploratory "datafication" of the archives of the League of Nations leads us to reconsider and recontextualize the personal commitment of the individuals who made up the ICIC.

The development of a multilayer model that allows a comparison between the official institutional framework of the international organizations of this time and the structuring of information exchanges at an individual level helps to highlight the discrepancy from a

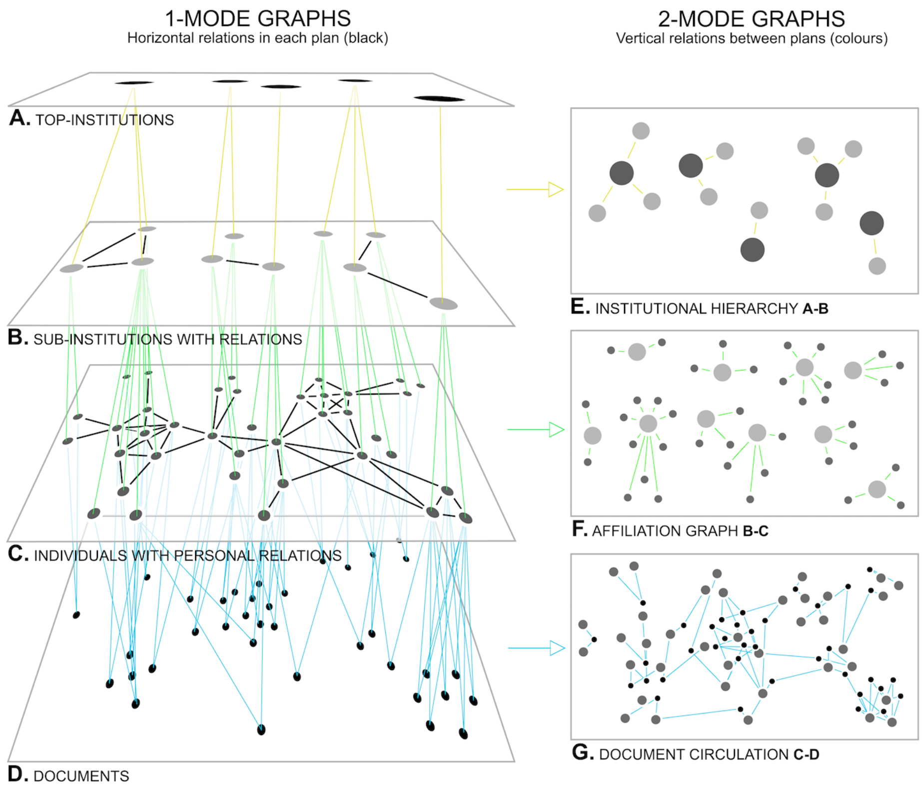


Figure 1 Example of a multilayer model.

hierarchical level to another. In our example, we propose to use this multilayer model to flatten the hierarchical structure of the two upper levels so that they contain the individual nodes. Creating such a stable visualisation is the condition for a comparative analysis of the edges connecting groups in time of based on a thematic filtering of the data.

References

1 Martin Grandjean. *Les réseaux de la coopération intellectuelle, la Société des Nations comme actrice des échanges scientifiques et culturels dans l'entre-deux-guerres*. University of Lausanne, Switzerland, 2018.
2 Martin Grandjean. *Analisi e visualizzazioni delle reti in storia: l'esempio della cooperazione intellettuale della Società delle Nazioni*. Memoria e Ricerca, 25, 2, 2017, 371-393.
3 Martin Grandjean. *Multimode and Multilevel: Vertical Dimension in Historical and Literary Networks*. Digital Humanities 2017, Montreal.

3.5 Multilayer Networks

Mikko Kivelä (Aalto University, FI)

Main reference Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter: "Multilayer networks", J. Complex Networks, Vol. 2(3), pp. 203–271, 2014.
URL https://doi.org/10.1093/comnet/cnu016

Network science has been very successful in investigations of a wide variety of applications from biology and the social sciences to physics, technology, and more. In many situations, it is already insightful to use a simple (and typically naive) representation as a simple, binary graph in which nodes are entities and unweighted edges encapsulate the interactions between those entities. This allows one to use the powerful methods and concepts for example from graph theory, and numerous advances have been made in this way. However, as network science has matured and (especially) as ever more complicated data has become available, it has become increasingly important to develop tools to analyse more complicated structures. For example, many systems that were typically initially studied as simple graphs are now often represented as time-dependent networks, networks with multiple types of connections, or interdependent networks. This has allowed deeper and more realistic analyses of complex networked systems, but it has simultaneously introduced mathematical constructions, jargon, and methodology that are specific to research in each type of system. Recently, the concept of "multilayer networks" was developed in order to unify the aforementioned disparate language (and disparate notation) and to bring together the different generalised network concepts that included layered graphical structures. In this talk, I will introduce multilayer networks and discuss how they have been studied and visualised in the complex networks literature.

3.6 Multi-omics Data in Non-model Plants. Treatment, Integration and Visualization

Sylvain Legay (Luxembourg Inst. of Science & Technology, LU)

These last decades, the progress in plant sciences has vastly increased the amount of data that researchers must deal with. The recent high-throughput technologies offer extremely powerful tools, but also raise many issues linked to big data management, as well as visualization of complex biological processes. The realm of living organisms constitutes a natural multilayer system with multiple type of driven/non-driven, intra/inter-network and time-dependent interactions. Currently, integrative sciences usually focused on the "holy trinity": Gene expression, protein expression and metabolites. In order to simplify the investigation of these three heavily complex networks and to connect them to concrete biological meanings, scientists developed new layers (networks and/or databases) considering metabolic pathways, biological processes or molecular function to name a few. This strategy offered the scientific community useful tools to deeply investigate their topic of interest but it also increased the number of inter/intra network interactions. By the time, the visualization and investigation of these interconnected networks became a challenge. This is even more of an issue for scientists from applied research, which are dealing with non-model plants with species-specific metabolic pathways. In such cases, knowledge repositories are more incomplete due to lower efforts engaged to build them. As research topics are more and more turned to technology transfer and because biological networks are growing exponentially, the need of highly efficient, user-friendly visualisation tools to investigate complex biological processes is a crucial challenge for years to come.

3.7 Multilayer Social Networks

Matteo Magnani (Uppsala University, SE)

Joint work of Matteo Magnani, Luca Rossi
Main reference Mark E. Dickison, Matteo Magnani, Luca Rossi: "Multilayer Social Networks", Cambridge University Press, 2016.
URL https://doi.org/10.1017/CBO9781139941907

The objective of this talk is to introduce some of the main types of multilayer networks used to analyse social networks. We first give a historical presentation of early examples of multilayer networks /multiplex, multi-mode, etc.) leading to the proposal of a unified model (by Kivela et al.). Then, we present two concrete examples of multilayer social networks extracted from online social media, including different types of vertices (users and messages), different types of edges (communication, following, etc.), and attributes such as text and time. We also provide a list of open questions that would require advanced visualization methods to be addressed.

3.8 Multilayer Networks and Population Health

Mark McCann (University of Glasgow, GB)

Modern population health science has had a predominantly medical focus. Analysis often relies on the assumption of "no interference" – meaning that the health risk exposure of one individual does not affect the outcomes of other individuals.

Such an assumption is untenable in social systems. There are many relational features of experience, attitude formation and behaviours that relate to health outcomes. Network science, and thus information visualisation have important roles to play in population health.

Conceptualising population health as a feature of a complex system has been identified as a priority for the discipline. A key task is thus moving from conceptualisation, to practical application of complex systems science methods. Adopting a multilayer network approach may help to advance population health applications.

The complex system of a school may comprise pupil interaction: 'friends', 'foes', and 'prestigious' peers; co-participation of pupils in school events; the causal connections between health behaviours, and symptoms of poor physical and mental health; as well as the social transmission of health behaviours and outcomes.

Considering each of the elements above as multiple aspects of a multilayer network makes it easier to study the interactions within and between layers; for example, change in pupil interactions over time, or across levels of 'friend' and 'prestige' networks, or across levels of the social-ecological model of health. Layering a school social system may provide a relational equivalent of conditioning in causal inference, where statistical associations are layered by conditioning variables.

The sharing of ideas around visualising multilayer systems at Dagstuhl shows promise for the development of systems methods for population health science; and I strongly encourage continued collaborative work on visualising population health systems.

3.9 Multilayer Networks Overview and Examples across Domains

Fintan McGee (Luxembourg Inst. of Science & Technology, LU)

Joint work of Mohammad Ghoniem, Fintan McGee, Guy Melançon, Benoît Otjacques, Bruno Pinaud
Main reference Mohammad Ghoniem, Fintan McGee, Guy Melançon, Benoît Otjacques, Bruno Pinaud: "The State of the Art in Multilayer Network Visualization", Computer Graphics Forum, 2019.
URL https://doi.org/10.1111/cgf.13610

Multilayer networks may be a new term to some in the field of visualization, however there are many related concepts and ideas that will be familiar to those with experience in network visualization. The goal of this talk was to provide a gentle introduction from a network visualization perspective, justifying the use of multilayer networks and differentiating their visualization from related approaches. Multilayer network visualization is required as complex systems are often not accurately described by a single network, and modelling it as such means making many simplifying assumptions. Real life is better modelled as several interdependent subsystems or layers, as real-world systems are not general 100% closed and independent. For example, the professional and personal social networks of an individual can be considered independent networks, but in reality one may interact with the other. A dramatic changes in a professional network (resulting perhaps from a change of job) may precede a more gradual change of personal social network. Within biological networks interactions at different "omics" levels (e.g. proteomic, metabolomics and genomics) are best considered multiple layers rather than individual networks.

Multilayer networks often crop up under different nomenclature (see Kivela *et al.* [1]), and there some similarities to existing visualization approaches. Scheiber *et al.* [3] describe "heterogeneous networks" as a class of multivariate network, however multilayer networks as described by Kivela *et al.* are more complex, allowing multiple aspects to be defined characterising layers. While layer information can be reflected in an attribute it is important to remember that layers are based on a physical or conceptual reality related to the system being modelled, and is not just an arbitrary slicing of data.

The facets of faceted data visualization (see Hadlak *et al.*, [2]) may seem superficially similar to layers, however facets are more focused on visualization semantics (relating to the type of data) rather than a structure within the data, layers can interact with each other and are entities within their own right. There is also some overlap with dynamic graph visualization as time can be considered an aspect of a multilayer networks, characterising different layers. Multilayer networks visualization are of course more general however integrating time among other aspects may also raise other challenges. A bipartite graph may also be considered a type of multilayer network, albeit one without edges within the layers. However, some bi-partite analytics may be of interest for understanding the relationships between layers. For more detail on visualization approaches related to multilayer network visualization see [4].

The application domains for multilayer network visualization are numerous, and include infrastructure, biology, sociology and digital humanities, among many others. In this talk we briefly discussed these fields as well as giving examples of data sources available on line:

- CoMuNe Lab : https://comunelab.fbk.eu/data.php
- Queen Mary, University of London http://www.maths.qmul.ac.uk/~vnicosia/sw.html
- The Colorado Index of Complex Networks (ICON): https://icon.colorado.edu/

References

1 Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter (2014) Multilayer Networks, Journal of Complex Networks, vol 2, issue 3
2 Steffen Hadlak, Heidrun Schumann, Hans-Jörg Schulz (2015). A survey of multi-faceted graph visualization. In Eurographics Conference on Visualization (EuroVis). The Eurographics Association (Cagliary, Italy, 2015), pp. 1–20. doi:10.2312/eurovisstar.20151109.
3 Falk Schreiber, Andreas Kerren, Katy Börner, Hans Hagen, Dirk Zeckzer (2014). Heterogeneous networks on multiple levels. In Multivariate Network Visualization: Dagstuhl Seminar #13201, Dagstuhl Castle, Germany, May 12-17, 2013, Revised Discussions, Andreas Kerren, Helen Purchase Matthew O. Ward,(Eds.). Springe rInternational Publishing, 2014, pp. 175–206
4 Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, Bruno Pinaud (2019). The State of the Art in Multilayer Network Visualization. Computer Graphics Forum, DOI 10.1111/cgf.13610

3.10 Survey on visualization of multilayer networks – Towards a tasks taxonomy

Guy Melançon (University of Bordeaux, FR)

Joint work of Mohammad Ghoniem, Fintan McGee, Guy Melançon, Benoît Otjacques, Bruno Pinaud
Main reference Mohammad Ghoniem, Fintan McGee, Guy Melançon, Benoît Otjacques, Bruno Pinaud: "The State of the Art in Multilayer Network Visualization", Computer Graphics Forum, 2019.
URL https://doi.org/10.1111/cgf.13610

The talk briefly presented a survey work accomplished in parallel with the organization of the workshop, as a teaser of the paper that has been published since then [3].

The talk focused on the task taxonomy part developed in the paper. Tasks on multilayer networks share specificities with those conducted on "ordinary", single layer graphs. However many tasks revolve around the notion of layers as a central device to explore and query the data. Tasks are organized into different categories emerging from existing literature.

- Task category A – Cross layer entity connectivity (e.g., inter-layer path) definitely is one. Numerous papers indeed propose generalization of path-based metrics to multilayer networks, for instance.
- Task category B – Cross layer entity comparison. Being able to observe whether a same node or gorup of nodes have simialr position on different layers (or layer subsets) is a natural question that can be asked.
- Task category C – Layer manipulation, reconfiguration (split, merge, clone, project) allows users to combine layers according to domain questions.
- Task category D1 – Layer comparison based on numerical attributes. Comparing layers is a fundamental task that can either rely on numerical attributes reflecting layer structure. A typical task is to form the distribution of some node or edge statistics and look at correlation values to infer layer similarities.

- Task category D2 – Layer comparison based on topological, connectivity patterns, layer interaction. Layer coparison can also be performed using network topology (link structure), typically looking at how network communities compare for instance.

While we focus here on tasks, it is important to note that layers do not reduce to some operational apparatus. The concept goes far beyond a simple intent to capture data heterogeneity. While it is true this notion is most of the time embodied as nodes and edges of a network being of different "types", its roots lie deeply in sociology [4, 6, 5]. The concept of a multilayer network builds on and encompasses many existing network definitions across many fields, some of which are much older, e.g., from the domain of sociology [1, 2]. Examples of multilayer networks can be found in the domains of biology (the so-called "omics" layers), epidemiology, sociology (in a broad sense, including fields such as criminology, for instance), digital humanities, civil infrastructure and more. Multilayer networks have been explicitly recognised as promising for biological analysis. See [3] for and expanded dicussion and extended references.

References

1 Jacob L. Moreno. Who Shall Survive? Beacon House Inc., 2nd edition, 1953.
2 Lois M Verbrugge (1979). Multiplexity in adult friendships. Social Forces, 57(4):1286–1309.
3 Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, Bruno Pinaud (2019). The State of the Art in Multilayer Network Visualization. Computer Graphics Forum, DOI 10.1111/cgf.13610
4 Ronald Burt and Thomas Schøtt (1985). Relation content in multiple networks. Social Science Research, 14(4):287–308.
5 Emmanuel Lazega and Philippa. E. Pattison (1999). Multiplexity, generalized exchange and cooperation in organizations: A case study. Social Networks, 21:67–90.
6 Nicholas Geard and Seth Bullock (2007). Milieu and function: Toward a multilayer framework for understanding social networks. In Workshop Proceedings of the Ninth European Conference on Artificial Life (ECAL): The Emergence of Social Behaviour, pages 1–11.

3.11 Digital Ethnography

Anders Kristian Munk (Aalborg University, DK)

URL https://docs.google.com/presentation/d/1BElobz_X7W3l5gze83KY96YpODlHJdf43_–NKmnQu0/edit?usp=sharing

In this talk I map the potential overlaps between ethnography, as a conventionally qualiative and situated field practice born in anthropology, and network analysis of big social data typically associated with computational social science. I argue that data science approahces to ethnographic work raise questions about the interface between qualitative and quantiative modes of reasoning, about the way in which findings are grounded, and about the potential effects of media platforms and algorithmic environments. I then provide a case example to illustrate these challenges, namely the contruction and analysis of the the Atlas of Danish Facebook Culture and suggest ways in which multi-layered network analysis might be a solution to some of them.

References

1 Anders Kristian Munk and Torben Elgaard Jensen (2015). Revisiting the histories of mapping. Ethnologia Europaea, 44(2):31.

2 Anders Kristian Munk (2013). Techno-anthropology and the digital native. What is techno-anthropology, pages 287–310

3.12 Interacting with multilayer networks

Benjamin Renoust (Osaka University, JP)

Joint work of Benjamin Renoust, Guy Melançon, Tamara Munzner, Haolin Ren, Marie-Luce Viaud, Shin'Ichi Satoh, Youssef Mourchid, Hocine Cherifi, Mohammed El Hassouni
Main reference Benjamin Renoust, Guy Melançon, Tamara Munzner: "Detangler: Visual Analytics for Multiplex Networks", Comput. Graph. Forum, Vol. 34(3), pp. 321–330, 2015.
URL http://dx.doi.org/10.1111/cgf.12644

In this presentation we introduce three types of interactive visualization with multilayer networks. Through Detangler [1], we introduce the visualization of multilayer networks without displaying a network with multiple layers, all based on careful interaction design and linked highlighting. With the Visual Clouds [2] we introduce the visualization of multilayer networks without displaying actual network. We present different facets captured through different networks with an advanced and coordinated tag-cloud-inspired viusalization that is designed for multimedia search engines. We finally introduce an animated drag-and-drop interaction [3] for the tracking of communities in dynamic multilayer networks. In addition, this paper presents a dataset that combines many characteristics of multilayer networks built on top of movie scripts, though both NLP and computer vision processing [4].

References

1 Benjamin Renoust, Guy Melançon, Tamara Munzner (2015, June). Detangler: Visual analytics for multiplex networks. In Computer Graphics Forum (Vol. 34, No. 3, pp. 321–330).

2 Haolin Ren, Benjamin Renoust, Marie-Luce Viaud, Guy Melançon (2018, September). Generating Visual Clouds from Multiplex Networks for TV News Archive Query Visualization. In 2018 International Conference on Content-Based Multimedia Indexing (CBMI) (pp. 1–6). IEEE.

3 Benjamin Renoust, Haolin Ren, Guy Melançon, (2019). Animated Drag and Drop Interaction for Dynamic Multidimensional Graphs. arXiv preprint arXiv:1902.01564 (PacificVis 2019). IEEE.

4 Youssef Mourchid Benjamin Renoust, Hocine Cherifi, Mohammed El Hassouni (2018, December). Multilayer Network Model of Movie Script. In Complex Networks and their Applications (pp. 782–796). Springer, Cham.https://github.com/renoust/multilayermovies.

4 Working groups

4.1 Human Factors and Multilayer Networks

Kathrin Ballweg (TU Darmstadt, DE), Margit Pohl (TU Wien, AT), and Helen C. Purchase (University of Glasgow, GB)

From the user's point of view the design of multilayer networks is a challenging task. Multilayer networks are very complex resulting in a significant amount of cognitive load on the users. It is an open question how to design such networks so that users can derive insights from such visualizations. The working group 6 on Human Factors and Multilayer Networks identified three main research areas in this context:

1. *Overview of existing evaluation studies on multilayer networks or other complex networks:* There are still very few studies doing evaluations of multilayer networks. In this context, it might be useful to look at studies of other complex networks and try to transfer the results of these studies to multilayer networks. The results of such studies can inform the design of multilayer networks.

2. *Overview of psychological research relevant for the evaluation of multilayer networks:* There are several areas in psychology that might be relevant for evaluation studies of multilayer networks, especially in the area of cognitive load. As mentioned above, increased cognitive load is probably the most serious challenge facing designers of multilayer network interfaces. It has been argued that restricting investigations of visualizations to simple tasks can be at times misleading. Cognitive load theory can form a theoretical foundation for getting a more comprehensive picture of cognitive processes users engage in when interacting with complex visualizations.

3. *Development of tentative recommendations for the design of multilayer networks based on the literature review.*

4.2 Interactive Layer Creation

Fabian Beck (Universität Duisburg – Essen, DE), Marten Düring (University of Luxembourg, LU), Mohammad Ghoniem (Luxembourg Inst. of Science & Technology, LU), Sylvain Legay (Luxembourg Inst. of Science & Technology, LU), Matteo Magnani (Uppsala University, SE), and Jason Vallet (University of Bordeaux, FR)

A multilayer graph may have predefined layers. However, in many practical applications – as it turned out in discussions with domain experts during the seminar – the definition of layers can and should be adapted according to the research question studied. For instance, a social media dataset might contain users, posts, comments, and tags with various possible types of connections between them; research questions related to the spread of information might require different network representations than others related to trending topics. Instead of creating an overarching multilayer network that considers all data entities and all possible relations in one complex structure, it might be more appropriate to let analysts derive appropriate network abstractions on the fly along with querying subsets of the data. Layers of a network can be a powerful abstraction in this process as they reflect different perspectives.

To support interactive layer creation, it is first important to acknowledge that there are different ways to define a layer and several interactive solutions to do so have been already suggested. A certain layer might contain only nodes of a specific type. Similarly, we can use edge types to discern layers. However, layers can also mix different types of nodes and edges. They can be derived from other layers by filtering, flattening, projection, partitioning, or applying set operations like union, intersection, difference, and exclusivity. The starting point of the process might be an overview visualization of the whole dataset but could also be a query interface. Existing querying approaches already provide support to visually define network structures. Both Orion (Heer and Perer, 2014) and Ploceus (Liu et al., 2014) introduce a visual query interface to derive various types of networks from tabular data; subdividing the network covers aspects of multilayer networks. While Orion focusses on diverse subdivision methods and recommending relationship types, Ploceus puts more emphasis on a visual network schema editor. In contrast to using a tabular basis, Cuenca et al. (2018) present a solution for visually querying of multilayer graphs and exploring the results. Although these approaches already support the interactive definition or querying of networks, they still lack that a multilayer network is interactively created, with sophisticated options to contrast different layers. Also, using existing layers to derive and combine new layers is not yet fully leveraged.

As it is difficult, in general, to display a multilayer graph of non-trivial size in a single view across all layers, we suggest using multiple views. In our mental model, imagining a multilayer graph as a vertical stack of planes where each plane represents a layer, we can describe the different views as cuts and aggregations of this stack. We discern the following types of views:

- Intra-layer view(s): A horizontal cut through the stack, i.e., a single layer of the graph.
- Inter-layer view(s): A vertical cut through the stack, i.e., a view that focuses on the edges that connect different layers rather the ones within the layers.
- Conceptual view: An overview of the stack, i.e., a representation of layers and their characteristics.

The intra- and inter-layer views can be borrowed from existing graph visualization research; especially, approaches for multilayer graph visualization as surveyed by McGee et al. (2019) might be used as a basis. However, the conceptual view introduces a novel aspect and forms one cornerstone of an interactive layer creation and management. Instead of directly showing the data, the conceptual view that we propose would abstract from the data and would visually represent the query that was used to define the layer. Statistics on the resulting network layer can augment this representation to provide a preview and make the layers easier to compare.

In addition to the conceptual view, interactions to create and combine the layers will be a second cornerstone of the approach. It is desirable to use direct manipulation as a natural way to interact with the visually represented objects directly (i.e., with the layers, nodes, edges, etc.). Dragging a layer and dropping it on another, for instance, may trigger a combination of the two layers. However, as discussed above, options for combining layers are versatile; hence, different drag-and-drop modes might need to be available. Projections specifically might also be described as a path through the layers. A path from layer A to layer B back to layer A can indicate that a new layer C should be formed that contains the nodes of layer A; two nodes are now connected if there exists an intra-layer path between the nodes from layer A through layer B. Some interactive relationship creation and projection methods are already contained in Orion and Ploceus. Other approaches that support interactive layer manipulation are also surveyed by McGee et al. (2019, Table 2, "C – Layer manip."). These approaches can be extended to integrate with the suggested conceptual view.

In conclusion, the challenges of designing a visualization system to support interactive layer creation and analysis comprise both the design of the conceptual view as well as corresponding interactions. Since the approach shall be used by domain experts (who are often not trained specifically in computer science and data analysis), the interactive layer creation should easy to apply – a general challenge will be to find a good balance between analytical power and ease of use.

References

1 Erick Cuenca, Arnaud Sallaberry, Dino Ienco, Pascal Poncelet (2018). Visual querying of large multilayer graphs. In Proceedings of the 30th International Conference on Scientific and Statistical Database Management (p. 32). ACM.
2 Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, Bruno Pinaud (2019). The State of the Art in Multilayer Network Visualization. Computer Graphics Forum, DOI 10.1111/cgf.13610
3 Zhicheng Liu, Shamkant B. Navathe, John T. Stasko (2014). Ploceus: Modeling, visualizing, and analyzing tabular data as networks. Information Visualization, 13(1), 59–89.
4 Jeffrey Heer and Adam Perer (2014). Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. Information Visualization, 13(2), 111–133.

4.3 Unifying the framework of Multi-Layer Network and Visual Analytics

Søren Knudsen (University of Calgary & University of Copenhagen, DK), Jan Aerts (Hasselt University – Diepenbeek, BE), Daniel Archambault (Swansea University, GB), Remco Chang (Tufts University – Medford, US), Jean-Daniel Fekete (INRIA Saclay – Orsay, FR), Martin Grandjean (Université de Lausanne, CH), Jessie Kennedy (Edinburgh Napier University, GB), Mikko Kivelä (Aalto University, FI), Matteo Magnani (Uppsala University, SE), Helen C. Purchase (University of Glasgow, GB), Christian Tominski (Universität Rostock, DE), Paola Valdivia (INRIA Saclay – Orsay, FR), and Tatiana von Landesberger (TU Darmstadt, DE)

The notion of multi-layer networks introduces a general framework and common vocabulary for existing ideas in complex network theory [4]. In doing so, it is possible to understand and compare these different ideas in a new and more fruitful manner. However, to make this operationalizable to the visualization and visual analytics community, we need more clarity. For example: What is a layer? What are the semantics of interlayer edges, and specifically, identity links between layers? Can different multilayered networks be expressed or implemented in the same way? And vice versa, can one multilayered network be expressed or implemented in different ways?

It seems it is difficult to agree on a unifying framework for Visual Analytics (VA) and multilayer networks (MLN). The complex notation and diverging conceptualizations of multilayer networks are hard to unify when starting with complex constructs, and they are not easily transferred to a VA domain. The existing framework are very powerful and general, but at the same time there are several aspects that make it difficult to utilize the models from a VA perspective. If we, as computer scientists, cannot agree on how to understand this framework, how can we enter into collaborations with domain experts that we are often involving as collaborators? While considering their use of MLNs might provide new and fruitful insights, it is necessary to agree on the fundamental aspects of the framework, and how to use it for VA. Further, in doing so, these collaborations might facilitate appropriation of the model by researchers with concrete analysis needs. Finally, not agreeing on the foundations makes it difficult to build systems and tools. If we do not understand the foundations, how might we implement software on top of it?

4.3.1 How might we consider MLN from VA?

We think it is possible to respond to the issues from different perspectives, which might in different ways help us approach a general answer to the question "how might we approach MLN for VA?" We offer three such perspectives in the following.

4.3.1.1 Faceting perspective

The faceting perspective very concretely considers the types of visualizations we can create by gradually introducing more and more complexity of MLN. Doing so, we can identify "facets" – visualization techniques – that characterize multilayer networks (see Figure 1).

Facets that might be considered include, those related to nodes of a multilayer network:

- No layers (baseline)
- Existence of layers plus non-layer nodes
- Overlap-free layers
- Sequenced layers
- Overlapping layers
- Nested (or hierarchical) layers

Facets related to edges of a multilayer network:

- No layers (baseline)
- Intra-layer edges
- Inter-layer edges
- Overlapping edges

These offer a starting point for discussion but are not the only possibilities. Further aspects might be considered, such as attributes (categorical or quantitative) at nodes and edges.

The illustrations in Figure 1 are deliberately kept simple. Obviously, dealing with more than two layers is more complicated than the simple depictions might suggests. For multiple overlapping layers there is a clear connection to Euler Diagrams. Yet, for understanding and communicating the framework, we think it is beneficial to provide simple illustrations.

The faceting perspective suggests dividing the complex problem of defining a unifying framework into simple conceptual units. One might describe them as a divide-and-conquer approach to the problem. The simple conceptual units may help in constructing a comprehensive model in a step-wise and modular fashion. Depending on which facets are present in a multilayer network, different complex underlying data structures, visual representations, and interaction facilities might be required. This allows for scalability in terms of the complexity of the domain problem.

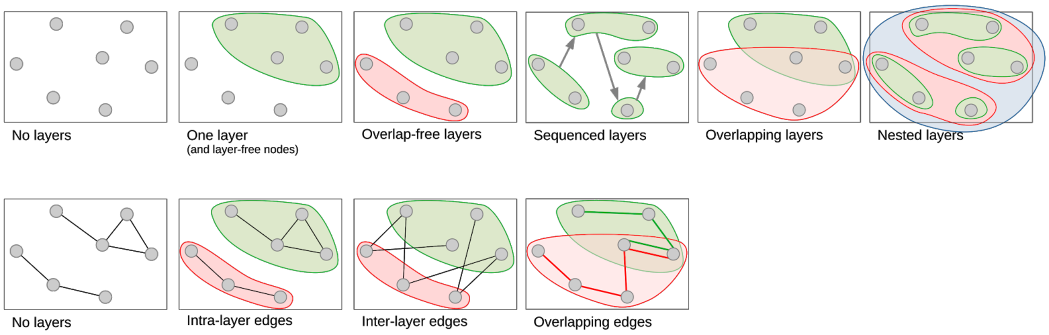


Figure 2 Faceting perspective. In this, we consider the types of visualizations we can create by gradually introducing more and more complexity of MLN.

4.3.1.2 Application perspective

The application perspective considers VA from the application domain, i.e. what MLNs are used for. In this perspective, we think creating and using visual representations of the formal model itself to convey the idea of MLN to domain experts might be fruitful. Such visual representation might articulate a multiplicity of levels that clarify the different possible networks and facilitate the appropriation of the model by researchers. The use of "levels" supports how social scientists (and many other disciplines) think about relationships and data modeling (e.g., [3]). Figure 2 shows one example of such a representation, where each level represents one entity type and relationships within. It can have many layers organized by "aspects" and include edges between these layers. For example, a level consisting of "users" can have n x m layers where n is the number of social network platforms (twitter, FB, etc.), and m is the number of mechanisms for posting (e.g. phone, computer, etc.). In this case, it is possible to represent a tweet sent from a phone and received on a computer.

The purpose of this perspective is not necessarily to be presented as a whole to the final user but can be used to question the research hypothesis and data: "do you have a two-mode graph in your data?" "Do you have multiple relationships between same type nodes?" – "So here are different ways to visualize and analyze them", that can be articulated, or viewed from this or this angle (the "scenarios").

4.3.1.3 Systems perspective

The systems perspective consider how MLNs for VA are implemented, how they are stored, and how they are queried. We recognize that MLN reminds of OLAP structures. For example, Kivelä et al. [4] p. 209 discuss their terminology, saying that they use aspect and not dimension to "avoid terminology clash". In OLAP structures, MLN might be realised by reserving one of the axes specifically for "entity type".

While the framework of MLN is not a cube, the cubic and well aligned appearance is a convenient way of representing things to make them understandable and less abstract so that they can be easily applied. However, in the model of Kivelä et al., layers are less strictly organised. By considering a unification of MLN based on OLAP cubes, we think we might introduce slightly more organisation so that the operations on the data are facilitated and visually explicit. Realising that the cubic form of OLAP cubes suggest that there are only three dimensions available, we stress that MLNs might have many more aspects than that. However, in the examples that we have observed, 3 aspect-networks are common.

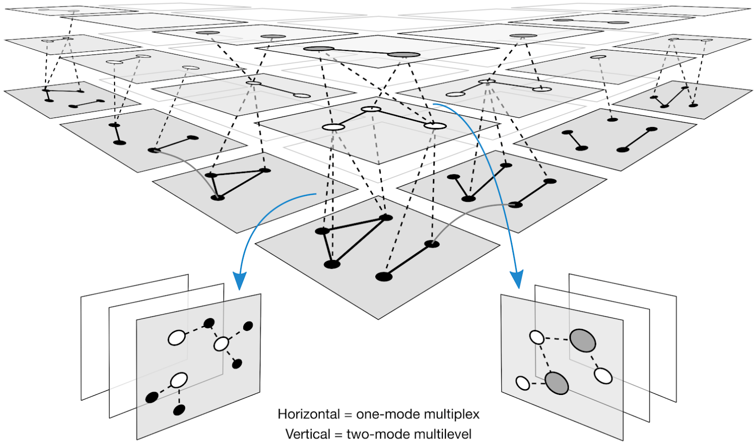


Figure 3 Figure 2: Application perspective. In collaborating with domain experts, using visual representations of the formal model might be beneficial. This figure provides just one example of a visual representation of the formal model itself, which might be used in collaborating with domain experts to establish a common grounding of what we might consider MLNs. In this representation, the multilayer network system is made of three 1-mode and two 2-mode networks organized on three interconnected levels. At each level, this representation explicitly shows the possibility of developing the graph depending on two (or more) aspects, which is also the case for the 2-mode network connecting the levels (vertical layers outside the aligned layers).

The inconvenience of the concept of the OLAP cube is that not all levels will have the same number of dimensions, and these dimensions will not necessarily be aligned. This recalls that this is only an intellectual framework and not a grid where all the layers will be filled and analyzed at the same time.

As a side note, space-time cubes [1] might be a useful starting point for considering a unification of OLAP and MLN.

4.3.2 Discussion

We presented three different perspectives on MLN and VA. While these have differences, they have important and relevant similarities. For example, the faceting perspective can also be used for collaborating between VA experts and domain experts. The facet list can serve as a checklist for inquiring whether a domain problem requires a certain characteristic. This might be more approachable than working directly with the visual representation of the MLN model discussed in the application perspective and might be a more direct way of discussing with domain experts what kinds of complex multilayer networks are required for the problem at hand.

Agreeing on an understanding of the MLN framework and how it might be used, the VA community might stand to create more fruitful visualization techniques for MLN, improve our understanding and appropriation of MLN concepts for different application domains, and allow us to build systems that, not implement MLN concepts, but allow interoperability between different tools and systems in this area. While existing visualization tools and techniques (e.g., [5], [2], [6]) have almost literately provided visualization designs that show MLN, a more structured approach based on a framework can bring more clarity to the table.

4.3.3 Conclusion

We have outlined challenges in unifying the framework of multi-layer network and visual analytics. In discussing these challenges, we offered three perspectives (faceting, application, and systems perspectives). The faceting perspective discusses gradually introducing more and more complexity of MLN based on visualization techniques. The application perspective discusses how we might collaborate on MLN projects with domain experts. The systems perspective discusses how we might start to think about implementing these concepts in concrete systems and tools based on OLAP cubes, and how these might be interoperable. Having these three conceptual perspectives allow us to reason about the concept of multilayer network and visualization from different perspectives. While we think these are useful starting points, other perspectives might also be fruitful.

References

1 Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, Sheelagh Carpendale (2017). A Descriptive Framework for Temporal Data Visualizations Based on Generalized Space-Time Cubes." Computer Graphics Forum 36, no. 6 (September 1, 2017): 36–61.
2 Christopher Collins and Sheelagh Carpendale (2007). VisLink: Revealing Relationships Amongst Visualizations. IEEE Transactions on Visualization and Computer Graphics 13, no. 6 (November 2007): 1192–99.
3 Martin Grandjean (2017). Multimode and Multilevel: Vertical Dimension in Historical and Literary Networks, Digital Humanities 2017. Montreal QC, Canada, Aug 2017.
4 Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter (2014). Multilayer Networks, Journal of Complex Networks, vol 2, issue 3, pp. 203–271
5 Ben Shneiderman and Aleks Aris (2006). Network Visualization by Semantic Substrates. IEEE TVCG 12, no. 5 (2006): 733–740.
6 Marc Streit, Michael Kalkusch, Karl Kashofer, Dieter Schmalstieg (2008). Navigation and Exploration of Interconnected Pathways. Computer Graphics Forum 27, no. 3 (May 2008): 951–58.

4.4 Analytics, Communities Comparison and attributes

Bruno Pinaud (University of Bordeaux, FR), Ariful Azad (Indiana University – Bloomington, US), Alberto Cottica (Edgeryders – Brüssel, BE), Maria Malek (EISTI – Cergy Pontoise, FR), Mark McCann (University of Glasgow, GB), Fintan McGee (Luxembourg Inst. of Science & Technology, LU), Guy Melançon (University of Bordeaux, FR), Anders Kristian Munk (Aalborg University, DK), and Blaž Škrlj (Jozef Stefan Institute – Ljubljana, SI)

The task taxonomy presented as part of McGee *et al.*'s state of the art report on multilayer network visualization [1], highlights the multilayer specific tasks related to interaction between layers which are themselves important entities in addition to the nodes and edges. Part of understanding the relationship between layers is understanding the relationships between the nodes of the layers. This concept has been explored in the context of bipartite graphs by Latapy *et al.*, with the notion of node redundancy [2]. Kivelä *et al.* [3] also refer to a range of metrics for relating layers.

A natural question that arises from the treatment of multilayer networks is how much layers interact. We propose a general measure of between-layers interaction that lends itself to comparing different real-life multilayer networks. We base it on Shannon entropy and call it multilayer entropy or ML. It is based on a null model of random-uniform distributions of inter-layer links from one node in one level to all other nodes in the other level, corresponding to maximum entropy. Low levels of ML correspond to high preferentiality in linking from that node to only some (or even only one) nodes in the other layer considered. Multilayer entropy across the layers of a multiplex network is always 0.

Let us be more precise. We are looking for:

- One or more *quantitative* measures.
- That would apply to *multilayer* networks.
- With an *intuitive interpretation.* A $[0,1]$ normalized measure would be preferred.
- And are *general*: they apply to both multiplex networks and networks of networks.

Our final goal is to apply this hypothetical measure to several different multilayer datasets. This gives us a comparative dimension, leading to statements like "this measure is close to 0 when the layers represent hierarchies", or "this measure is exactly 1 when the layers are randomly connected", and so on.

The perspective of our our working group discussion is along the lines of "Real world multilayer networks are (not) X", where X is some property. This would be the multilayer equivalent of Barabási's claim that scale free networks abound in real life, or, conversely, Braido and Clauset's that they are rare. A full paper elaborating on this type of claim requires diverse data from different disciplines for validation, which was the case of the seminar.

References

1 Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, Bruno Pinaud (2019). The State of the Art in Multilayer Network Visualization. Computer Graphics Forum, DOI 10.1111/cgf.13610

2 Matthieu Latapy, Clémence Magnien, Nathalie Del Vecchio (2008), Basic notions for the analysis of large two-mode networks, *Social Networks*, vol. 30, no. 1, pp. 31–48, 2008.

3 Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, Mason A. Porter (2014). Multilayer Networks, Journal of Complex Networks, vol 2, issue 3, pp. 203–271

4.5 Visual Encodings

Björn Zimmer (Univ. of Applied Sciences – Hagenberg, AT), Andreas Kerren (Linnaeus University – Växjö, SE), Stephen G. Kobourov (University of Arizona – Tucson, US), Benjamin Renoust (Osaka University, JP), Arnaud Sallaberry (University of Montpellier, FR), and Michael Wybrow (Monash University – Caulfield, AU)

There are several ways to visualize Multilayered Networks [1]. In an interactive analysis, however, not only the final visualization plays an important role, but also the ability to quickly create and link new layers depending on the filtering or relevance of the underlying data. It is also important to keep an overview of the specific properties and important parameters of the individual layers. Another important question is how analysts explore multilayered networks. Either via a bottom-up approach based on individual interest (e.g. a specific node in one layer) or a top-down approach (e.g. similarity between layers, identifying clusters across layers) could be facilitated. A top-down approach would help users to get a high-level (holistic) view of the layered networks. But it is unclear how to achieve such an informative high-level view. There does not seem to be any real approach to providing an overview for multilayered networks. Hiveplots or parallel cooridnates, for instance, cannot show all the data for large inputs and for smaller ones they still do not give a high-level view of the structure. The working group discussed the following questions:

- How is the data aggregated?
- Is there a possibility to use t-SNE for multilayered data?
- Are there limits of specific visual encodings?
- How to design visualizations to give users an overview of layers and their interconnections?

Goal of this working group is to define a design space of possibilities to draw multi-layer approaches. How could we facilitate the reduction of information to show the important measurements of layers and their connections? A possible way would be to visualize layers as a supergraph/layergraph in order to show possibile path's to interactively aggregate and visualize data and connections. This is done to assist users in getting an overview of their data and provide a way to interactively create individual network views based on selections of nodes and edges in the supergraph. The supergraph could show meta-information about the underlying graphs and interconnections, such as the type of graph (unconnected, tree, directed/undirected graph, completely connected graph, etc.). The details of the visualization and what kind of interactions are used exactly have to be discussed in detail.

References

1 Fintan McGee, Mohammad Ghoniem, Guy Melançon, Benoît Otjacques, Bruno Pinaud (2019). The State of the Art in Multilayer Network Visualization. Computer Graphics Forum, DOI 10.1111/cgf.13610

Participants

- Jan Aerts
Hasselt University – Diepenbeek, BE
- Daniel Archambault
Swansea University, GB
- Ariful Azad
Indiana University – Bloomington, US
- Kathrin Ballweg
TU Darmstadt, DE
- Fabian Beck
Universität Duisburg – Essen, DE
- Remco Chang
Tufts University – Medford, US
- Alberto Cottica
Edgeryders – Brüssel, BE
- Marten Düring
University of Luxembourg, LU
- Jean-Daniel Fekete
INRIA Saclay – Orsay, FR
- Mohammad Ghoniem
Luxembourg Inst. of Science & Technology, LU
- Martin Grandjean
Université de Lausanne, CH
- Jessie Kennedy
Edinburgh Napier University, GB
- Andreas Kerren
Linnaeus University – Växjö, SE
- Mikko Kivelä
Aalto University, FI
- Søren Knudsen
University of Calgary & University of Copenhagen, DK
- Stephen G. Kobourov
University of Arizona – Tucson, US
- Sylvain Legay
Luxembourg Inst. of Science & Technology, LU
- Matteo Magnani
Uppsala University, SE
- Maria Malek
EISTI – Cergy Pontoise, FR
- Mark McCann
University of Glasgow, GB
- Fintan McGee
Luxembourg Inst. of Science & Technology, LU
- Guy Melançon
University of Bordeaux, FR
- Anders Kristian Munk
Aalborg University, DK
- Bruno Pinaud
University of Bordeaux, FR
- Margit Pohl
TU Wien, AT
- Helen C. Purchase
University of Glasgow, GB
- Benjamin Renoust
Osaka University, JP
- Arnaud Sallaberry
University of Montpellier, FR
- Christian Tominski
Universität Rostock, DE
- Paola Valdivia
INRIA Saclay – Orsay, FR
- Jason Vallet
University of Bordeaux, FR
- Tatiana von Landesberger
TU Darmstadt, DE
- Michael Wybrow
Monash University – Caulfield, AU
- Björn Zimmer
Univ. of Applied Sciences – Hagenberg, AT
- Blaž Škrlj
Jozef Stefan Institute – Ljubljana, SI

Report from Dagstuhl Seminar 19062

Bringing CP, SAT and SMT together: Next Challenges in Constraint Solving

Edited by

Sébastien Bardin[1], Nikolaj S. Bjørner[2], and Cristian Cadar[3]

1 **CEA LIST, FR,** sebastien.bardin@cea.fr
2 **Microsoft Research – Redmond, US,** nbjorner@microsoft.com
3 **Imperial College London, GB,** c.cadar@imperial.ac.uk

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19062 *"Bringing CP, SAT and SMT together: Next Challenges in Constraint Solving"*, whose main goals were to bring together leading researchers in the different subfields of automated reasoning and constraint solving, foster greater communication between these communities and exchange ideas about new research directions.

Constraint solving is at the heart of several key technologies, including program analysis, testing, formal methods, compilers, security analysis, optimization, and AI. During the last two decades, constraint solving has been highly successful and transformative: on the one hand, SAT/SMT solvers have seen a significant performance improvement with a concomitant impact on software engineering, formal methods and security; on the other hand, CP solvers have also seen a dramatic performance improvement, with deep impact in AI and optimization. These successes bring new applications together with new challenges, not yet met by any current technology.

The seminar brought together researchers from SAT, SMT and CP along with application researchers in order to foster cross-fertilization of ideas, deepen interactions, identify the best ways to serve the application fields and in turn help improve the solvers for specific domains.

Seminar February 3–6, 2019 – http://www.dagstuhl.de/19062

2012 ACM Subject Classification Theory of computation → Logic, Theory of computation → Automated reasoning, Mathematics of computing → Solvers, Theory of computation → Constraint and logic programming, Hardware → Theorem proving and SAT solving, Software and its engineering → Formal methods, Software and its engineering → Software verification, Hardware → Functional verification

Keywords and phrases Automated Decision Procedures, Constraint Programming, SAT, SMT

Digital Object Identifier 10.4230/DagRep.9.2.27

1 Executive Summary

Sébastien Bardin (CEA LIST, FR)
Nikolaj S. Bjørner (Microsoft Research – Redmond, US)
Cristian Cadar (Imperial College London, GB)
Vijay Ganesh (University of Waterloo, CA)

The scattered landscape of constraint solving. Constraint solving is at the heart of several key technologies, including program analysis, testing, formal methods, compilers, security analysis, optimization, and AI. During the last two decades, constraint solving has been highly successful and transformative: on the one hand, SAT/SMT solvers have seen a significant performance improvement with a concomitant impact on software engineering, formal methods and security; on the other hand, CP solvers have also seen a dramatic performance improvement, with deep impact in AI and optimization.

These successes bring new applications together and new challenges: some fundamental constraints still lack efficient reasoning (e.g., floating-point arithmetic); quantifiers are rarely taken into account; current approaches focus essentially on satisfiability and/or validity while some applications would benefit from queries such as optimization or model counting. While each of the SAT, SMT and CP communities has made progress on some of these problems, no approach is able to tackle them all. Moreover, while historically strongly connected, the SAT/SMT communities have had minimal interactions with the CP community over the recent years.

Goals. The aim of this seminar was to reunify the Constraint Solving landscape and identify the next big challenges together with promising approaches. The seminar brought together researchers from SAT, SMT and CP along with applications researchers in order to foster cross-fertilization of ideas, deepen interactions, identify the best ways to serve the application fields and in turn help improve the solvers for specific usages.

An overview of constraint solving.

- **CP.** *Constraint Programming* [1] focuses on finding a solution (satisfiability) or a best solution (optimization) to constraint problems seen as sets of atomic constraints over arbitrary domains. Traditionally, CP is interested in problems defined over finite-domain variables (typically: bounded integers), yet a lot of work has also been devoted to infinite domains such as real numbers. The basic scheme of CP approaches (in the finite setting) consists in exploring the search tree of all partial valuations of the problem until a solution is found, or all possible valuations have been explored. At each step, *propagation* allows to refine further the admissible values for yet-unlabeled variables and, once no more propagation is possible, *labeling* assigns a value to a yet-unlabeled variable (yielding a backtrack point) and then propagation takes place against this, etc. CP has been highly successful in AI-related domains such as planning or scheduling, and promising applications to program verification have emerged recently.

  Strong points: advance propagation techniques based on the key notion of arc-consistency; specific reasoning, especially for finite-domain theories (e.g. floats, bounded arithmetic, bitvectors); queries beyond satisfiability, e.g. optimization

- **SAT.** While the seminal DPLL procedure [3] follows mostly the procedure described above for CP but specialized to the Boolean case[1], the true miracle of SAT comes from its modern version [2], where conflict-driven learning allows significant driven-by-need pruning of the search space—making the technique equally good at finding solutions or proving there is none. Many more improvements have been explored over the years, with carefully tuned propagation, data structures and branching heuristics. DPLL-style SAT solvers are at the core of hardware design and verification tools, and they have shown unreasonable efficiency on very large industrial problems.

  Strong points: conflict-driven clause learning methods; efficient search/propagate procedure, with optimized branching and look-ahead.

[1] Seeing CP as a generalization of SAT is also possible.

- **SMT.** *Satisfiability Modulo Theory* [4] extends SAT by considering the satisfiability problem over combinations of first-order theories, for examples formulas involving complex boolean structure plus uninterpreted functions, arrays and linear arithmetics. While first restricted to the unquantified case, the technique has been extended with partial support for quantifiers. The core of SMT techniques is the combination of efficient theory-dedicated conjunctive-only decision procedures (typically through the Nelson-Oppen combination framework) together with their lifting to the general (disjunctive) case thanks to the DPLL(T) framework, where a DPLL-style SAT solver works in interplay with theory solvers. SMT problems arise naturally in software analysis, where programs are built over combinations of basic data types. Hence, SMT solvers are naturally at the heart of most modern software verification technologies.

  Strong points: first-order decision procedures, including theories over infinite domains; elegant combinations of solvers; partial handling of quantifiers.

Research questions. The seminar allows to highlight several key challenges to current constraint solving techniques. They have been discussed during the meeting from different research perspectives.

- Hard-to-handle data types: several common data types and associated theories are still not managed in an efficient-enough way, typically finite-but-large domains such as modular arithmetic, bounded arithmetic with non-linear operations, floating-point arithmetic or bitvector constraints deeply mixing arithmetic and bit-level reasoning, sets with cardinality, strings with size, etc.
- Quantifiers: quantifiers can be added to SMT solvers but often at the price of losing model generation, while there is some support for finite quantification in SAT and CP but at the price of a significant drop in performance; yet, quantifiers are useful in practice (initial state, pre/post-conditions, summaries, etc.);
- Beyond satisfiability: while the first applications of constraint solving were concerned with finding solutions or proving validity / infeasibility, new applications bring new types of queries, such a optimization, soft constraints, solution counting, over-approximating sets of solutions, etc.
- New trade-offs between learning and propagation: while the SAT community seems to have reached a sweet spot on this question (with efforts put on a posteriori learning rather than on a priori propagation), the issue is not settled yet for SMT and CP, and may be theory and/or application dependent.

Potential synergies. We have also identified the following potential synergies between CP, SAT and SMT, and expect strong interactions around these points in a near future:

- CP researchers have advanced propagation techniques, domain-dedicated reasoning and (deep) constraint combination. SAT and SMT researchers can learn from that.
- SAT researchers have significantly advanced branching heuristics, look-ahead and conflict-clause learning methods. CP and SMT researchers can learn from that.
- SMT researchers have focused on theory solvers and well-defined solver combinations. How can we do "lightweight" theory integration in SAT/CP solvers that trade off generality for cheaper and focused implementation of theories aimed at very specific applications? SAT and CP researchers can take advantage of these points.
- How can we better serve the needs of applications researchers? Application researchers can tell solver designers about which of these features (and combinations thereof) they would like the most in a single solver.
- Finally, an important question is how do we leverage machine learning in these contexts. The experience of the SAT community may bring here some answers.

Outcome. The main goal of this Dagstuhl seminar was to bring together leading researchers in the different subfields of automated reasoning and constraint solving, foster greater communication between these communities and discuss new research directions.

The seminar had 28 participants from Australia, Austria, France, Germany, Finland, Italy, Spain, Sweden, Switzerland, United Kingdom and United States, from both academia, research laboratories and the industry. More importantly, the participants represented several different communities, with the topics of the talks and discussions reflecting these diverse interests in both solving technologies (CP, SAT, SMT), challenges (floating-point constraints, quantifiers, etc.) and application domains (testing, verification, security, compilation, commercialization, among others).

It was the first time such an *inclusive* meeting was held, bringing together leading researchers from SAT/SMT (typical interest: formal verification), CP (typical interest: optimization) and applications (typical interest: testing, verification, security). All participants agreed the event was fruitful, and we expect to see more collaborations between SAT/SMT and CP in a near future.

References

1. R. Dechter. *Constraint processing.* Elsevier Morgan Kaufmann, 2003.
2. L. Zhang, C. F. Madigan, M. H. Moskewicz, and S. Malik. Efficient conflict driven learning in a boolean satisfiability solver. In *International Conference on Computer-aided design.* IEEE Press, 2001.
3. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Communications of the ACM*, 1962.
4. C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability.* 2009.

2 Table of Contents

3 Tutorials

3.1 The State-of-the-Art in SAT Solving: Search, Simplify, Prove!

Armin Biere (Johannes Kepler Universität Linz, AT)

This tutorial covers algorithmic aspects of conflict-driven clause learning and important extensions developed in recent years, including decision heuristics, restart schemes, various pre- and inprocessing techniques as well as proof generation and checking. Programmatic incremental usage of SAT solvers is discussed next. The talk closes with an overview on the state-of-the-art in parallel SAT solving and open challenges in general.

The speaker contributed to the core technology of modern SAT solving, has developed 12 SAT solvers since 1999, which won 36 medals including 16 gold medals in international SAT competitions.

3.2 SMT – Basics and Recent Trends

Nikolaj S. Bjørner (Microsoft Research – Redmond, US)

The tutorial provides a refresher on Satisfiability Modulo Theories, SMT [1, 2]. SMT is used for a branch of automatic theorem proving that integrates satisfiability search with specialized procedures for theories of relevance. We describe the basic architecture of SMT solvers as a combination of propositional SAT search with integrated theory reasoning. The theories described in the tutorial includes arithmetic reasoning, floating point reasoning, strings, and domains found in CP solvers. The tutorial highlights recent advances in arithmetic reasoning, such as reducing non-linear integer arithmetic solving to linear integer arithmetic by introducing tangent lemmas and other properties of linear arithmetic formulas. The approach complements a technique based on reductions to cylindric algebraic decomposition. A number of other recent advances around reducing integer feasibility using strengthened linear real constraints, and centering search around policy iteration are also discussed. The second part of the tutorial describes uses of SMT solving. First from the view of which functionality is available in SMT solvers as a service to applications, second from the perspective of a set of timely applications of SMT solvers, including network verification, quantum compilation, smart contract verification, trusted financial software, DNN analysis, axiomatic economics, and uses of strings and regular expression reasoning.

References

1 C. W. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. In *Handbook of Satisfiability*. 2009.
2 Leonardo de Moura and Nikolaj Bjørner. Satisfiability Modulo Theories: Introduction and Applications. CACM, September 2011

3.3 The State of the Art in CP Solving: Infer, Relax, Search!

Pierre Flener (Uppsala University, SE)

I explain the declarative structure-based high-level modelling and the composition of specialised algorithms within the satisfaction and optimisation solvers of constraint programming (CP) technology, whether they work by systematic search or local search or a hybrid thereof, aided by a lot of inference and, to a lesser extent, by relaxation. If desired, the solving process can be parametrised by a user-provided search strategy, either by choice among predefined ones or programmatically. A lot of important extensions have been developed in recent years, including preprocessing, symmetry handling, autonomous search, hybridisation with SAT, etc.

3.4 Learning in Constraint Programming

Peter J. Stuckey (The University of Melbourne, AU)

Joint work of Peter J. Stuckey, Olga Ohrimenko, Michael Codish, Thibaut Feydy, Geoffrey Chu, Graeme Gange
Main reference Olga Ohrimenko, Peter J. Stuckey, Michael Codish: "Propagation via lazy clause generation", Constraints, Vol. 14(3), pp. 357–391, 2009.
URL https://doi.org/10.1007/s10601-008-9064-x

Nogood learning is a powerful mechanism for improving combinatorial search, by remembering what went wrong in the past, we can avoid making the same mistakes in the future. Nogood learning originated in the CP community, but had its first major impact in the SAT community where nogood learning SAT solvers are now universal. Modern CP solvers make use of the same learning mechanisms as in SAT, but there are many different features that arise in making them effective. In this talk I describe the nitty-gritty details about how CP solvers use nogood learning to improve performance. We discuss atomic constraints versus Booleans, integer variable (*theory*) propagators, lazy literal generation, structure based extended resolution, lifting explanations, and theory propagators in CP.

References

1 Thibaut Feydy, Peter J. Stuckey: Lazy Clause Generation Reengineered. CP 2009. Springer, 2009
2 Andreas Schutt, Thibaut Feydy, Peter J. Stuckey, Mark G. Wallace: Explaining the cumulative propagator. Constraints 16(3): 250-282 (2011)

4 Overview of Talks

4.1 Redundancy in Clausal Proofs and Satisfaction Driven Clause Learning

Armin Biere (Johannes Kepler Universität Linz, AT)

Joint work of Marijn J. H. Heule, Benjamin Kiesl, Armin Biere

We discuss recent notions of redundancy in clausal propositional proof system, including variants of blocked clauses, resolution asymmetric tautologies (RAT), as well as our new redundancy notion of propagation redundancy (PR). These concepts can be used to obtain short clausal proofs of hard combinatorial problems. We further proposed a new SAT solving paradigm, called satisfaction driven clause learning (SDCL), which can generate such proofs automatically.

References

1 Marijn J. H. Heule, Benjamin Kiesl, Armin Biere: Short Proofs Without New Variables. CADE 2017
2 Marijn J. H. Heule, Benjamin Kiesl, Martina Seidl, Armin Biere: PRuning Through Satisfaction. Haifa Verification Conference 2017

4.2 Bigly solving with Z3

Nikolaj S. Bjørner (Microsoft Research – Redmond, US)

The talk describes solving hard scheduling constraints using distributed SAT solving. Using the method of cube-and conquer, Z3 is distributed on hundreds of CPUs in Azure. We describe these solving methods that leverage cloud infrastructure.

4.3 COLIBRI: CP for FP (and BV)

François Bobot (CEA LIST, FR) and Sébastien Bardin (CEA LIST, FR)

Joint work of Sébastien Bardin, François Bobot, Zakaria Chihani, Bruno Marre

At first sight people misunderstand floating point numbers as reals. Then with more experiences their counter-intuitive behavior are patent. However sometimes they still behave like reals. We are going to see how a CP approach is able to prove these kind of assertions. We are going to see also how bitvectors are handled in a CP way.

We presented an efficient Constraint Programming approach to the SMTLIB theory of quantifier-free floating-point arithmetic (QF-FP). We rely on dense interreduction between many domain representations to greatly reduce the search space. We compare our tool to current state-of-the-art SMT solvers and show that it is consistently better on large problems involving non-linear arithmetic operations (for which bit-blasting techniques tend to scale badly). We also briefly present results on the theory of bit-vectors (QF-BV) following the same high-level CP philosophy. These results emphasize the importance of the conservation of the high-level structure of the original problems, compared with standard bitblasting approaches.

References

1 Zakaria Chihani, Bruno Marre, François Bobot, Sébastien Bardin: Sharpening Constraint Programming Approaches for Bit-Vector Theory. CPAIOR 2017. Springer, 2017
2 François Bobot, Zakaria Chihani and Bruno Marre: Real Behavior of Floating Point. SMT Workshop 2017
3 Sébastien Bardin, Philippe Herrmann, Florian Perroud: An Alternative to SAT-based Approaches for Bit-Vectors. TACAS 2010. Springer, 2010

4.4 Replayable Symbolic Execution

Frank Busse (Imperial College London, GB)

Joint work of Frank Busse, Cristian Cadar

Symbolic execution is a dynamic program analysis technique that heavily relies on constraint solving. Constraint solving is computationally expensive, and many symbolic execution engines employ query caches to reduce solving time. Still, most of the execution time is spent solving constraints. Even worse, current engines are not able to reuse solver results and have to re-compute all results in every run and for every new version of a program under test. We present *Replayable Symbolic Execution*, a lightweight technique that persistently stores and re-uses solver results to significantly reduce the solving time in subsequent executions. Additionally, we give an overview of the distribution of solving times for individual queries in symbolic execution runs on real-world software. Symbolic execution often generates queries that are challenging for modern solvers, but the majority of queries is easily solvable. However, this majority accumulates to a substantial amount of solving time, and we argue that developers of SMT solvers should also consider this use case and optimise for short start-up phases.

4.5 Just Fuzz-it: An Unconventional Approach to SMT Solving

Cristian Cadar (Imperial College London, GB)

Joint work of Daniel Liew, Alaistair Donaldson, Cristian Cadar, J. Ryan Stinnett

In this ongoing work, we investigate the use of coverage-guided fuzzing as a means of proving satisfiability of SMT formulas over finite variable domains, with specific application to floating-point constraints. We show how an SMT formula can be encoded as a program containing a location that is reachable if and only if the program's input corresponds to a satisfying assignment to the formula. A coverage-guided fuzzer can then be used to search for such an assignment via a test input that covers the location. We have implemented this idea in a tool, Just Fuzz-it Solver (JFS), and we present a large experimental evaluation showing that JFS is both competitive with and complementary to state-of-the-art SMT solvers with respect to solving floating-point constraints, and that the coverage-guided approach of JFS provides significant benefit over naive fuzzing in the floating-point domain. Applied in a portfolio manner, the JFS approach thus has the potential to complement traditional SMT solvers for program analysis tasks that involve reasoning about floating-point constraints.

A publication is coming up and will be posted at https://srg.doc.ic.ac.uk/publications/ JFS is publicly available at https://github.com/delcypher/jfs/

4.6 Greybox Fuzzing with Cost-Directed Input Prediction

Maria Christakis (MPI-SWS – Kaiserslautern, DE)

Joint work of Valentin Wüstholz, Maria Christakis
Main reference Valentin Wüstholz, Maria Christakis: "Learning Inputs in Greybox Fuzzing", CoRR, Vol. abs/1807.07875, 2018.
URL https://arxiv.org/abs/1807.07875

Greybox fuzzing is a lightweight testing approach that effectively detects bugs and security vulnerabilities. However, greybox fuzzers randomly mutate program inputs to exercise new paths; this makes it challenging to cover code that is guarded by narrow checks, which are satisfied by no more than a few input values.

In this work, we present a technique that extends greybox fuzzing with a method for predicting new inputs based on costs computed along already explored program executions. The new inputs are predicted such that they guide exploration toward optimal executions, which minimize a certain cost, for instance, the cost of covering a new path or revealing a vulnerability. We have evaluated our technique and compared it to standard greybox fuzzing on real-world benchmarks. In comparison, our technique detects significantly more bugs, often orders-of-magnitude faster.

4.7 Better Bit Blasting in Yices 2

Bruno Dutertre (SRI – Menlo Park, US)

Joint work of Dejan Jovanovic, Stéphane Graham-Lengrand, Jorge A. Navas

We present recent developments in solving bit-vector problems in Yices using the standard *bit-blasting* method, which amounts to converting a bit-vector SMT problem into SAT. This is work in progress. New techniques discussed include: better use of information available at the SMT level to guide preprocessing and variable elimination in the SAT solver, and the use of cut-sweeping as a preprocessing and in-processing technique.

4.8 SMT for Binary-Level Security Analysis

Benjamin Farinier (CEA LIST, FR)

Joint work of Sébastien Bardin, Richard Bonichon, Robin David, Matthieu Lemerre, Marie-Laure Potet, Benjamin Farinier

Program verification is an undeniable success of formal methods. Driven by progress in Satisfiability Modulo Theories, it led to the development of several tools for automatic bugs search. However, these decision procedures remain unsuitable when looking for vulnerabilities. Indeed, not all the bugs are vulnerabilities, and being able to distinguish them requires the resolution of formulas of appreciably larger size, but also belonging to more expressive logics which are poorly supported by current solvers. In this presentation, I will explain why the search for vulnerabilities leads to such formulas, then I will present two of our results on their resolution.

References

1 Benjamin Farinier, Sébastien Bardin, Richard Bonichon, Marie-Laure Potet: Model Generation for Quantified Formulas: A Taint-Based Approach. CAV 2018. Springer, 2018
2 Benjamin Farinier, Robin David, Sébastien Bardin, Matthieu Lemerre: Arrays Made Simpler: An Efficient, Scalable and Thorough Preprocessing. LPAR 2018. Springer, 2018

4.9 SMT-Based Exploit Generation: Past, Present and Future

Sean Heelan (University of Oxford, GB)

SMT solvers are at the core of the most popular approaches to exploit generation. In this talk I will first briefly outline the exploit generation problem and then explain how the existing state-of-the-art leverages SMT solvers to address it. Finally, I will give an overview of an entirely new approach to exploit generation that addresses some of the most significant limitations of existing systems, while still being tightly coupled with SMT solving technology.

4.10 Symbolic Pointers

Timotej Kapus (Imperial College London, GB)

Joint work of Timotej Kapus, Cristian Cadar

Symbolic execution is an effective technique for exploring paths in a program and reasoning about all possible values on those paths. However, the technique still struggles with code that uses complex heap data structures, in which a pointer is allowed to refer to more than one memory object. In this talk I present and discuss three ways symbolic execution can handle such cases:

1. Symbolic execution forks execution into multiple states, one for each object to which the pointer could refer, this can lead to major state explosion.

2. Instead of forking the whole symbolic execution, the constraints of each potential fork can be grouped together in a big disjunction, however SMT solver often struggle more with disjunctions.
3. All the memory can be grouped together into a single objetct thus avoiding the problem, but makes the constraints too large for real programs.

4.11 Symbol Elimination and Vampire

Laura Kovács (TU Wien, AT)

Joint work of Laura Kovacs, Andrei Voronkov, Simon Robillard, Evgeny Kotelnikov, Bernhard Gleiss

We overview the symbol elimination method for using first-order theorem proving in software analysis and verification. Symbol elimination exploits consequence finding in saturation-based theorem proving and generates logical consequences of an input set S of formulas such that these consequences are using only a subset of the input symbols from S. To make symbol elimination practical and scalable for program analysis, we use symbol elimination in the first-order theories of various data structures, imposing the challenge of reasoning with both theories and quantifiers. The talk will overview recent recent developments on symbol elimination and its use within our Vampire theorem prover, and report on our experiments applying symbol elimination for generating loop invariants and proving program loops correct.

References

1 Bernhard Gleiss, Laura Kovács, Simon Robillard: Loop Analysis by Quantification over Iterations. LPAR 2018
2 Laura Kovács, Andrei Voronkov: Finding Loop Invariants for Programs over Arrays Using a Theorem Prover. FASE 2009
3 Evgenii Kotelnikov, Laura Kovács, Andrei Voronkov: A FOOLish Encoding of the Next State Relations of Imperative Programs. IJCAR 2018

4.12 Floating-point Program Verification with CP Technology

Laurent Michel (University of Connecticut – Storrs, US)

Joint work of Heytem Zitoun, Claude Michel, Michel Rueher, Laurent Michel
Main reference Heytem Zitoun, Claude Michel, Michel Rueher, Laurent Michel: "Search Strategies for Floating Point Constraint Systems", in Proc. of the Principles and Practice of Constraint Programming – 23rd International Conference, CP 2017, Melbourne, VIC, Australia, August 28 – September 1, 2017, Proceedings, Lecture Notes in Computer Science, Vol. 10416, pp. 707–722, Springer, 2017.
URL https://doi.org/10.1007/978-3-319-66158-2_45

The ability to verify critical software is a key issue in embedded and cyber physical systems typical of automotive, aeronautics or aerospace industries. Bounded model checking and constraint programming approaches search for counter-examples that exhibit property violations. The search of such counter-examples is a long, tedious and costly task, especially for programs performing floating point computations. Existing search strategies are dedicated to finite domains and, to a lesser extent, to continuous domains. In this talk, we outline how CP can be used to this end and how critical novel search strategies are to floating point constraints. Empirical results help position this work with respect to state-of-the-art SAT and SMT solvers applied to the same task.

4.13 Experiences with a Little Combinatorial-optimization Startup

Robert Nieuwenhuis (UPC – Barcelona, ES)

Experiences with a little combinatorial-optimization startup. Some of the technical and practical challenges we encountered are discussed. We also describe three of the very different real-world problems we have attacked, and discuss desirable improvements in solver technology.

4.14 On Division Versus Saturation in Cutting Planes

Jakob Nordström (KTH Royal Institute of Technology – Stockholm, SE)

Joint work of Stephan Gocht, Amir Yehudayoff, Jakob Nordström

The conflict-driven clause learning (CDCL) paradigm has revolutionized SAT solving over the last two decades. Extending this approach to pseudo-Boolean (PB) solvers doing 0-1 linear programming holds the promise of further exponential improvements in theory, but intriguingly such gains have not materialized in practice. Also intriguingly, the most popular PB extensions of CDCL have not employed the standard cutting planes method with division, but have instead used the saturation rule saying that no variable coefficient needs to be larger than the maximum contribution that the inequality can require from this variable. To the best of our knowledge, there has been no study comparing the strengths of division and saturation in PB solving.

In this work, we show that cutting planes with division can be exponentially stronger than cutting planes with saturation, even when all linear combinations of inequalities are required to cancel variables (as in PB conflict analysis). In the other direction we do not obtain an exponential separation, but we show that the number of division steps needed to simulate a single saturation step can be exponential in the bitsize of the coefficients involved. We also perform some experiments on crafted benchmarks to see to what extent these theoretical phenomena can be observed in actual solvers. Our conclusions are that a careful combination of division and saturation seems to be crucial to harness more of the power of the cutting planes method in PB solvers.

References

1. Marc Vinyals, Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, Jakob Nordström: In Between Resolution and Cutting Planes: A Study of Proof Systems for Pseudo-Boolean SAT Solving. SAT 2018
2. Jan Elffers, Jesús Giráldez-Cru, Jakob Nordström, Marc Vinyals: Using Combinatorial Benchmarks to Probe the Reasoning Power of Pseudo-Boolean Solvers. SAT 2018

4.15 Continuous Constraint Solving

Marie Pelleau (Laboratoire I3S – Sophia Antipolis, FR)

Joint work of Marie Pelleau, Antoine Miné, Charlotte Truchet, Frédéric Benhamou
Main reference Marie Pelleau, Antoine Miné, Charlotte Truchet, Frédéric Benhamou: "A Constraint Solver Based on Abstract Domains", in Proc. of the Verification, Model Checking, and Abstract Interpretation, 14th International Conference, VMCAI 2013, Rome, Italy, January 20-22, 2013. Proceedings, Lecture Notes in Computer Science, Vol. 7737, pp. 434–454, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-35873-9_26

Constraint Programming generally deals with discrete variables. In this short talk I will summarize some of the techniques used to deal with numerical problems containing continuous variables.

4.16 Reusing Solutions Modulo Theories

Mauro Pezzè (University of Lugano, CH)

Joint work of Andrea Aquino, Giovanni Denaro, Mauro Pezzè
Main reference Andrea Aquino, Giovanni Denaro, Mauro Pezzè: "Heuristically matching solution spaces of arithmetic formulas to efficiently reuse solutions", in Proc. of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017, pp. 427–437, IEEE / ACM, 2017.
URL http://dx.doi.org/10.1109/ICSE.2017.46

This talk presents an approach for reusing formula solutions for both satisfiability and unsatisfiability proofs in order to reduce the impact of Satisfiability Modulo Theories (SMT) solvers on the scalability of symbolic program analysis.

SMT solvers can efficiently handle huge expressions in relevant logic theories, but they still represent a main bottleneck to the scalability of symbolic analyses, like symbolic execution and symbolic model checking. Reusing proofs of formulas solved during former analysis sessions can reduce the amount of invocations of SMT solvers, thus mitigating the impact of SMT solvers on symbolic program analysis. Yet, early approaches to reuse formula solutions exploit equivalence and inclusion relations among structurally similar formulas, and are strongly tighten to the specific target logics.

In this talk, I present an original approach that reuses both satisfiability and unsatisfiability proofs shared among many different formulas – beyond the standard cases of equivalent or related-by-implication formulas. The approach straightforwardly generalises across multiple logics. The technique is based on the original concept of distance between formulas, which heuristically approximates the likelihood of formulas to share either satisfiability or unsatisfiability proofs.

4.17 Interpolation in SMT

Tanja Schindler (Universität Freiburg, DE)

Joint work of Jochen Hoenicke, Tanja Schindler
Main reference Jochen Hoenicke, Tanja Schindler: "Efficient Interpolation for the Theory of Arrays", in Proc. of the Automated Reasoning – 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 10900, pp. 549–565, Springer, 2018.
URL https://doi.org/10.1007/978-3-319-94205-6_36

Craig interpolants are used to derive invariants in interpolation-based model checking. The interpolants can be generated from proofs of unsatisfiability provided by an SMT solver. In the talk we discuss some approaches to produce interpolants for different first-order theories, and highlight specific techniques implemented in our interpolating SMT solver SMTInterpol that address a couple of different difficulties in proof-based interpolation.

References

1 Jochen Hoenicke, Tanja Schindler: Efficient Interpolation for the Theory of Arrays. IJCAR 2018
2 Jochen Hoenicke, Tanja Schindler: Solving and Interpolating Constant Arrays Based on Weak Equivalences. VMCAI 2019

4.18 Machine Learning Clause DB Managment

Mate Soos (Hobbyist – Berlin, DE)

Joint work of Mate Soos, Raghav Kulkarni, Kuldeep Meel

In this talk, we present a machine-learning based system for learnt clause database management. The system has three main parts. The data gathering, the data crunching and machine learning model generation, and validation. The data we have collected is many GBs of relevant, never-before seen data about SAT solver behavior. We then crunch this data through sampling and data modeling to create a machine learning model that can be executed inside the SAT solver. Finally, we run the SAT solver with the learnt model inside. The results validate the approach but more interestingly, the side-effect of having so much valuable data is something that we didn't anticipate and may well be as important as the final results themselves.

4.19 Unison 101

Christian Schulte (KTH Royal Institute of Technology – Stockholm, SE)

Joint work of Christian Schulte, Mats Carlsson, Roberto Castañeda Lozano, Gabriel Hjort Blindell
URL https://unison-code.github.io/

This talk shows how Unison improves code generation in compilers by using constraint programming (CP) as a method for solving combinatorial optimization problems. It presents how register allocation (assigning program variables to processor registers) and instruction scheduling (reordering processor instructions to increase throughput) can be modeled and solved using CP. Unison is significant as its addresses the same aspects as traditional code generation algorithms, yet is based on simple models and can robustly generate better code.

Unison is a collaboration between SICS, KTH, and Ericsson.

References

1 Roberto Castañeda Lozano, Mats Carlsson, Gabriel Hjort Blindell, Christian Schulte: Register allocation and instruction scheduling in Unison. CC 2016. Springer, 2016

2 Gabriel Hjort Blindell, Roberto Castañeda Lozano, Mats Carlsson, Christian Schulte: Modeling Universal Instruction Selection. CP 2015. Springer 2015

3 Roberto Castañeda Lozano, Mats Carlsson, Gabriel Hjort Blindell, Christian Schulte: Combinatorial Register Allocation and Instruction Scheduling. CoRR abs/1804.02452 (2018)

4.20 Solver Independent Rotating Workforce Scheduling

Peter J. Stuckey (The University of Melbourne, AU)

Joint work of Peter J. Stuckey, Andreas Schutt, Nysret Musliu
Main reference Nysret Musliu, Andreas Schutt, Peter J. Stuckey: "Solver Independent Rotating Workforce Scheduling", in Proc. of the Integration of Constraint Programming, Artificial Intelligence, and Operations Research – 15th International Conference, CPAIOR 2018, Delft, The Netherlands, June 26-29, 2018, Proceedings, Lecture Notes in Computer Science, Vol. 10848, pp. 429–445, Springer, 2018.
URL https://doi.org/10.1007/978-3-319-93031-2_31

We give two solver independent models for the rotating work-force scheduling and compare them using different solving technology, both constraint programming and mixed integer programming. We show that the best of these models outperforms the state-of-the-art complete approaches for the rotating workforce scheduling problem, and that solver independent modelling allows us to use different solvers to achieve different aims: e.g. speed to solution, robustness of solving (particular for unsatisfiable problems) and how quickly we can generate good solutions (for optimization versions of the problem).

The lessons learned from this problem are interesting. An expert modeller constructed a model targeting CP solvers and another model targeting MIP solvers, but in practice the best model for CP solvers was the MIP one, and the best solver for MIP solvers was the CP one. This shows the importance of solver independent modelling where we dont commit to the solving technology we use during the modelling process.

4.21 Counterexample-Guided Quantifier Instantiation in Logical Theories

Cesare Tinelli (University of Iowa – Iowa City, US)

Joint work of Cesare Tinelli, Andrew Reynolds, Haniel Barbosa, Clark Barrett, Pascal Fontaine, Amit Goel, Dejan Jovanovic, Sava Krstič, Leonardo de Moura, Aina Niemetz, Andres Noetzli, Mathias Preiner
Main reference Aina Niemetz, Mathias Preiner, Andrew Reynolds, Clark Barrett, Cesare Tinelli: "Solving Quantified Bit-Vectors Using Invertibility Conditions", in Proc. of the Computer Aided Verification – 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part II, Lecture Notes in Computer Science, Vol. 10982, pp. 236–255, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-319-96142-2_16

This talk provides an overview of a general approach to reason with quantified formulas in SMT. The approach maintains a set S of ground formulas incrementally expanded with selected instances of quantified input formulas, with the selection based on counter-models of S. In particular, in first-order theories that admit quantifier elimination and have a decidable universal fragment, this approach leads to practically efficient decision procedures for the full theory.

4.22 Creating a Program Schedule in EasyChair

Andrei Voronkov (University of Manchester, GB)

After giving a short overview of EasyChair and constraint satisfaction problems in it, the talk focuses on the problem of creating automatically a high-quality program schedule.

The talk formulates the problem, explains why it is hard and proposes it as a challenge to the SAT/CSP community.

5 Discussions

5.1 Food for Thought on CP and SAT/SMT

Pierre Flener (Uppsala University, SE)

I discuss existing bridges between the CP and SAT/SMT solving technologies, awareness issues, as well as differences and cross-fertilisation opportunities.

5.2 Numerical Challenges

Yannick Moy (AdaCore – Paris, FR)

SPARK is a subset of the Ada programming language targeted at formal verification. It comes with a formal verification tool called GNATprove based on the Why3 platform and relies on SMT solvers (Alt-Ergo, CVC4 and Z3) as the main engines of proof. Customers and users of SPARK face three main challenges regarding proof of numerical properties: non-linear arithmetic (in integers or bitvectors), floating-point arithmetic, combining theories. Many properties involving one of these are not provable by SMT solvers, and currently require proof in Coq of corresponding lemmas. For each of these, the challenge is both to prove true properties, and to generate counterexamples for false properties.

5.3 A SAAS Solver

Robert Nieuwenhuis (UPC – Barcelona, ES)

Shouldn't we use hundreds of machines (not cores), available at 0.01€ per machine · h, to offer a SAAS solver? Indeed, CDCL (underlying SAT, SMT, Pseudo-Boolean solving, lazy-clause-generation-based CP, etc.) is terribly sequential (perhaps even more than you think). So, why are we stubbornly trying to parallelize it with portfolios (sharing what?)? Alternative ideas could be to exploit community structure (perhaps computed by some short CDCL runs?) and parallel inprocessing. In this setting, are we married with clauses? Couldn't we handle any (more or less efficiently unit-propagatable?) representation like PB-constraints or even BDDs?

5.4 Inprocessing in SAT?

Mate Soos

This talk discusses how inprocessing has been used in modern SAT solvers. There have been a lot of papers about inprocessing, but they are rarely used. Why?

6 Programme

The seminar was mainly organized around short talks (15 min) in order to give the opportunity to each participant to present his work and to share his thoughts on the topic. The short duration was intended to keep a fast pace and good interactions. Besides, the first day featured several longer tutorials (SAT – 1h, CP – 1h, SMT – 30 min, learning in CP – 30 min) in order to bring the audience a minimal common background and to efficiently bootstrap interactions. Finally, four discussion sessions (30 min) were devoted to open discussions on hot topics, the speakers briefly introducing the challenge and their views and then the organizers leading the discussion. This schedule is only indicative: many sessions took longer because of intense and fruitful discussions.

Programme for Monday 4th of February

- Organizers – *Overview to seminar, introduction of participants, proposals*
- Armin Biere – *(tutorial) The State-of-the-Art in SAT Solving: Search, Simplify, Prove!*
- Nikolaj Bjørner – *(tutorial) Overview of SMT Solving*
- Pierre Flener – *(tutorial) The State of the Art in CP Solving: Infer, Relax, Search!*
- Peter Stuckey – *(tutorial) Learning in Constraint Programming*
- Maria Christakis – *Greybox Fuzzing with Cost-Directed Input Prediction*
- Frank Busse – *Replayable Symbolic Execution*
- Timotej Kapus – *Symbolic Pointers*
- Yannick Moy – *(discussions) Numerical challenges*
- Robert Nieuwenhuis – *Experiences with a little combinatorial-optimization startup*

Programme for Tuesday 5th of February

- Laurent Michel – *Floating-point Program Verification with CP Technology*
- François Bobot – *Real Behavior of Floating Point numbers*
- Mate Soos – *(discussion) Inprocessing in SAT – what happened?*
- Peter Stuckey – *Solver Independent Modelling for Rotating Workforce Scheduling*
- Christian Schulte – *Unison 101: Generating Code with Constraint Programming*
- Cristian Cadar – *JFS: Constraint solving via fuzzing*
- Mauro Pezzè – *Reusing Solutions Modulo Theories*
- Robert Nieuwenhuis – *(discussion) A SAAS Solver*
- Jakob Nordström – *On Division Versus Saturation in Cutting Planes*
- Laura Kovács – *Symbol Elimination and Vampire*
- Armin Biere – *Redundancy in Clausal Proofs and Satisfaction Driven Clause Learning*
- Pierre Flener – *(discussion) Food for Thought on CP and SAT/SMT*
- Benjamin Farinier – *SMT solving for security*
- Cesare Tinelli – *Quantifier Instantiation Techniques in SMT*

Programme for Wednesday 6th of February

- Mate Soos – *Supervised Machine Learning for Clause Deletion Strategies*
- Marie Pelleau – *Continuous constraint solving*
- Tanja Schindler – *Interpolation in SMT*
- Andrei Voronkov – *Creating a program schedule in EasyChair*
- Sean Heelan – *SMT-Based Exploit Generation: Past, Present and Future*
- Nikolaj Bjørner – *Bigly solving with Z3*
- Bruno Dutertre – *Better Bitblasting in Yices*

Participants

- Sébastien Bardin
CEA LIST, FR
- Armin Biere
Johannes Kepler Universität Linz, AT
- Nikolaj S. Bjorner
Microsoft Research – Redmond, US
- François Bobot
CEA LIST – Nano-INNOV, FR
- Frank Busse
Imperial College London, GB
- Cristian Cadar
Imperial College London, GB
- Maria Christakis
MPI-SWS – Kaiserslautern, DE
- Bruno Dutertre
SRI – Menlo Park, US
- Benjamin Farinier
CEA LIST – Nano-INNOV, FR
- Pierre Flener
Uppsala University, SE
- Sean Heelan
University of Oxford, GB
- Matti Järvisalo
University of Helsinki, FI
- Timotej Kapus
Imperial College London, GB
- Laura Kovács
TU Wien, AT
- Laurent Michel
University of Connecticut – Storrs, US
- Yannick Moy
AdaCore – Paris, FR
- Robert Nieuwenhuis
UPC – Barcelona, ES
- Jakob Nordström
KTH Royal Institute of Technology – Stockholm, SE
- Marie Pelleau
Laboratoire I3S – Sophia Antipolis, FR
- Mauro Pezzè
University of Lugano, CH
- Tanja Schindler
Universität Freiburg, DE
- Christian Schulte
KTH Royal Institute of Technology – Stockholm, SE
- Laurent Simon
University of Bordeaux, FR
- Mate Soos
Hobbyist – Berlin, DE
- Peter J. Stuckey
The University of Melbourne, AU
- Cesare Tinelli
University of Iowa – Iowa City, US
- Andrei Voronkov
University of Manchester, GB

Report from Dagstuhl Seminar 19071

Specification Formalisms for Modern Cyber-Physical Systems

Edited by

Jyotirmoy V. Deshmukh[1], Oded Maler[2], and Dejan Ničković[3]

1 **University of Southern California – Los Angeles, US,** `jdeshmuk@usc.edu`
2 **VERIMAG – Grenoble, FR,** `oded.maler@univ-grenoble-alpes.fr`
3 **AIT Austrian Institute of Technology, AT,** `dejan.Ničković@ait.ac.at`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19071 "Specification Formalisms for Modern Cyber-Physical Systems." Specifications play a major role in evaluating behaviors of modern cyber-physical systems (CPS). There is currently no specification language that allows joint description of safety, performance, security, privacy, and reliability aspects of CPS applications. The Dagstuhl seminar brought together researchers and practitioners from formal methods, control theory, machine learning and robotics to discuss the state-of-the-art and open challenges in specifying properties of modern CPS. Special attention was given to exploring the intersection of machine learning and formal specification languages, where formal specifications can serve as a bridge between the world of verification and the world of learning and data-mining.

Seminar February 10–15, 2019 – http://www.dagstuhl.de/19071

2012 ACM Subject Classification Theory of computation → Timed and hybrid models, Theory of computation → Modal and temporal logics, Theory of computation → Formalisms, Computer systems organization → Embedded and cyber-physical systems, Computer systems organization → Real-time system specification

Keywords and phrases Cyber-physical systems, formal specifications, runtime verification and control

Digital Object Identifier 10.4230/DagRep.9.2.48

1 Executive Summary

Jyotirmoy V. Deshmukh
Dejan Ničković

Modern Cyber-Physical Systems (CPS) represent the convergence of the fields of control theory, artificial intelligence, machine learning, and distributed communication/coordination. CPS applications range from small quad-rotor based aerial vehicles to commercial airplanes, from driverless autonomous vehicles to vehicle platoons, from nano-scale medical devices in closed-loop with a human to giga-scale industrial manufacturing systems. While several application domains can claim to be cyber-physical systems, a unique aspect of CPS is a strong focus on model-based development (MBD). The MBD paradigm allows analyzing the system virtually, examining its safety, performance, stability, security, privacy, or resilience. At a certain level of abstraction, a model of a CPS application can be roughly divided into three parts: (1) the plant model representing an encapsulation of the physical components in the system, (2) the controller model representing the software used to regulate the plant, and (3) an environment model representing exogenous disturbances to the plant.

Given plant, controller and environment models of a system, in order to perform any of the aforementioned analyses, a crucial step is to articulate the goal of the analysis as a formal specification for the system. The analysis problem can then check whether the system implementation is a refinement of its specification. However, the state-of-the-art in industrial settings is that formal specifications are rarely found. Specifications exist in the form of mental models of correctness formed by engineers through their design insights and experience, or visual depictions in the form of simulation plots, and occasionally as legacy scripts and monitors. None of these are formal, machine-checkable unambiguous specifications. In the industry, engineers often use the term requirements instead of specifications. Typical industrial requirements do not arise from principled software engineering approaches to develop CPS software, but rather are summaries of discussions between developers and their customers. While the state-of-the-art for requirements/specifications in industrial settings is far from ideal, in academic settings, there is a problem of having a wide choice between a number of specification formalisms, primarily being developed by the formal methods community. On the other hand, application-specific academic domains such as robotics, biological systems, and medical devices may not always articulate formal system specifications.

The overarching goal of the seminar was thus to address the following question: *Is there a universal specification formalism that can be used as a standard language for a variety of modern cyber-physical systems?* To address this question, this seminar was divided into three broad thrusts:

- State-of-the-art in general specification formalisms,
- Domain-specific needs and domain-specific specification formalisms,
- Expressivity, Monitoring Algorithms and Analysis concerns for a specification language.

Outcome of the seminar

The seminar had a total of 37 participants with a mix of research communities including experts (both theoreticians and practitioners) in formal methods, runtime monitoring, machine learning, control theorey, industrial IoT, and biological systems. The seminar focused on the cross-domain challenges in the development of a universal specification formalism that can accommodate for various CPS applications.

The seminar provided an excellent overview of requirements from various application domains that paved the road for identifying common features in a cross-domain specification language. As another outcome of the seminar, we defined as a community the following next steps:

1. Identification of various benchmark problems for monitoring specifications at runtime, and learning specifications from data.
2. Standardizing syntax for expressing time-series data, such as comma separated values (CSV) with a well-defined header file.
3. Creating a public repository containing traces, specifications, models, and pattern libraries.
4. Coordination with RVComp, a runtime verification competition collocated with the Runtime Verification (RV) conference, and possible coordination with SygusComp (Syntax-guided synthesis competition) and SYNTComp (Synthesis competition) to arrange special tracks on learning specifications.

5. Creating a public repository containing standard parsers for variety of specification formalisms such as variants of Signal Temporal Logic.

Sessions

The seminar was organized as a sequence of open discussions on pre-defined topics of interest. Each session had one or two moderators who introduced the topic and one or two scribes who recorded the proceedings of the discussions. The moderators had a short introduction of the topic, identifying the most important sub-topics for open discussion. The discussions were structured in following sessions:

Day 1 State-of-the-art in general specification formalisms
1. Specification languages in digital hardware
2. Tools perspective
3. Overview of declarative specification languages

Days 2 and 3 Domain-specific needs and domain-specific specification formalisms
1. Specifications in automotive systems
2. Specifications in robotics and perception
3. Specifications in Industry 4.0, EDA and mixed signal design
4. Specifications in smart cities
5. Specifications in bioloty
6. Specifications in medical devices
7. Specifications in security

Days 4 Expressivity, monitoring algorithms and analysis concerns
1. Algorithms for specifications: specifications for learning *versus* learning specifications
2. Streaming languages
3. Runtime monitoring
4. Expressivity

Day 5 Next steps and summary of the seminar outcomes

We also organized on Day 1 a session to honor the memory of Oded Maler, one of the co-organizers of this seminar, and who sadly passed away in September 2018.

2 Table of Contents

3 Summary of Discussions

3.1 Specification Languages in Digital Hardware

Dana Fisman (Ben Gurion University – Beer Sheva, IL)

Property Specification Language (PSL) is an IEEE standard specification language used in the digital hardware domain. PSL is an example of a successful development of a formal and mathematically rigorous language that is widely adopted in the semi-conductor industry. In this session, we recalled the steps that led to this success story.

Before year 2000, there was no standard for hardware specification language. Users of formal verification tools by different companies used different languages (Motorola's CBV, IBM's Sugar, Intel's Forspec and Verisity's Temporal e). Semi-conductor companies and Electronic Design Automation (EDA) tool vendors decided that there was a need for a standard specification language. A committee with 30 persons was created that included industry representatives but also leading formal methods researchers. 70 requirements were collected and 74 examples of industry properties were expressed in the 4 existing languages. It was decided that 1 of the 4 languages should be used as the basis for the standard, and after two rounds of votes IBM's Sugar was selected. The new language was named PSL.

There were 3 major changes from Sugar to PSL:
- PSL was partitioned into boolean, temporal, modeling and verification *layers*.
- Different *flavors* were added to PSL, supporting Verilog, VHDL, SystemVerilog, SystemC and GDL flavors.
- Linear-time temporal logic was used instead of its branching-time variant.

Major features in PSL are:
- Regular expressions and trigger operations,
- Reset and abort operators,
- Clock/sampling operators,
- Distinction between strong and weak regular expressions, and
- Local variables.

It took 6 years for PSL to become an IEEE standard and 8 years to complete all the features that are in the language today.

3.2 Tools Perspective

Alexandre Donzé (Decyphir – Moirans, FR) and Akshay Rajhans (MathWorks, US)

There are three key personas in the development of CPS: (1) the academic researcher, (2) the tool developer and (3) the industrial practitioner. The application of specification languages in practice is to a large extent conditioned by tool support. We discussed the potential use of specification languages in the development of CPS from the perspective of two tool vendors:
- A major international corporation that specializes in the development of mathematical computing software, which in particular supports data analysis and simulation.

- A small and medium enterprise that develops the toolbox for the creation of test suites, formal requirements and automatic model falsification and calibration.

The first observation made was regarding the commonly accepted assumption in the formal methods community that engineers have the models and that simulation is easy, which turns out to be wrong in reality. In the industrial practice, model-based design (MBD) is its infancy because modeling is hard and adds an additional burden to the engineers.

Requirements play an important role in the development of complex systems. Requirements can come from:

- Marketing and executive managers,
- Designers,
- Test, quality and verification engineers, and
- Certification and regulation bodies.

Requirements can be about many aspects of the design: function, architecture, implementation, integration, performance, memory consumption and energy consumption. One natural question that is studied in the field of requirement engineering is how to effectively communicate requirements, designs, test and certification results, and other design artifacts. Natural language documents are still the primary certificates that requirements are satisfied, and are commonly managed by tools that are organized around a database. An engineer usually creates a written requirement paper trail in a two step process:

- Actually designs and tests the systems.
- Chooses the nicest curve in the set of simulations/data measurement and attaches it to the report/requirement tool.

This process can be improved by using specification languages to express requirements. Specification languages, such as Signal Temporal Logic (STL), provide readable specifications that are actionable on modeling and simulation tools. There is an inherent conflict between the academic researcher, who aims for an elegant and formal specification language with proofs and guarantees, and the industrial practitioner, who wants a scalable and usable solution that works for all use cases and can be adapted to the existing workflows. The tool developer attempts to resolve this conflict. Specification-based tools are used in some industries to support MDB, and were able to successfully find bugs in production models with legacy code, helped synthesize a controller that is entering production and supported the development of a full suite of requirements for a next generation of automotive production model. However, the adoption of such a methodology is not straightforward and some challenges have been identified:

- Practically all requirements are of the form $\mathsf{always}(\mathsf{A}\ \mathsf{implies}\ \mathsf{eventually}\ \mathsf{B})$, where A represents steps, ramps and steady-state conditions, and B represents ranges, overshoots and undershoots– do we then need the full power of a formal specification language?
- A often comes from a noisy environment and is hard to capture with a formal specification language such as STL

There are additional challenges due to the heterogeneity of:

- Domains – automotive, avionics, maritime, medical, etc. Each domain comes with its terminology, standards, regulations and best practices.
- Modeling formalisms – acausal, causal, continuous time, discrete time, event-driven, etc.
- Simulation platform: ODE, DAE, FEM, geometric, etc.

An additional challenge is that the continuous engineering support during system operation becomes increasingly important, requiring a design-operation continuum, with time scales that range from milliseconds to months.

The main lesson learned is that tools should support, and not additionally burden, the modeling effort. Crafting requirements is almost as hard as modeling.

3.3 Declarative Specification Languages – Expressiveness

Thomas Ferrère (IST Austria – Klosterneuburg, AT)

Declarative languages based on temporal logic and regular expressions have been popular specification formalisms for decades. In the context of discrete-time Boolean-valued reactive systems, the properties of different specification languages have been extensively studied. Linear temporal logic (LTL) a popular specification language introduced in the 70s has been shown to be expressively equivalent to start-free regular expressions, monadic first-order logic and counter-free languages. Adding the modulo-counting capability gives Linear Dynamic Logic (LDL), which is expressively equivalent to regular expressions and monadic second-order logic. One of the most important results is that all these formalisms translate to finite automata. Both the complexity and the succinctness are well-studied problems for these classes of specification languages.

Increasing the expressiveness of these languages can be done in several directions. For instance, proper counters and adders can be added, resulting in formalisms that translate to infinite state systems. Another way to increase the expressiveness of classical specification formalisms is to add explicit notion of real-time, resulting in formalisms such as Metric Temporal Logic (MTL), Metric Interval Temporal Logic (MITL), Metric Dynamic Logic (MDL), Metric Interval Dynamic Logic (MIDL), Timed Regular Expressions (TRE), Balanced Timed Regular Expressions (BTRE) and Metric Monadic First-Order logic. Allmost all of these languages can be translated to either timed automata (TA) or alternating timed automata (ATA). The dense time lifting of the discrete time formalisms comes at a cost. For instance, universality of TRE is undecidable, while emptiness is PSPACE. Universality is non-elementary for MTL and MDL and EXPSPACE for MITL and MIDL. There are partial results on the expressiveness of real-time formalisms. TRE and MTL are expressively incomparable, star-free MIDL is more expressive than MITL (assuming constants in $\mathbb{Z}$) and MMFO is equally expressive to MTL (assuming constants in $\mathbb{Q}$).

Dense-time formalism have been used as the basis for specification languages applied to cyber-physical systems (CPS). For instance, Signal Temporal Logic (STL) extends MTL with numerical predicates. In the context of CPS, one of the main extensions that increased the power of the specification languages was the introduction of quantitative semantics that allowed to replace the binary satisfaction relation with a real-valued robustness degree, which indicates how well a behavior satisfies or how badly it violates a specification. Another increase of expessiveness was achieved by adding quantification in Signal value-freezing logic and in Signal First-Order Logic.

Some system aspects cannot be specified with any of the described formalisms, such as controllability, security, multi-agent system properties and spatial relations. There are already existing efforts that develop dedicated formalisms, such as hyper-temporal logics and spatio-temporal logics, which address some of these aspects.

3.4 Specification in Automotive Systems

James Kapinski (Toyota Research Institute North America- Ann Arbor, US) and Jens Oehlerking (Robert Bosch GmbH – Stuttgart, DE)

The session was scribed by Katie Driggs-Campbel and Nikos Aréchiga.

In the automotive domain, formal specifications are slowly being integrated into the model-driven development methodologies. However, there are multiple open challenges that were identified in the seminar.

Existing formalisms are difficult for engineers to use. Cars are designed by mechanical engineers, raising the question what type of formalisms are appropriate for people with this background. An ideal formalism shall not only be powerful and expressive, but also succinct and intuitive. We collected several challenges in using formal specifications:

- Requirements often do not match what the engineer mean, making the translation from natural-language requirements to formal specifications non-trivial.
- Specifications also abstract the expected behavior, raising the question how accurate a specification shall be.
- Describing corner cases often results in the blow-up of the specification size.
- Although it is important to improve the specification languages, it is equally important to broaden the education of the engineers.

Some of the above challenges can be addressed by providing structured natural languages and domain-specific template languages to bridge the gap between theoretical computer scientists and mechanical engineers. Another way to bridge this gap is to adapt specification languages to the modeling formalisms, such as Simulink, that the engineers are already familiar with. Many concerns do not seem to be fundamental and rather relate to the lack of tools or poor implementations. There is also a vocabulary gap – engineers in the automotive industry usually use the term "specification" to denote a simulation model of the systems. Finally, regarding the education of engineers, it may be only question of time. For instance, it took decades for model predictive control (MPC) to become mainstream, and that was a matter of graduating enough PhDs who took courses on MPC. Perhaps it is the same for formal specifications, which may become widespread after sufficient people with proper training become available on the market.

Incorporating specification formalisms into the existing development processes represents a considerable effort. For production people, they already don't have enough time. Formal specifications are not yet mandated as a necessity, which prevents their widespread use in the automotive domain. In that context, the aerospace domain seems to be more progressive in terms of adoption of formal specifications.

Increasing the adoption of formal specification could be done by facilitating specification activity. There is existing work on learning specifications from traces, but the quality of learned specs is not always satisfactory (e.g., because specs are complex). To improve the situation, specification mining should have a benchmark; there should be a model and some requirements that are known to be true, and then there should be a competition to see who can identify the most parsimonious requirements. The goal shouldn't be learning one single specification, what you really want is a distribution of beliefs over possible concepts.

Lots of resources are currently devoted to OBD (on-board diagnostic) development. It seems like a great application for specification-based monitoring. This requires certification of monitor compilers and qualified monitors.

There are systemic (organizational) roadblocks to implemented formalized specs (e.g., legacy code, dogmatic development philosophies). How can we mitigate these? If you were thinking in terms of development processes, what could we do differently, in terms of developing new formalisms that minimally impact existing processes?

3.5 Specification in Robotics and Perception

Katie Driggs-Campbell (University of Illinois – Urbana Champaign, US) and Georgios Fainekos (Arizona State University – Tempe, US)

The session was scribed by Ezio Bartocci and Marcell Vazquez-Chanlatte.

Robotic and other autonomous (multi-agent) applications are extremely complex and are difficult to analyze. Typically, the decision flow in autonomous systems starts with the sensor measurements, which are fed to the perception module. The perception module detects objects, tracks lanes, etc. Based on this information, the autonomous vehicles makes tactical decisions and plans the trajectory, which is executed by the low lever controller. Verification and validation is often done in a simulation environment before deploying the autonomous system. We identified several requirements for synthesis and analysis of autonomous systems.

Requirements for synthesis:

- Flexible mission specification and expressivity
- Multi-agent systems and interaction
- Multi-objective tasks
- Assumptions vs. guarantees in unstructured environments

Requirements for analysis:

- Safety assessment and validation
- Reasoning about uncertainty
- Safe domain transfer
- Providing guarantees for complex systems like localization and perception

Based on the identified requirements, the following questions naturally arise:

- Some aspects can be formally specified, but how do you formally prove them?
- Some aspects are less evident to specify – how do you formally specify a classifier?
- $\rightarrow$ What does and does not make sense to formalize?

There have been different approaches to tackle specification, analysis and synthesis of robotic applications. Instead of doing verification and validation, some researchers and companies suggest incorporating heuristics that avoid putting the autonomous vehicle at fault.

The perception module seems particularly problematic with respect to formal specifications. How would one even describe a property of correct perception (or any other classifier)? The state-of-the-art in vision is to compare the algorithms to some ground truth. This does not seem to be sufficient for safety critical applications, as demonstrated by recent works on adversarial testing. One way to tackle this problem is to specify perception properties indirectly – for instance one can define a specification that requires the object (and its label) permanence once that it is detected.

There have been recently new approaches for testing autonomous systems with machine learning components in the loop. Such techniques allow to reason about the impact of machine learning to the overall system safety.

In autonomous and robotic applications, both assumptions and guarantees are likely to be from time to time violated. In that case, the system should aim for minimum violation of its constraints. There are approaches that address this problem by quantifying violations. Reasoning under uncertainty is a related error, where mostly probabilistic models are used. Specifications are often implicitly embedded in the cost function, with a lot of effort going into cost engineering.

Autonomous and robotic applications are good examples of domains with multiple types of specifications. There is first a distinction between soft and hard constraints. Then, there are functional specifications, but also ethical and legal requirements. There are also descriptions of abstract and concrete scenarios. One way to address the multitude of specification classes is to partially order specifications according to their importance and urgency.

3.6 Specification in Industry 4.0, EDA and mixed-signal design

Gustavo Quirós (Siemens – Princeton, US) and Radu Grosu (TU Wien, AT)

The session was scribed by Thomas Ferrère and Niveditha Manjunath.

The goal of this session was to identify challenges of using specification languages in (1) Industry 4.0 and (2) semiconductor industry. The session was started with the introduction of Industry 4.0 and highlighting the new aspects brought with the so-called fourth industrial revolution: (1) more horizontal organization structure and flexible manufacturing, (2) digitalization of the production, (3) use of artificial intelligence and of digital twins.

Specifications are foreseen to play an important role in many aspects – plants already have requirements regarding cost, efficiency, safety, regulations and feasible production. Specifications can support model-driven development methodologies and monitoring various properties of the plant. The state-of-the-practice is to use natural language for specifications. Currently, only specific aspects of a plant a modeled in a simulation environment. For instance, the whole plant can be modeled as a flow (with the queue semantics) to find bottlenecks, while the behavior of an industrial robot can be explored using simulation with geometry. Process control is typically responsible for monitoring different aspects of the plant. There are visual aids and logical alarm systems observed over multiple shifts, which allows automated monitoring of thousands of variables. The human is nevertheless needed in the loop to supervise the process and intervene if necessary.

Using formal specifications in Industry 4.0 raises some basic questions and challenges:

- What are the boundaries of CPS in Industry 4.0?
  - These boundaries are complex, but need to be well defined
- What aspect should be specified?
  - Specification of the overall process, plan automation and timing, communication, etc.
- What properties should be specified?
  - Behavior, timing, safety, security, cost and productivity, availability, etc.
- What is the intention of the specification?
  - Procurement, scheduling, monitoring, etc.

- Who will write and use the specifications?
  - Device manufacturers, plant designers, integrators etc.
- How to compose, update and maintain specifications?

The semiconductor and Electronic Design Automation (EDA) industries were early adopters of specification languages in the context of digital hardware. This situation is however very different when considering mixed signal design. Mixed signal design brings an entirely new set of challenges. These challenges are illustrated by the difficulty to formally specify a spike, a temporal pattern that is found not only in the semiconductor domain, but also in many other fields such as medical electrocardiograms.

What is a spike? There is no satisfactory formal specification of a spike yet. People use an informal procedural description: "follow the signal up, then down, and do not consider spikes for some time following a spike, etc". This procedural description is translated into a program that detects spikes and outputs segments of the observed behavior that match it. It follows that the spike is defined by the monitoring program itself. This is problematic for several reasons:

- The monitoring program is the spike specification → the program is always correct
- Without a formal specification of a spike, one cannot compare two competing monitoring programs that detect spikes

Describing declaratively a spike in time domain seems to be a challenging problem. The noise in the observed waveform may have a similar shape to a spike. The shape of the signal also varies in frequency. As a consequence, some properties seem to be more naturally expressible in the frequency domain, for instance using wavelets. Another possibility would be to use spatial logics in order to describe patterns in 2-dimensional spectrograms.

The problem of detecting spikes opens another natural question – can we use machine learning to infer a model of spikes from previous observations and labeled data and use it as a classifier to detect newly observed spikes? In principle yes, but we lose guarantees about what we detect and what we miss. The added value of a formal specification with respect to machine learning is that specifications can be understood, interpreted and hence debugged. However, formal specifications and machine learning can be complementary, exploiting the strengths of both approaches.

3.7 Specifications in Biology

Paul Bogdan (USC – Los Angeles, US) and Thao Dang (VERIMAG – Grenoble, FR)

The session was scribed by David Šafránek.

The problem of multi-scale modeling is becoming increasingly important in the areas of intra-cellular processing, cell-to-cell communication as well as communication between groups of heterogeneous cells and populations of different cell types (e.g., microbiome to brain cell interactions).

The key features highlighted in the presentation were the following:

1. Time-dependent signature of biological systems;
2. Stochasticity signature of biological systems;

3. Hybrid modelling (hybrid systems) of biological systems
4. Fractal dynamics (correlated growth) signature of biological systems;
5. Specifying patterns targeting collective behaviour (organisation) at non-equilibrium based on geometric metrics such as orientation angles;
6. Specifying performance of the biological system – typical metrics are not the ones we use in traditional engineering such as delay and throughput, but rather need to quantify and cover learning, adaptation, robustness, and evolution;
7. Need for better metrics to quantify degree of emergence, self-organisation, missing information, intelligence, and complexity;
8. Performance optimization: for a swarm of agents – a performance envelope will imply that the collective system has or exhibits a certain range of degree of emergence, self-organization, robustness, and complexity;
9. Model validation against data: here, formal specifications are hypotheses based on an experimental observation.

Some of the highlights of the discussion were as follows:

- Geometric metrics associated with patterns seem to provide a cost function that is optimised in every step of the dynamics depending on the local neighbourhood (local rules) - indeed these geometric metrics can also be linked with statistical physics concepts such as entropy and free energy.
- Specification of patterns can be probably reduced to the constraints on a cost function, however, find the correct cost function may not be always possible either due to the complexity introduced by the system cardinality, system heterogeneity or unpredictability in reaction behavior to targeted environmental perturbations (the environment does not affect all individuals in the same way).
- The form of hybridisation used for the models (discrete switches vs. ramps) can have implications on optimization, e.g., ramps are better for purposes of optimization.
- Can spatio-temporal behavioral patterns be considered as requirements? The answer was, "In general , yes," but in the context of the discussion, there were only four different collective behaviours considered. The patterns discussed were just representative examples, and there might be richer patterns in reality for testing mathematical formalisms for quantifying emergence, self-organization and complexity of collective biological systems and swarms.
- The problem has a counterpart in synthetic biology – can we design a synthetic population that follows the selected patterns, where patterns can be used to measure system complexity (specification therefore targets the system's complexity)?
- Specifications can often be viewed as probabilistic distributions, so patterns generally describe distributions.
- Specification frameworks should allow to determine critical points that distinguish the patterns through mean-field approximation. There is a need to cope with emergent behaviour – new behaviour not known in advance. Emergence is defined as being proportional with the rate of change of multiscale entropic functionals, which can quantify the seen and unseen.
- Sometimes a specification can be an approximation of the desired property, this does not mean the specification is bad.

Summary

Biological systems that have to be considered and studied in 3D space present a number of mathematical characteristics and challenges for behaviour specification such as time dependent, multiscale, multifractal / hierarchical, heterogeneous, emergent, self-organizing, robust and complex / intelligent behavior. So, how do we mathematically express all these characteristics? Uncertainty quantification is a major challenge. We are required to deal with uncertainty in modeling because it is not guaranteed that all we observe represents all variables that are intrinsic the signature variables of the complex system, the mode structure is uncertain and implicitly also there is uncertainty of the inferred parameters. Models are uncertain (the need for parameter synthesis wrt the given constraints including performance metrics). During the evolution of the population the information entropy and the degree of emergence / self-organisation / robustness / intelligence and complexity change therefore the specification must deal with such time varying collective system metrics and allow to capture emerging knowledge (evolutionary specification). Spatio-temporal reasoning needs to be combined with the ability to describe time varying multi-fractal probabilistic distributions. There is the need to capture time-dependence and higher-order variability (difference between distributions).

3.8 Specifications in Medical Devices, Cyber-Human Systems and Smart Cities

Houssam Abbas (Oregon State University – Corvallis, US), Lu Feng (University of Virginia – Charlottesville, US) and Katie Driggs-Campbell (University of Illinois – Urbana Champaign, US)

This session was scribed by Caleb Stanford.

Medical devices

We discussed some of the issues in specifying CPS specific to medical devices. Some of these issues are:

1. Lack of standardized criteria: Different manufacturers are using different criteria on their devices, and everyone says their algorithm is right. Typically, the overall detection algorithm is structured as a *decision tree* of several different criteria.
2. Lack of a standardized language for different devices: For example, one particular heart monitor vendor uses regular expressions for programming them, but the others don't. It was also discussed how different types of devices (e.g. heart monitor vs. brain EEG) can be very different: although a heart monitor only has 2-3 sensors, EEGs have 100s. And a heart monitor is implanted whereas an EEG is not (so different performance concerns). Can temporal logic support what is needed in all different device types, as well as different device manufacturers for the same type?
3. For high-risk medical devices, resistance to huge changes due to difficulty of the approval process. Each change has to go through clinical trials (small-size, then medium size, then full scale), which takes 2-3 years, millions of dollars, and multiple doctors on staff, as well

as dealing with some legal preparations. The result of this is that incremental changes are preferred, and manufacturers want to be confident of the changes before going through the approval process.
4. Interpretability is key! For medical devices to be useful in practice, doctors need to know what the device is doing and be able to modify a small number of parameters to tune to the specific patient (who will go to regular check ups to make sure the device is working well). There was a discussion about whether neural networks (or some other opaque machine learning model) would be able to overcome this barrier. Would it be possible to adapt the algorithm to a particular patient if it was an opaque machine learning model?
5. Energy consumption is the most important performance concern. If you can make it last 3 more months, over 9 years, that's worth it. Secondary concerns include memory (which is very cheap) and decision time (so medical problems can be detected faster).

Specifications for Cyber-Human Systems

We discussed some of the problems specific to cyber-human systems:
1. Device failure due to human error: Unlike in purely autonomous systems, cyber-human systems can fail due to no fault of the system itself, but due to the fault of the human component. It is an open research challenge how to specify and deal with this kind of error. Error seems to arise especially when humans are new to the system – see plane example, where crashes start out extremely high on each new type of aircraft, but then decay off as time goes on. It turns out this is NOT due to software bugs, but due to human error of the pilots. In the diabetes patient example, patients were ignoring the amount of insulin to inject and instead just doing the maximum, or none at all, in some cases. To deal with human error, traditional research models the *reachable space* of human behaviors.
2. The level of trust that the human has in the system becomes a relevant variable that we have to model. Lots of discussion on why measuring and/or controlling the human trust might be useful, and what approach should be taken. If the goal is to make the human trust the system, research shows humans should be provided with just the right amount of detail – not too little, and not too much – about how the system works. Examples: alarm fatigue; info about how a self-driving car works. There is a logic, PRTL* that models an agent's trust over another. How does trust change over time? A comment was that trust may be related to knowledge about the system (see epistemic logic, common knowledge). But is the goal always to make the human trust the system more? We agreed that if the system may fail, the goal may be to make the human trust it less. So you want to avoid undertrust and overtrust. For measuring trust: heart rate and other physiological signals were suggested. Finally, who should really be trusted in case of disagreement – the human or the machine? Who is the certified agent? This may change over time as systems reach super-human performance; for now in domains like autonomous driving, we trust the human driver more.

Smart Cities

Smart cities represent urban environments where there is a combination of sensor networks gathering real-time data about various aspects of the city, and control algorithms that seek to regulate certain characteristics. Examples include sensor networks to monitor air-quality, noise-level, traffic, wait-times for emergency response, etc. Typical requirements are spatio-temporal, requiring specific geographic locations in the city to have certain restrictions on the behaviors of constituents of the city such as cars, construction crews, energy consumers, etc. These requirements often require encoding aggregate statistical properties. Certain approaches have sought to control violations of smart-cities requirements through dynamic monitoring and enforcement approaches. In large-scale, distributed cyber-human systems: properties have to be checked locally *distributed monitoring*, not globally. Specifically we discussed smart cities. It's too expensive to aggregate for the whole city, so sensor measurements are aggregated locally.4. In large-scale, distributed cyber-human systems: synchronization of measurements? In current smart cities, the measurements are all at the same rate (1 per minute, e.g.) but not necessarily synchronized due to clock drift or small clock differences. Still, they may be able to be treated as synchronized.

3.9 Specifications for Security

Borzoo Bonakdarpour (Iowa State University – Ames, US) and Bernd Finkbeiner (Universität des Saarlandes, DE)

The session was scribed by Ana Oliveira da Costa.

In the domain of cyber-physical security, use of formal specification languages is fairly new. Inspired by work in cyber-security for software systems, the formalism of *hyperproperties* has been gaining traction. The motivation for hyperproperties comes from information flow and security. The basic idea is that there are two classes (public and secret) for input and output ports, and we want to avoid public outputs to depend on the secret inputs. Checking any property of a single trace does not typically reveal any leakage of secret information; instead we need to typically inspect a set of traces. A hyperproperty is thus equivalent to a set of sets of traces. A system satisfies a hyperproperty if the set of traces generated by the system is an element of the hyperproperty. Traditional formalisms such as LTL and CTL cannot express such properties. HyperLTL solves this limitation by extending LTL with quantification over traces. There are many hyper-logics: HyperLTL, HyperCTL, HyperCTL*, HyperPCTL, HyperPCTL*, HyperSTL (which respectively augment LTL, CTL, CTL*, PCTL, PCTL* and STL with universal and existential quantifiers over trace variables).

We discussed a number of questions related to hyper-properties and adapting them to the domain of cyber-physical systems. As the study of hyperproperties is a relatively nascent field, initial discussions revolved around specific use cases for hyperproperties. In particular, the discussion identified the following system properties that can be expressed using different hyper-property formalisms (such as HyperLTL, HyperCTL, HyperCTL*, HyperSTL, HyperPCTL, etc.):

1. *Observational Determinism*: A general definition of observational determinism is that if the public data in an input $u(t)$ is identical to that of another input (say $u'(t)$) for all time t, then the system observations for $u(t)$ and $u'(t)$ should be identical. A particular definition of this property in HyperLTL was discussed, where the public inputs were required to be identical only initially.
2. *Non-interference*: Various versions of non-interference can also be expressed as hyper-properties, that can also be captured with HyperLTL.
3. *Non-inference*: This is another important property that can be expressed using HyperLTL.

4. *Fault-Tolerance*: The property that the output of a program always produces a string that is a guaranteed Hamming distance away from some other run of the program is a hyper-property that is related to fault-tolerance.
5. *Robustness*: Robustness defined as a specific form of continuity is a hyper-property. Abstractly, this property states that for two inputs to a program at a certain distance d, the outputs of the program are within some factor of the distance d. This hyper-property can be expressed in HyperSTL.
6. *Differential Privacy*: Differential Privacy is a property that says that if we remove any one user's data from a given database, then the results of computing certain kind of statistical properties of the database should not change significantly. The logics HyperPCTL and HyperPCTL* can express such properties.
7. *Causality*: The logic HyperPCTL*, which allows expressing conditional probability was an extension of HyperPCTL that was introduced to express causality properties. One form of causality, known as Granger causality states that a random variable x is believed to be a cause for a random variable y if the correlation of x and y is stronger than the auto-correlation of y. Computing autocorrelations and correlations require expression of conditional probability.
8. *Opacity*: Opacity is a condition used in imperfect information games, where the attacker is required to never know some secret condition of the defender. This is a hyperproperty, and there was a brief discussion on quantifiers in strategy logic, and its relation to the opacity condition for imperfect information games. There was limited discussion on how such questions to be tied to zero-knowledge proofs.

We discussed several theoretical issues related to hyper-properties, many of these questions do not have answers yet, but we are hopeful that the seminar participants will investigate these questions in future research. We enlist the prominent questions raised below:

- Can HyperLTL express input-output automata, which can be useful to specify asynchronous systems?
- What kinds of equivalence relations hold true for models satisfying the same hyperproperties? Logics such as CTL have the property that bisimilar models satisfy the same CTL properties. Is there an analogous property for HyperLTL or other Hyper logics? Linear time properties may be preserved by trace equivalence, what about branching structure?
- What kinds of tools exist for monitoring and model checking hyper-properties? Model checking is not decidable in general, but there are decidable fragments. Is there a taxonomy of Hyper-logics vis-à-vis complexity of satisfiability or model checking?
- Can certain kinds of HyperSTL properties be recast as STL properties? This is always possible for properties which can be checked by creating several copies of the system model and running these several copies independently (on possibly different inputs). However, we discussed some HyperSTL properties where this is not possible. Examplex include properties relating to incremental stability, Lyapunov stability, etc.
- Is there any natural way to have a state-based/automata based verification for hyper-properties? If we prefer automata over LTL formulas, we can use automata as with LTL but we then need to deal (explicitly) with the prefix of the hyper-formula. An automaton that builds a relational formula would be required, where states of the automaton have some explicit relation. For certain fragments, it is possible to build automata that represent self-compositions, but this may lead to an exponential blow-up in the number of quantifiers.

- There was a detailed discussion on the robustness semantics for HyperSTL, specifically the need for an existential quantifier in the robustness semantics, especially when we have stochastic hyper-properties.
- One commenter observed that notions such as bias and fairness in machine learning algorithms can also be expressed as a hyper-property.
- Finally, there was some discussion on testing for hyper-properties. Particularly, it is possible that test coverage could itself be expressed as a hyper-property. Monitoring of a set of set of traces may be useful, and given a set of traces, the goal is to quantify how good this set is. If the notion of robustness is added to the system, then it can be used so check whether the tests have proper coverage.

3.10 Specifications for Learning/AI vs. Learning Specifications

Rupak Majumdar (MPI-SWS – Kaiserslautern, DE) and Marcell Vazquez-Chanlatte (University of California – Berkeley, US)

This session was scribed by Marcello M. Bersani.

Some of the questions guiding the discussion were: (1) How do techniques from formal methods such as reactive synthesis compare against reinforcement-learning based control synthesis? (2) What are specific challenges in CPS that need to be addressed for learning specifications?

Learning Specifications

Learning specifications from data is inspired from automata learning from examples. Inputs to learning algorithms are typically positive and negative examples or unlabeled examples. In automata learning, typically, we learn only from positive examples. In learning specifications, we can either output formulas in some logic, or automata. There was some discussion on whether the result of learning should be an automaton or a logical formula. There were arguments that logic is better for learning because it is better to understand than automata, while there were counter-arguments that automata are actually easier to understand. There was a comment that if there is a fixed structure (either in automata or logic), then it may be useful. Machine learning techniques can, in general, introduce new behaviors in the system, whereas automata-based learning does not permit this.

Specification mining focuses on learning specifications from an implemented system: here, the purpose is to identify a concise way to state how the system behaves. There may need to be a relation between inputs and outputs of the system in such specifications. In contrast, traditional specifications for reactive synthesis focus on which next move the system should make.

There was a discussion on whether STL is a good logic for learning from time-series data. There is a need for benchmark problems to compare the results of learning. A benchmark problem would consist of a well-known system with "ground truth" learned formulae obtained from the system. Using a logical formalism may be helpful to limit exploration in some way. There are many logics where learning remains an open problem. These include spatial, spatio-temporal logics and logics for hyper-properties.

There was a discssion on interpretability of the learned formulas, where it was observed that learning local properties construct a global property might help. There was some discussion on STL where inputs and outputs can be separately labeled, and this could help better learn causality among the events.

There was a discussion on learning abstractions. Based on system traces, we can learn only a partial system. This may not be a right abstraction, but could be viewed as either an over or an under approximation (or a combination thereof). In continuous systems, e.g. linear systems, approaches such as system identification help with learning the system dynamics from data. There does not seem to be a correspondence in logic, possibly because linear systems have many results that come from a frequency domain treatment. Moving to discrete world it's not clear how this can be done.

Specifications for Synthesis vs. Specifications for RL

There was an interesting analogy drawn between specifications based on logical formulas and statistical specifications for system correctness through the use of state-based rewards used in reinforcement learning. With state-based rewards you can play with a dense exploration of the state-space of a system updating rewards closer and closer to the target. We discussed if there is an analogue in the world of logic-based, specification-based synthesis. Some solutions were to use template-based logical formulas, use of counterexamples, and translation of counterexamples into automata to add knowledge to the learning process. Similarities drawn between rewards-based reinforcement learning (RL) and formula-based reactive synthesis were discussed. RL is typically successful because the global behavior is captured with local structures.

There was some discussion on why RL algorithms scale so well vs. reactive synthesis algorithms. An observation was that in reinforcement learning, the reward function used for searching is clear and its memory footprint is known, wheras it is difficult to predict the size of a BDD used for verification during reactive synthesis. An interesting aspect in RL is that multiple rewards can be combined. In reactive synthesis, we can use parity games to synthesize winning strategies. Strategies guaranteeing satisfaction of every ω-regular language can be learned using such techniques.

In RL, during model building there is no assumption on the underlying model and distributions are learned on-the-fly. Aside from learning the main behavior, we can learn also side behaviors that are invariants around that behavior. There are theoretical limits on reactive synthesis based strategy learning: Given a mean payoff game finding an automaton that describes the strategy is an open question.

3.11 Streaming Languages

César Sánchez (IMDEA Software – Madrid, ES)

This session was scribed by Hazem Torfah.

This session was an invited talk by César Sánchez on stream runtime-verification. In stream runtime verification, a formal specification is translated to a stream-based monitor. A stream-based monitor translates streams of input data into streams of output data. Output streams are used to statistically evaluate and to check assertions over the system. The goal of stream runtime verification is to easily express monitors with formal guarantees that allow for the computation of rich verdicts beyond yes and no.

Stream-based monitors are usually specified using a stream-based specification language. Prominent stream-based specification languages include LOLA Striver and TeSSLa. They provide a simplistic way for defining monitors by defining a set of equations expressing the relation between input and output streams. Stream-based specification languages:

- provide engineering friendly specification languages,
- subsume formal logics like LTL and STL,
- come with a formal semantics that allows for the computation of formal guarantees for a large classes of properties,
- provide an outline approach to runtime monitoring that is independent of the system, and allow for both online and offline monitoring of systems.

Offline vs. Streaming: A key question is how offline monitoring works with future and past, and the answer is that it has the ability to use two-way alternation. In general; however, you want to find algorithms that go in one direction to avoid memory issues if you want to use approaches like map reduce.
Landscape of streaming languages Stream-based specification languages can be classified according to three dimensions:

1. Datatypes
2. Temporal expressivity
3. Time domain

LOLA is a specification language that allows for the definition of monitors for regular properties over rich datatypes. Real-time extensions like RTLola, Striver and TeSSLa allow for the specification of monitors for timed events. Parameterization does not strictly fit into this classification, and requires a fourth dimension. An example of parameterized stream-based languages is LOLA2.0. The advantage of parameterization is that you want the compiler to handle parameterization. Otherwise you have to manage sets in the syntax.

Comparing stream-based verification vs. STL: Streaming languages provide a framework, where STL formulas can be implemented easily. If we talk about future in STL we have to introduce memory bounds. In stream-based languages, one can compute tight memory bounds. Assumptions such as bounded variability ensures bounded memory when considering real-time specifications. Stream-based languages are very controlled and specific languages that allow to encode robustness, but at the price of undefinedness for monitoring. Changing the different versions of robustness is easier in stream languages, and from experience, debugging specifications is easier for engineers. On the other hand, a disadvantage of stream-based languages is that compact formulas in STL need a lot of case distinctions in stream-based languages. Currently, there has been no comparision between expressiveness of real-time stream-based languages and STL. There are also no equivalence results between stream-based languages and a specific class of automata.

3.12 Runtime Monitoring

Caleb Stanford (University of Pennsylvania – Philadelphia, US) and Dejan Ničković (AIT – Austrian Institute of Technology – Wien, AT)

This session was scribed by Rayna Dimitrova.

This session included an invited talk by Caleb Stanford on *Quantitative Runtime Monitoring*, followed by a discussion.

Quantitative Runtime Monitoring

Typical runtime monitoring assumes that we are given a specification that is compiled to a monitor. Quantitative runtime monitoring refers to performing quantitative computations over the input stream. Such computations can include many simple operations such as adding all numbers in the stream, computing an average, or other statistics, but could permit more complicated costs. Applications of quantitative runtime monitoring include medical devices and IoT. There are several types of specification formalisms. LTL-based specification formalisms,including restrictions, extensions of LTL, as well as quantitative extensions such as STL are formalisms based on logic. Formalisms such as RT-LOLA and TeSSLa are more expressive, and are executable specifications (versus declarative specifications such as LTL).

Quantitative Regular Expressions and Cost Register Automata are formalisms that lie between stream-runtime processing languages and declarative temporal logic specifications. We explain a QRE with an example: given an input stream word w and two monitors f1, f2 that produce output on that word, compute max(f1(w),f2(w)). The main idea is to combine regular parsing of the input stream, followed by a computation over that. Example application: input stream of blood pressure measurements, partitioned into episodes of high blood pressure measurements. Match sequence of high blood pressure episodes, then match a whole day. Matching the input stream constitutes the parsing stage, then the computation stage follows. For example, compute the average for each episode, then the maximum over all episodes.

Broadly, different specification formalisms for runtime monitoring can be compared with respect to various criteria such as expressiveness, succinctness, online monitorability, and distributed monitoring. Further, specifications may be discrete-time or dense-time. Formalisms can be low-level or high-level: high-level formalisms need to be interpretable/compilable, low-level formalisms need to be compositional. We now enumerate some of the topics discussed in this session:

1. Expressiveness of QREs: For parsing of the input stream the regular expressions are assumed to be unambiguous. Unambiguous QRE-Past can be compiled to monitors of quadratic size without exponential blow-up. Check for unambiguity is done through type annotations: well-typed QRE areunambiguous. It can happen that the given QRE is not unambiguous, but in practice not that often. QREs support quantitative concatenation: given a word w, for a unique split w=w1w2, compute op(f1(w1),f2(w2)), where op is some operator. QRE supports two types of temporal windows: (1) pattern-based windows: for example, maximum over last 3 episodes, (2) time-based windows: for example, average over last 10 minutes. For each type, there are two variations: tumbling (i.e., “last ...”) vs sliding (i.e., “each ...”). One question was why not directly have built-in constructs like

average, but instead have time windows and do computations over them. The reason for this is that the window has to be a regular pattern.F inally, we noted that QRE-Past are efficiently monitorable.

2. Online Monitorability: The classical example of a specification that cannot be checked in a streaming fashion is the LTL property $\mathbf{G}(p \implies \mathbf{F}q)$. However, if $\mathbf{G}$ and $\mathbf{F}$ are replaced with the corresponding past operators, then the specification is streamable (albeit with different semantics). Another option is to consider bounded future operators, which can then be translated to purely past formulas. Determinization incurs exponential blow-up.
3. Overhead of quantitative monitoring: For the Boolean setting, it suffices to do an online subset construction, while in the quantitative setting, with each state there is also an associated cost. One approach: use a table that enumerates all the states and dynamic programming to update cost to be in each state (cost can be infinity).
4. Trade-off between expressiveness of the formalism and compositionality: some streaming algorithms are not compositional, so cannot support arbitrary streaming algorithms.
5. Automata-based approach to monitorability: NFA for classical RE. The benefits are compilation for free, extensibility with additional constructions. Guaranteed streamability and cost of evaluation. Question: How about using RE to sequential circuits construction? Answer: Register automata go in this direction: instead of storing 0 and 1, storing different values.
6. Succinctness: Generally, there is an exponential blow-up in translation of logical formulas, i.e., LTL to Nondeterministic Büchi Automata translation. Translating QREs to cost register automata incurs an exponential blow-up. One solution is to use data transducers (DT) instead. DTs allow for modular compilation and are succinct. DT are expressive, compositional, allow efficient steaming evaluation, and are succinct. Furthermore, every well-typed query α can be translated to a DT of $O(|\alpha|^2)$ size. Evaluation of DT can be done in $O(|\alpha|)$ time per element of input stream (Under the assumption that data values are stored in constant space and operations on data values take constant time). Furthermore, it is feasible to simulate DT in some SRV stream language by simulating the transfer function of the automaton. Encoding LOLA programs into DTs is a bit tricky as at this time, LOLA assumes a single data stream.
7. Distributed online monitoring: The central question in distributed monitoring is whether once can evaluate the monitoring algorithm in distributed fashion instead of sending everything to one location. The high level goal is to optimize. There are many potential applications of distributed monitoring in edge and cloud computing. A key question is whether the distributed monitoring should be done in a bottom up or a top down fashion? Here, bottom-up means that the monitor combine values from different lower-level monitors, while top-down means that there is an explicit global property that is partitioned and distributed to different agents. Top-down is generally hard, as decompositing a global specification is hard in general.
8. QREs and Stream Runtime Verification can be possibly combined. Streaming languages provide the foundation to reason about big data streaming platforms. One issue with QREs and data transducers is that they are not dense time.
9. For monitoring IOT, or during cloud computing, the typical approahc is to use *ad hoc* approaches (e.g., code injected in routers to do filtering/monitoring). On the other hand, for edge computing: – One needs to monitor past.– Typically one needs to monitor sensor data which is slow, even if the edge computer works on much higher frequency. Thus, some latency and low rates are permissible.

10. *Parameter Invariant Monitoring*: Here, the problem is to monitor specifications that may be parametric. This is required, for example, in noisy environments, such as when monitornig patient data, or when we are required to handle inaccuracies of sensors and actuators.
11. *Specification Synthesis*: There are many extant approaches to do specification synthesis. These include learning specifications using a sketching-like approach, where we are given a template specification and are required to learn the parameter valuations. It can be also used to tune partial specifications.
12. *Learning*: Runtime verification also has applications in learning from dynamic environments. It can additionally be used for runtime enforcement and control.

There are several practical applications of runtime verification in various industries. We include a partial list below:

- Inline monitors in Java code.
- Built-in tests in Aerospace: Here, we may have automated tests upon starting the aircraft (check engine), that do runtime monitoring, but may be hand-written.
- Onboard diagnostics in car: These are simple runtime monitors in car, but more applications could be coming with autonomy. Here, some tests may include checking if a given trace that you see is it an execution of some hybrid automaton. Runtime monitors could be used as triggers for data collection, as it is too much to send all the data. Predictive monitors for maintenance such as vibration sensors, temperature sensors, could also be used.
- There may be other approaches required for system-level properties (for example hyperproperties could be used).

3.13 Expressivity extensions to Signal Processing, Spatial Reasoning

Laura Nenzi (University of Trieste, IT) and David Šafrànek (Masaryk University – Brno, CZ)

This session was scribed by Necmiye Ozay.

This session discussed extensions to logics such as STL to allow reasoning about dynamical systems, signal processing, spatial reasoning, etc. We discussed three main topics:

Specifications for phase portraits of dynamical systems

Phase portraits of dynamical systems can reveal many interesting properties of the system. Some of these properties include identifying whether a given equilibrium point is a sink, source or saddle, identifying bifurcations, etc. A hybrid logic called HUCTL (Hybrid UCTL) was discussed. The key idea is that elementary patterns describe temporal behavior in states, which could be related to stability, stabilisation, direction of the flow of the vector field, etc. HUCTL is a logic that expresses branching over labeled transitions, future and past state variables. Several interesting patterns were discussed as being expressible in HUCTL; examples include: sinks, sources, 2d-saddle points, states being in nontrivial strongly connected components, etc. Hybrid logics are expressive up to graph isomorphism. State-bindings cause exponential blow up in model checking. One of the examples discussed was expressing monotonicity in biological systems.

STL with memory (freeze operators)

A freeze operator stores the past values of a signal at specific time points. STL* is a logic that augments STL with the freeze operator. STL* has both Boolean and quantitative semantics. STL* is highly expressive; however, monitoring becomes expensive with the number of times a freeze operator (in the value domain) is nested. In applications like biology, nesting of freeze operators is not needed much. The discussion centered around possible applications where we may need to nest freeze operators? One possibility is to specify the property that the signal shape is roughly like a "staircase". STL* can perhaps express derivatives so things like ODEs can be expressed using STL*.

Signal Convolution Logic (SCL): handling noise

Traditional robust satisfaction semantics have focused on robustness in the value domain of a signal. A signal that violates the property *always* $x > 0$ at a singular time-point, e.g. $x(2) = -1$, is treated equally violating as a constant signal with value -1. This is a problem when dealing with noisy signals that may rarely violate a given threshold while almost always satisfying the threshold. The idea in SCL is to have custom operators that can filter noise in the logic. This is done by introducing a convolution operator with various kernels such as the flat, exponential, Gaussian kernel etc. SCL has Boolean and quantitative semantics and allows expressing properties such as 95% of the times, $x > 0$. Applications include diabetes monitoring.

Spatio-temporal Logic

A key question is how to combine spatial and temporal operators? One approach is to consider a signal as a spatio-temporal signal. Spatio-temporal operators such as *everywhere* and *somewhere* can be then used. Satisfaction becomes a multi-valued spatial signal, where satisfaction is a property of a node in a given graph, but is not a property of the graph. Spatio-temporal logics can be used to monitor smart city applications.

3.14 Summary and Next Steps

Dejan Ničković and Jyotirmoy V. Deshmukh

The seminar provided an excellent overview of requirements from various application domains that paved the road for identifying common features in a cross-domain specification language. We divided the summary of the seminar into broad areas and challenges and learnings that the participants identified as the most interesting to them as follows:

Specification Engineering

1. How can we communicating specifications to users?
2. How do we introduce different levels of abstractions for specs, e.g., trace-local vs. trace-global vs. system-level?
3. How do we create frameworks for expressing different formalisms based on how the spec is used (spec for monitoring vs. synthesis vs. learning vs. coverage/test-quality)?
4. Frameworks for evaluating succinctness tradeoffs.
5. Evaluating and quantifying interpretability of specs.

Standardization: Process

1. How can we translate lessons learned during PSL standardization to formulate specification languages for CPS?
2. Perhaps we can have a common parser for some of the popular logical formalisms, inspired by the SMT-Lib effort.
3. We should create benchmark problems and domain-specific tutorials.
4. We should have an industrial advisory board that consists of tool vendors to lead the effort.

Standardization: Tools

1. We identified the need for tools in specification mining, tools that allow the logic being considered to be easily extended.
2. Interoperability between tools was identified as sorely missing.
3. Identification of various benchmark problems for monitoring specifications at runtime, and learning specifications from data is needed.
4. Standardizing syntax for expressing time-series data, such as comma separated values (CSV) with a well-defined header file was suggested.
5. Creating a public repository containing traces, specifications, models, and pattern libraries was suggested.
6. It was suggested that we seek for avenues to increase coordination with competitions like RVComp (for runtime verification), SygusComp (Syntax-guided synthesis) and SYNTComp (Synthesis). These could include special tracks for learning specifications.
7. Creating a public repository containing standard parsers for variety of specification formalisms such as variants of Signal Temporal Logic.

Specification formalisms

There were many participants that felt that they learned about a new specification formalisms from a domain that was not very close to their own:

1. Specifications for stability/robustness,
2. Spatio-temporal logics,
3. Trigger-based syntax for STL-like logics (inspired by PSL),
4. Hyperproperties,
5. Streaming Languages,
6. Spatial Dynamic Logics,
7. Mixed-time logics,
8. Information theory based specifications on emergence

The seminar also recognized that certain formalisms such as epistemic logics, knowledge-based specifications were not discussed. A general question of how we can clean-up specification formalisms to make them easily understandable was also identified as a challenge.

Acknowledgments

This report was partially supported by the Productive 4.0 project (ECSEL 737459). The ECSEL Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway.

Participants

- Houssam Abbas
Oregon State University – Corvallis, US
- Nikos Aréchiga
Toyota Research Institute – Los Altos, US
- Ezio Bartocci
TU Wien, AT
- Marcello M. Bersani
Polytechnic University of Milan, IT
- Paul Bogdan
USC – Los Angeles, US
- Borzoo Bonakdarpour
Iowa State University – Ames, US
- Chih-Hong Cheng
fortiss GmbH – München, DE
- Thao Dang
VERIMAG – Grenoble, FR
- Jyotirmoy Deshmukh
USC – Los Angeles, US
- Rayna Dimitrova
University of Leicester, GB
- Alexandre Donzé
Decyphir – Moirans, FR
- Katie Driggs-Campbell
University of Illinois – Urbana Champaign, US
- Georgios Fainekos
Arizona State University – Tempe, US
- Lu Feng
University of Virginia – Charlottesville, US
- Thomas Ferrère
IST Austria – Klosterneuburg, AT
- Bernd Finkbeiner
Universität des Saarlandes, DE
- Dana Fisman
Ben Gurion University – Beer Sheva, IL
- Felipe Gorostiaga
IMDEA Software – Madrid, ES
- Radu Grosu
TU Wien, AT
- James Kapinski
Toyota Research Institute North America - Ann Arbor, US
- Martin Leucker
Universität Lübeck, DE
- Rupak Majumdar
MPI-SWS – Kaiserslautern, DE
- Niveditha Manjunath
AIT – Austrian Institute of Technology – Wien, AT
- Stefan Mitsch
Carnegie Mellon University – Pittsburgh, US
- Laura Nenzi
University of Trieste, IT
- Dejan Ničković
AIT – Austrian Institute of Technology – Wien, AT
- Jens Oehlerking
Robert Bosch GmbH – Stuttgart, DE
- Ana Oliveira da Costa
TU Wien, AT
- Necmiye Ozay
University of Michigan – Ann Arbor, US
- Pavithra Prabhakar
Kansas State University – Manhattan, US
- Gustavo Quirós
Siemens – Princeton, US
- Akshay Rajhans
MathWorks, US
- David Šafrànek
Masaryk University – Brno, CZ
- César Sánchez
IMDEA Software – Madrid, ES
- Caleb Stanford
University of Pennsylvania – Philadelphia, US
- Hazem Torfah
Universität des Saarlandes, DE
- Marcell Vazquez-Chanlatte
University of California – Berkeley, US

Report from Dagstuhl Perspectives Workshop 19072

The Role of Non-monotonic Reasoning in Future Development of Artificial Intelligence

Edited by

Anthony Hunter[1], Gabriele Kern-Isberner[2], Thomas Meyer[3], and Renata Wassermann[4]

1 **University College London, GB,** anthony.hunter@ucl.ac.uk
2 **TU Dortmund, DE,** gabriele.kern-isberner@cs.tu-dortmund.de
3 **University of Cape Town, ZA,** tmeyer@cs.uct.ac.za
4 **University of Sao Paulo, BR,** renata@ime.usp.br

Abstract

This report documents the program and the outcomes of Dagstuhl Perspectives Workshop 19072 "The Role of Non-monotonic Reasoning in Future Development of Artificial Intelligence". The workshop brought together researchers both from core topics and peripheral areas of non-monotonic reasoning (NMR), but also attracted researchers from other scientific domains in which recent developments have shown an increased relevance of NMR topics. The overall goal of this workshop was to reshape NMR as a core methodology for artificial intelligence being able to meet present and future challenges. Participants of this workshop discussed in what shape NMR would be useful for future AI, and how NMR can be developed for those requirements. The workshop started with brief survey talks and had some technical talks on central topics of NMR afterwards. These were followed by working groups on core aspects of NMR and potential links with learning. On the last day of the seminar, each working group presented their ideas and future plans. The workshop closed with a plenary discussion on the future of NMR.

Seminar February 10–15, 2019 – http://www.dagstuhl.de/19072

2012 ACM Subject Classification Computing methodologies → Artificial intelligence, Computing methodologies → Knowledge representation and reasoning, Computing methodologies → Nonmonotonic, default reasoning and belief revision, Computing methodologies → Probabilistic reasoning, Computing methodologies → Logic programming and answer set programming, Computing methodologies → Ontology engineering, Computing methodologies → Cognitive science, Computing methodologies → Machine learning

Keywords and phrases Artificial intelligence, Knowledge representation and reasoning, Non-monotonic, default reasoning and belief revision, Probabilistic reasoning, Logic programming and answer set programming, Ontology engineering, Cognitive science, Machine learning

Digital Object Identifier 10.4230/DagRep.9.2.73

Edited in cooperation with Ivan José Varzinczak

The Role of Non-monotonic Reasoning in Future Development of Artificial Intelligence, *Dagstuhl Reports*, Vol. 9, Issue 2, pp. 73–90
Editors: Anthony Hunter, Gabriele Kern-Isberner, Thomas Meyer, and Renata Wassermann
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Anthony Hunter (University College London, GB)
Gabriele Kern-Isberner (TU Dortmund, DE)
Thomas Meyer (University of Cape Town, ZA)
Renata Wassermann (University of Sao Paulo, BR)

Nonmonotonic reasoning (NMR) addresses a fundamental problem that classical logic methods in computer science encounter when modelling real-world problems: New information may not only extend previously held knowledge (this would correspond to a monotonic extension) but can drastically change knowledge in that conclusions turn out to be wrong and need to be withdrawn. Nonmonotonic phenomena are present in all areas of our everyday lives mostly due to uncertain and incomplete information, but also due to humans reasoning with restricted ressources; on the other hand, humans do very well in determining relevant contexts of reasoning, so reasoning from incomplete information only may well be on purpose and for sake of efficiency. Nowadays, with computer systems taking on increasingly sophisticated roles in our lives, the need for computational intelligence to be able to also reason in a nonmonotonic way becomes increasingly urgent.

The international Nonmononotonic Reasoning (NMR) workshops have provided a premier specialized forum for researchers in non-monotonic reasoning and related areas since 1984. Over the years, NMR topics and results have been developed in areas such as answer set programming, computational models of argument, and description logics for ontologies. However, research on core topics of NMR has been scattered into different subcommunities that no longer collaborate in depth on a regular basis. As a consequence, much time and effort for solving specific, but in principle similar problems is wasted, general relevance of proposed solutions is overlooked, and general methodological competence is no longer developed to the same degree as ten years ago.

This Perspectives Seminar brought together researchers both from core topics and peripheral areas of NMR, but also attracted researchers from other scientific domains in which recent developments have shown an increased relevance of NMR topics. More precisely, researchers from various subcommunities within computer science and engineering (e.g., artificial intelligence, classical and non-classical logics, machine learning, agent and multiagent systems) met in Dagstuhl, but also researchers from other disciplines like philosophy and psychology contributed to the seminar. The overall goal of this seminar was to reshape NMR as a core methodology for artificial intelligence being able to meet present and future challenges. For AI to progress from pattern recognition and machine learning to broader cognitive reasoning, it needs to have commonsense reasoning, and this in turn calls for a deeper understanding of NMR. So participants of this workshop discussed in what shape NMR would be useful for future AI, and how NMR can be developed for those requirements. We started the seminar with brief survey talks on answer set programming, belief revision, argumentation, argument mining, machine learning, conditional reasoning, description logics, as well as NMR and cognition, and had some technical talks on central topics of NMR afterwards. For the rest of the week, we had working groups on NMR and learning, NMR and cognition, engineering NMR, and commonsense reasoning. We let people freely choose which working groups they wanted to attend each day, which resulted in vivid discussions and a particularly dynamic exchange of ideas. On the last day of the seminar, each working group presented their ideas and future plans, and we closed this seminar with a plenary discussion on the future of NMR. This report shows brief summaries of the presentations and of the results of the working groups.

2 Table of Contents

3 Overview of Talks

3.1 Conditional Approaches in NMR

James P. Delgrande (Simon Fraser University – Burnaby, CA)

A *conditional approach* in NMR is one that is intended to *model* a *defeasible* statement such as "birds fly". This is in contrast with most other approaches (such as default logic or circumscription) where the goal is to provide a nonmonotonic inference *mechanism*, in which a default could be encoded. So in a conditonal approach, defaults are objects that one can reason about. There are two broad approaches, nonmonotonic consequence relations, in which Gentzen-style rules for an operator $\mid\!\sim$ are specified, and conditional logic, in which a *binary modal operator* $\Rightarrow$ is developed.

3.1.1 Nonmonotonic Consequence Relations

The overall goal in these approaches is to develop a nonmonotonic consequence relation $\mid\!\sim$. This might be in order to study the properties of other formalisms or to directly study nonmonotonc inference principle. The emphasis in either case is on syntactic considerations, and axiomatic systems are the main object of interest.

Two types of relations have received significant attention, *preferential* and *rational* systems. Preferential systems are characterised as follows:

Ref: $\phi \mid\!\sim \phi$
LLE: If $\models \phi \equiv \psi$ and $\phi \mid\!\sim \gamma$ then $\psi \mid\!\sim \gamma$.
RW: If $\models \psi \supset \gamma$ and $\phi \mid\!\sim \psi$ then $\phi \mid\!\sim \gamma$.
And: If $\phi \mid\!\sim \psi$ and $\phi \mid\!\sim \gamma$ then $\phi \mid\!\sim \psi \wedge \gamma$
OR: If $\phi \mid\!\sim \gamma$ and $\psi \mid\!\sim \gamma$ then $\phi \vee \psi \mid\!\sim \gamma$
CM: If $\phi \mid\!\sim \psi$ and $\phi \mid\!\sim \gamma$ then $\phi \wedge \psi \mid\!\sim \gamma$

For rational systems, the following is added:
RM $\phi \mid\!\sim \gamma$ and $\phi \not\mid\!\sim \neg\psi$ then $\phi \wedge \psi \mid\!\sim \gamma$

3.1.2 Conditional logics

A conditional logic is a modal logic based on a binay modal operator $\Rightarrow$. The idea is to define in a Kripke structure what's meant by one world being *at least as normal as* another. For this, one can denote a (universal) accessibility relation on worlds by $\leq$, where $w_1 \leq w_2$ just if w_1 is at least as normal as w_2. Then in the two main approaches, $\leq$ is required to be either a partial or a total preorder. In either case, satisfaction in a model M at a world w is defined by:

$$M, w \models \gamma \Rightarrow \phi \quad \text{iff} \quad \text{if} \quad [\gamma] \neq \emptyset \ \text{ then } \ \min_{\leq}([\gamma]) \subseteq [\phi]$$

The resulting logics can be axiomatised as consisting of propositional logic along with:

ID: $\phi \Rightarrow \phi$
CC: $(\phi \Rightarrow \psi \wedge \phi \Rightarrow \gamma) \supset (\phi \Rightarrow \psi \wedge \gamma)$
RT: $\phi \Rightarrow \psi \supset (\phi \wedge \psi \Rightarrow \gamma \ \supset \ \phi \Rightarrow \gamma)$
CC': $(\phi \Rightarrow \gamma \wedge \psi \Rightarrow \gamma) \supset (\phi \vee \psi \Rightarrow \gamma)$

CM: $(\phi \Rightarrow \psi) \supset (\phi \Rightarrow \gamma \;\supset\; \phi \wedge \psi \Rightarrow \gamma)$
RCM: From $\psi \supset \gamma$ infer $\phi \Rightarrow \psi \;\supset\; \phi \Rightarrow \gamma$.

or the above along with the axiom:

CV: $(\phi \Rightarrow \gamma) \supset (\neg(\phi \Rightarrow \neg\psi) \;\supset\; \phi \wedge \psi \Rightarrow \gamma)$

The logics are sound and complete with respect to a partial preorder or total preorder for $\leq$, respectively. As well, preferential and rational consequence relations are interdefinable with conditional logics. However, these approaches do not handle irrelevant properties, for example $B \Rightarrow F \not\vdash B \wedge G \Rightarrow F$. This led to the *rational closure*; intuitively the formulas in a theory are ranked under the assumption that they are as normal as possible. The idea is that if there is no reason for a property to be relevant to a conditional, it is assumed to be irrelevant. In the approach, one obtains for example that a green bird flys. However the rational closure has problematic properties. Essentially, if an individual is exceptional in some way, then *all* other normality assertions are also inapplicable, and so nothing more can be concluded about that individual. While some solutions have been proposed (like the *lexicographic closure*), none appears to be wholly adequate.

3.1.3 Other Issues

These approaches have been extended and explored, e.g. to give probabilistic conditionals and to explore theorem proving. To date, approaches have focussed almost exclusively on normality conditionals; it would seem that there could be alternative approaches, dealing with e.g. deontic, counterfactual, or likelihood conditionals. As well, the problem of nested conditionals remains unaddressed, and first-order reasoning remains a challenge.

References

1. C. Boutilier. Conditional logics of normality: A modal approach. *Artificial Intelligence Journal*, 68(1):87–154, 1994.
2. James Delgrande. What's in a default? In *Nonmonotonic Reasoning: Essays Celebrating its 30th Anniversary*. College Publications, 2011.
3. S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence Journal*, 44(1-2):167–207, 1990.
4. D. Lehmann and M. Magidor. What does a conditional knowledge base entail? *Artificial Intelligence Journal*, 55(1):1–60, 1992.
5. D. Makinson. General theory of cumulative inference. In *Proc. of the Second International Workshop on Non-Monotonic Reasoning*, volume 346 of *Lecture Notes in Artificial Intelligence*, pages 1–18. Springer Verlag, 1989.

3.2 Tutorial on Probabilistic Logic Programming

Luc De Raedt (KU Leuven, BE)

Joint work of Luc De Raedt, Angelika Kimmig

Probabilistic programming is emerging as a new paradigm for programming in which one combines the power of probabilistic modeling and learning with that of programming languages. The talk presented an introduction to this topic with a focus on probabilistic logic programming languages [2]. Probabilistic logic programming is closely related and extends probabilistic databases, and as such also fits the statistical relational artificial intelligence paradigm [1]. Throughout the tutorial we used the language ProbLog [3] to illustrate the key concepts. ProbLog is based on Sato's distribution semantics [5], which essentially assigns probabilities to ground facts, which then, together with the logic programs, defines a probability distribution over possible worlds. Various applications of probabilistic logic programming were presented, and also the newest results on the integration of ProbLog with neural networks, resulting in a framework for neuro-symbolic computation [4].

This tutorial was based on joint work with Angelika Kimmig.

References

1 Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

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

3 Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *Theory and Practice of Logic Programming*, 15(3):358–401, 2015.

4 Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. In *NeurIPS*, 2018.

5 T. Sato. A statistical learning method for logic programs with distribution semantics. In L. Sterling, editor, *Proceedings of the 12th International Conference on Logic Programming*, pages 715–729. MIT Press, 1995.

3.3 The Role of Answer Set Programs for Non-Monotonic Reasoning

Thomas Eiter (TU Wien, AT)

Answer Set Programming (ASP) is a well-known problem solving approach which has roots in Knowledge Representation, Databases & Logic Programming, and Non-Monotonic Reasoning (NMR). The term was coined by Lifschitz [13, 14] and the paradigm proposed by others at about the same time [15, 16]. At an abstract level, ASP relates to Satisfiability Solving (SAT) and Constraint Programming (CP), but has some distinctive features. A rich literature is available, with books, [1, 7, 9], handbook articles [8, 4], broad surveys [2], and special issues of journals, e.g. *AI Magazine* [3] and *KI Zeitschrift* [17], on ASP. Furthermore, numerous talks[1] and tutorials on ASP, e.g. [5, 10, 12, 6] are available.[2] In this talk, we review possible reasons why ASP has gained popularity (if at all), and which factors have supported this and which are still impediments. We then consider the role of applications and implementations for the development of ASP, starting out with a look at the history of ASP. We then present some lessons learned, among them

- that theory use might differ from the original intention;

[1] Historical reflections on ASP, relevant here: http://www.cs.uky.edu/~mirek/stuff/kr-2018-gm.pdf
[2] See also the Potsdam ASP course: http://potassco.sourceforge.net/teaching.html

- that implementations and systems are a chicken-egg problem, yet even draft and imperfect implementations are vital to push research;
- that theory and applications are the two legs on which any area in the computing sciences has to stand upon;
- and that community building is essential.

We then discuss whether ASP could be a role model for NMR, where we critically review the current recognition of NMR.

Finally, we consider possible usages of ASP for NMR in the future, and we also address the issue of the role of NMR for ASP in its future development.

References

1 Baral, C.: Knowledge Representation, Reasoning and Declarative Problem Solving. Cambridge University Press (2002)

2 Brewka, G., Eiter, T., Truszczyński, M.: Answer set programming at a glance. Communications of the ACM 54(12), 92–103 (2011), http://dx.doi.org/10.1145/2043174.2043195

3 Brewka, G., Eiter, T., Truszczynski, M.: Answer set programming: An introduction to the special issue. AI Magazine 37(3), 5–6 (2016), http://www.aaai.org/ojs/index.php/aimagazine/article/view/2669

4 Dovier, A., Formisano, A., Pontelli, E.: Parallel answer set programming. In: Hamadi, Y., Sais, L. (eds.) Handbook of Parallel Constraint Reasoning., pp. 237–282. Springer (2018), https://doi.org/10.1007/978-3-319-63516-3_7

5 Eiter, T., Ianni, G., Krennwallner, T.: Answer set programming: A primer. In: Tessaris, S., Franconi, E., Eiter, T., Gutiérrez, C., Handschuh, S., Rousset, M., Schmidt, R.A. (eds.) Reasoning Web. Semantic Technologies for Information Systems, 5th International Summer School 2009, Brixen-Bressanone, Italy, August 30 – September 4, 2009, Tutorial Lectures. LNCS 5689, pp. 40–110. Springer (2009), https://doi.org/10.1007/978-3-642-03754-2_2

6 Eiter, T., Kaminski, T., Redl, C., Schüller, P., Weinzierl, A.: Answer set programming with external source access. In: Ianni et al. [11], pp. 204–275, https://doi.org/10.1007/978-3-319-61033-7_7

7 Gebser, M., Kaminski, R., Kaufmann, B., Schaub, T.: Answer Set Solving in Practice. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2012), http://dx.doi.org/10.2200/S00457ED1V01Y201211AIM019

8 Gelfond, M.: Answer sets. In: van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, chap. 7, pp. 285–316. Elsevier (2007)

9 Gelfond, M., Kahl, Y. (eds.): Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press (2014), http://dx.doi.org/10.1007/978-0-387-30164-8

10 Hölldobler, S., Schweizer, L.: Answer set programming and CLASP – A tutorial. In: Hölldobler, S., Malikov, A., Wernhard, C. (eds.) Proc. Young Scientists' International Workshop on Trends in Information Processing (YSIP). CEUR Workshop Proc. 1145, pp. 77–95. CEUR-WS.org (2014), http://ceur-ws.org/Vol-1145/tutorial1.pdf

11 Ianni, G., Lembo, D., Bertossi, L.E., Faber, W., Glimm, B., Gottlob, G., Staab, S. (eds.): Reasoning Web. Semantic Interoperability on the Web – 13th International Summer School 2017, London, UK, July 7-11, 2017, Tutorial Lectures, LNCS 10370. Springer (2017), https://doi.org/10.1007/978-3-319-61033-7

12 Kaminski, R., Schaub, T., Wanko, P.: A tutorial on hybrid answer set solving with clingo. In: Ianni et al. [11], pp. 167–203, https://doi.org/10.1007/978-3-319-61033-7_6

13 Lifschitz, V.: Answer set planning. In: ICLP. pp. 23–37 (1999)

14 Lifschitz, V.: Answer Set Programming and Plan Generation. Artificial Intelligence 138, 39–54 (2002)

15 Marek, V. W., Truszczyński, M.: Stable Models and an Alternative Logic Programming Paradigm. In: Apt, K., Marek, V. W., Truszczyński, M., Warren, D.S. (eds.) The Logic Programming Paradigm – A 25-Year Perspective, pp. 375–398. Springer (1999)
16 Niemelä, I.: Logic Programming with Stable Model Semantics as Constraint Programming Paradigm. Annals of Mathematics and Artificial Intelligence 25(3–4), 241–273 (1999)
17 Schaub, T., Woltran, S.: Special issue on answer set programming. KI 32(2-3), 101–103 (2018), https://doi.org/10.1007/s13218-018-0554-8

3.4 Reasoning about Exceptions in DL Ontologies

Laura Giordano (University of Eastern Piedmont – Alessandria, IT)

The study of nonmonotonic extensions of Description Logics is motivated by a problem in standard ontology languages (and, specifically, in OWL [1]) where a class inherits the properties of its superclasses, and where the treatment of exceptions is required in many application domains, from those concerning laws and regulations (where new laws override old ones) to medical ontologies.

The presentation is a short survey of the main approaches to reasoning about exceptions in Description Logics (DLs). Many non-monotonic extensions of DLs have been developed incorporating non-monotonic features from most of the non-monotonic formalisms in the literature, from default and autoepistemic logics, to circumscription and preferential logics, including also the approaches based on Answer Set Programming and, in general, on rule languages.

The landscape is very rich and the complexity of the various approaches has been studied, both for low complexity description logics and for high complexity description logics. The case of Description logic is an interesting case study for non-monotonic reasoning, which encompasses a limited treatment of non-monotonic reasoning in first order logic, namely, the treatment of the decidable fragment including only unary and binary predicates.

For highly expressive description logics, tractable constructions are especially important. Among the approaches, the rational closure has a polynomial construction, and it has been adapted to DLs [3], but it suffers from the well known problem called by Pearl [4] “the blocking of property inheritance problem". Refinements of his construction, avoiding this problem, have been presented in the literature, the prominent one being the lexicographic closure. The presentation compares these approaches, through examples, with other proposals, such as the logic of overriding [2], which has been defined in the setting of DLs, pointing at the achievements and open issues.

References

1 P. F. Patel-Schneider, P. H. Hayes, and I. Horrocks. OWL Web Ontology Language; Semantics and Abstract Syntax. In http://www.w3.org/TR/owl-semantics/, 2002.
2 P. A. Bonatti, M. Faella, I. Petrova, and L. Sauro. A new semantics for overriding in description logics. *Artif. Intell.*, 222:1–48, 2015.
3 G. Casini and U. Straccia. Rational Closure for Defeasible Description Logics. In T. Janhunen and I. Niemelä, editors, *Proceedings of the 12th European Conference on Logics in Artificial Intelligence (JELIA 2010)*, volume 6341 of *Lecture Notes in Artificial Intelligence*, pages 77–90, Helsinki, Finland, September 2010. Springer.

4 J. Pearl. System Z: A natural ordering of defaults with tractable applications to non-monotonic reasoning. In R. Parikh, editor, *TARK (3rd Conference on Theoretical Aspects of Reasoning about Knowledge)*, pages 121–135, Pacific Grove, CA, USA, 1990. Morgan Kaufmann.

3.5 Logics and Human Reasoning: What do we Know?

Marco Ragni (Universität Freiburg, DE)

Classical monotonic logics such as propositional or first-order logic have been considered the normative framework in psychology and cognitive science for evaluating human inferences. Core psychological results like the Wason Selection Task or the Suppression Task demonstrate that human common sense reasoning systematically deviates from classical logical reasoning. This has recently lead to a shift towards probabilistic or heuristic modeling approaches.

From a computational perspective, we can observe that recent cognitive theories have neither been formalized nor systematically analyzed or optimized. From this starting point, I will first introduce core findings and theories from cognitive science. Second, by applying methods from the AI field knowledge representation and reasoning and mathematical psychology I will analyze existing cognitive theories. As an example, I will present a reanalysis of the most prominent cognitive theories for syllogistic reasoning and show that any existing cognitive theory (including the probabilistic and heuristic theories) significantly deviates from the empirical data from psychological experiments. Hence, cognitive science still needs better theories in the sense of a better fitting (and predicting) empirical data. As a consequence, I will demonstrate that methods based on AI approaches can contribute to develop better cognitive theories. A discussion of the important role of non-monotonic logics concludes the presentation.

3.6 Argumentation Systems: A Brief Glimpse

Guillermo R. Simari (National University of the South – Bahía Blanca, AR)

Giving a knowledge repository, a natural way of finding a secure footing for the conclusions that could be obtained from it is arguing; particularly, when conflicting outcomes are reached arguing about which is to be supported is a rational way of handling such dispute. The process of arguing, and the very nature of an argument, have been the point of in-depth analysis in Philosophy since ancient times (see for instance [8, 12]) or more recently by proposing concrete models for arguments [16, 17, 14, 18]. Furthermore, the very nature of the discipline of Logic comes from the effort to clarify the presentation and exchange of arguments [15, 9]. Recently, in the field of Artificial Intelligence where many disciplines participate in the task of elucidating the essence of reasoning, research on computational argumentation has been expanding, giving birth to a field that is both exciting and fecund. In this presentation, we provide a glance at some of the ideas that are important in the field. We start with a description of the mechanics of argumentation as a procedure that could be regarded as a confrontation of postures regarding a claim. This activity can take the form, in a dialogical view, of an exchange of arguments between two participants where an arbiter decides over the initial claim considered in the debate, or in a monological description, as an internal debate where an agent performs all the roles itself, including that of the arbiter.

Describing an argumentation system requires the specification of different parts [11]. The starting point is to introduce a repository of beliefs that are represented in a formal language constituting the initial component of the system. Then, it becomes necessary to provide explicit rules for the construction from the belief base of an argument for a claim. An inference mechanism associated with the belief base usually will provide the reasoning that links the claim and the beliefs that act as premises. The next three stages address the problem of how arguments interact, formally defining the conflict between arguments, the comparison criteria between arguments, and how defeat is decided between conflicting arguments.

As the extensive range of topics explored in the literature precludes a full exploration of today's computational argumentation, we will limit ourselves to provide a concise foundation for further discussion, giving a short presentation of abstract argumentation frameworks [7] and structured argumentation systems [4]. For the interested reader, several references to general works are provided below. Finally, we will succinctly describe the international meetings, and some of the initiatives carried out in the field.

References

1. Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Christopher Reed, Guillermo R. Simari, Matthias Thimm, and Serena Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.
2. Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10-15):619–641, 2007.
3. Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors. *Handbook of Formal Argumentation.* College Publications, 2018.
4. Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.
5. Philippe Besnard and Anthony Hunter. Argumentation based on classical logic. In Iyad Rahwan and Guillermo Ricardo Simari, editors, *Argumentation in Artificial Intelligence*, pages 133–152. Springer, 2009.
6. Carlos I. Chesñevar, Ana G. Maguitman, and Ronald P. Loui. Logical models of argument. *ACM Computing Surveys*, 32(4):337–383, 2000.
7. Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.
8. Charles Griswold. Plato on rhetoric and poetry. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy.* Fall 2009 edition, 2009.
9. Dov M. Gabbay and John Woods, editors. *Greek, Indian and Arabic Logic*, volume 1 of *Handbook of the History of Logic.* Elsevier, The Netherlands, 2004.
10. Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos I. Chesñevar, Wolfgang Dvořák, Marcelo A. Falappa, Xiuyi Fan, Sarah A. Gaggl, Alejandro J García, et al. The added value of argumentation. In *Agreement Technologies*, pages 357–403. Springer, 2013.
11. Henry Prakken and Gerard Vreeswijk. Logics for defeasible argumentation. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 4, pages 218–319. Kluwer Academic Pub., 2002.

12 Christof Rapp. Aristotle's rhetoric. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2010 edition, 2010.
13 Iyad Rahwan and Guillermo Ricardo Simari. *Argumentation in Artificial Intelligence*. Springer, 2009.
14 Michael Scriven. *Reasoning*. McGraw-Hill Book Company, 1976.
15 Stewart Shapiro and Teresa Kouri Kissel. Classical logic. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2018 edition, 2018.
16 Stephen E. Toulmin. *The Uses of Argument*. Cambridge University Press, 1958.
17 Stephen E. Toulmin. *The Uses of Argument. Updated Edition*. Cambridge University Press, 2003.
18 F. H. van Eemeren and R. Grootendorst. *Argumentation, communication, and fallacies: A pragma-dialectical perspective*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

3.7 Argument Mining: from Non-Monotonic Reasoning to Natural Language Processing and Back

Serena Villata (Laboratoire I3S – Sophia Antipolis, FR)

Argument(ation) Mining (AM) [1, 2] is a recent research area in Artificial Intelligence (AI), mainly across the standard areas of Knowledge Representation and Reasoning (KRR) on the one side, and Natural Language Processing (NLP), on the other side. Few approaches to what is now called argument mining started to appear around 2010, when the first methods to mine (different connotations of) *arguments* from natural language documents were proposed: [4] introduced the definition of argumentative zoning for scientific articles, and [3] proposed a way to detect arguments from legal texts. Since these seminal approaches, the need for automated methods to mine arguments and the relations among them from natural language text was brought to light, but it was only briefly touched upon. The parallel advances, from the formal point of view in the research field of computational models of argument, and from the point of view of the computational techniques for learning and understanding human language content in the NLP and the Machine Learning fields, boosted the almost contemporary organization of two events in 2014 targeting open discussions about the challenge of mining arguments from text.[3]. Since then, AM has became a topic in major AI and NLP conferences.

Argument mining involves several research areas from the AI panorama: NLP provides the methods to process natural language text, to identify the arguments and their components (i.e., premises and claims) in texts and to predict the relations among such arguments, KRR contributes with the reasoning capabilities upon the retrieved arguments and relations so that, for instance, fallacies and inconsistencies can be automatically identified in such texts, and Human-Computer Interaction guides the design of good human-computer digital argument-based supportive tools. The argument mining pipeline is composed of two main steps: first, the argument components are identified in the text (i.e., premises and claims),

[3] The workshop on Argument Mining (https://goo.gl/kF4Eep) co-located with ACL, and the workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing (https://goo.gl/ttVUZk)

and second, the relations between arguments (e.g., support and attack) are predicted. Usually supervised learning methods are used to face these tasks, leading to the need of defining beforehand annotated datasets for the specific task and application scenario.

In addition, AM is strongly connected with hot topics in AI, as deep learning (heavily used in AM), fact checking (the prediction of the attacks between arguments is a building block for fake news detection), explanations of machine decisions (AM can disclose how the information on which the machine relies to make its own decisions is retrieved), medicine (where AM can detect information needed to reason upon randomised clinical trials), politics (where AM can provide the means to automatically identify fallacies and unfair propaganda), and for cyberbullism prevention (where AM can support the detection of repeated attacks against a person).[4]

References

1. Elena Cabrio and Serena Villata. Five years of argument mining: a data-driven analysis. In Jérôme Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 5427–5433. ijcai.org, 2018.
2. Marco Lippi and Paolo Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Techn.*, 16(2):10, 2016.
3. Rachele Mochales and Marie-Francine Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.
4. Simone Teufel, Advaith Siddharthan, and Colin Batchelor. Towards domain-independent argumentative zoning: Evidence from chemistry and computational linguistics. In *EMNLP*, pages 1493–1502, 2009.

4 Working groups

4.1 Working Group on Commonsense Reasoning

Anthony Hunter (University College London, GB)

Commonsense is an innate ability of humans, and appears to be critically important for us to operate individually and collectively in the world. Wikipedia defines commonsense as, “the basic ability to perceive, understand, and judge things that are shared by (‘common to’) nearly all people and can reasonably be expected of nearly all people without need for debate”. To help us understand the ubiquity and complexity of commonsense reasoning, we can consider three spheres of human activity where commonsense reasoning is important: Physics – General understanding of how the physical world works (naive physics), e.g. explaining why a vase breaks when dropped on the floor; Psychology – Basic understanding of human motives and behaviors (i.e., a theory of mind), e.g. explaining emotions of a colleague who is unhappy when they get a paper rejected; Society – Basic knowledge about how people can operate in societies, e.g. process to pay for a meal, how to get a credit card, etc. If we are to develop more capable and robust artificial intelligence, then we need to incorporate commonsense reasoning. Current challenging problems in artificial intelligence research such

[4] http://creep-project.eu

as explainability in machine learning, and deeper semantic understanding in natural language processing, are just two examples where commonsense reasoning is required. Commonsense reasoning has been a key driver of research in nonmonotonic reasoning, and going forward, it will be valuable to further develop and apply this research in leading edge problems in areas such as machine learning, machine vision, planning, and natural language understanding.

4.2 Working Group on Integrating ML and KR, and the Relevance of Prototypical Reasoning

Gabriele Kern-Isberner (TU Dortmund, DE)

Participants: Zied Bouraoui, Giovanni Casini, Célia da Costa Pereira, Luc de Raedt, Eduardo Fermé, Gabriele Kern-Isberner, Ken Satoh, Serena Villata

The discussions in this working group focussed on two issues:

- How can the fields of argument mining, NLP (natural language processing), machine learning (ML), and NMR can benefit from one another?
- Links between ML and NMR (or knowledge representation and reasoning (KR) in general) have long been discussed, but obviously, we are still far from successful overall integration. What could be the next steps to promote collaborations? What challenges are interesting and rewarding for both areas?

First, we present plans for combining argument mining, NLP, and NMR, and then set up a roadmap for promoting the integration of ML and NMR.

4.2.1 Plans for Combining Argument Mining, NLP, and NMR

Argument mining extracts arguments from text the basic building blocks of which are often rules. Besides feeding these arguments into argumentative systems, the extracted rules could also be used to build up knowledge bases for NMR systems. NMR inferences can then be compared to inferences obtained via argumentation. Actually, knowledge bases are of crucial importance for real-world applications of NMR/KR, therefore argument mining groups could expand its scope to collaborate with NMR groups, and NMR groups (in particular, those focussing on inductive reasoning from knowledge bases) could apply their approaches to larger examples. Benchmarks may arise from this. On the more theoretical side, more material for elaborating on the differences/connections between argumentation and NMR would be available.

On the argument mining/NLP side, textual inference is an important topic. It seems obvious that NMR methods can be useful to formalize and compute such inferences. However, first the principles underlying textual inferences have to be investigated and better understood on the NMR side.

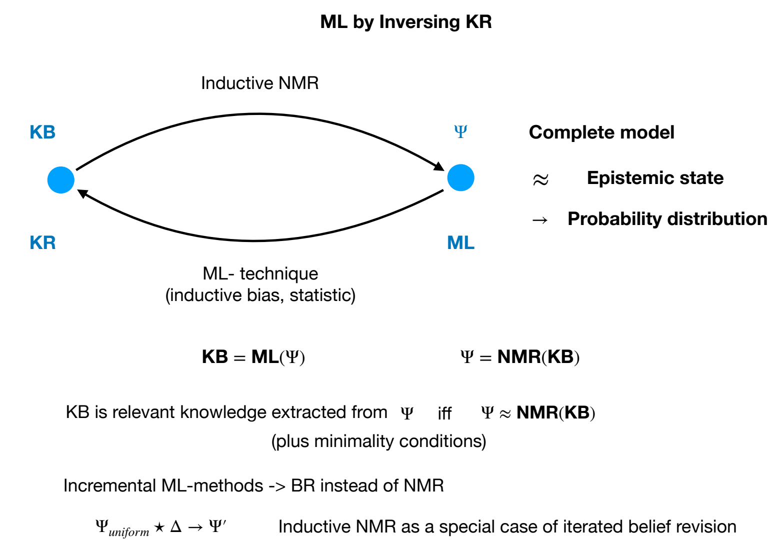
NMR and NLP need a better interface between them. A joint Dagstuhl-like workshop would be a good starting point for this. The plan is to set up an application for a Dagstuhl workshop, or to apply somewhere else (e.g., Madeira, Cape Town, Bertinoro, Leibniz-Center in NL).

4.2.2 ML and NMR – a Possible Roadmap

The working group discussed crucial challenges and innovative views that are interesting and beneficial both for ML and NMR, and apt to promote integrative approaches:

- Given that KR needs knowledge (bases) for real-world applications (but also for good toy and benchmark examples), extracting (qualitative) knowledge bases from data seems to be a major challenge. Inductive logic programming (ILP) and its extensions to answer set programming is a promising approach, but more algorithms to mine default rules from data that can be used in other NMR approaches are urgently needed. Only few approaches to that exist to date.
- ML algorithms often mine too much knowledge from data so that users are swamped by (e.g.) too many rules (when mining association rules from data). Here, formal NMR methods can help to compactify the set of mined rules so that a useful knowledge base results in the end.
- Explainable AI is broadly discussed currently. Can ML algorithms be explained in terms of NMR? Can we enhance ML methods in general by integrating (NM) reasoning? The other way round, is NMR a suitable context for ML? Which kind of reasoning/NMR approach should we use for that? Will reasoning improve the interpretability of ML results beyond explainability? How can strict rules be used to increase performance of ML algorithms? How can domain and contextual knowledge help interpreting the prediction of an ML model? Benchmark examples for comparing pure ML techniques and KR-enhanced techniques would be very useful here.
- ML usually makes use of statistical measures to validate their results and base decisions upon. Can the concept of plausibility help us to find better decisions?
- Each ML approach needs an inductive bias, how can this be related to NMR inductive reasoning?
- The role of prototypes (stereotypes) and their dynamics should be better explored in NMR and belief revision (BR), and relevance of prototypical reasoning for ML should be investigated.

A high-level vision on the connection between ML and NMR/KR is sketched in the picture below.



4.3 Working Group on Implications for NMR and Cognition

Thomas Meyer (University of Cape Town, ZA)

Participants on the first day of discussion: Arina Britz, Thomas Meyer, Abhaya Nayak, Marco Ragni, Hans Rott.

The number of participants went up significantly on the second day of the working group discussion, indicating substantial interest in this topic.

4.3.1 Topics of Discussion

In the working group we focused on two sub-topics:
1. Implications of NMR for Philosophy of Mind/Philosophy of Science (and vice versa)
2. Implications of NMR for cognition (and vice versa)

- **Philosophy of Mind:** The consensus was that this falls under Cognition, and should therefore be included under the discussions related to Cognition.
- **Philosophy of Science:** Current approaches in this field are mostly Bayesian in nature. There is a need to emphasise the advantages of the qualitative approaches in NMR. Any work on establishing links with NMR at present will depend heavily on Philosophers of Science recognising the need for this. At present it is not clear that there is such a need from this community.
- **Cognition:** The situation is different when it comes to links between NMR and Cognition. In this case there is clear interest in investigating the connections between the two areas from both communities. The primary benefits for the Cognition community relate to the need for formal systems to provide clarity. For the NMR community, the main benefits relate to the availability of a suite of benchmarks against which to test formal theories, as well as the potential for running experiments: a methodology that is well understood in the Cognition community, but not often employed in the NMR community.

4.3.2 Discussion Points

Because of the interest in Cognition, the more detailed points of discussion centred exclusively on this sub-topic.
- **Descriptive vs. Normative:** There is general agreement that it is important to investigate and model both descriptive and normative aspects of NMR with respect to Cognition. That is, we should investigate how people ought to reason in a non-monotonic context, as well as how they actually reason in such contexts. Building on that, there is interest in teasing out the difference between errors in reasoning on the one hand, and (sound) reasoning patterns that simply differ from what is expected, on the other hand.
- **Common Forms of Mistakes:** The goal here is to focus on systems that reason correctly (plausibly), that can point out mistakes by humans and correct them, and that can predict and explain mistakes made by humans. The question arises whether these processes can be automated. One possible avenue of investigation is whether this type of automation could benefit from machine learning.
- **Benchmark problems:** There is a clear need to update the 1990 list of NMR benchmark problems set up by Vladimir Lifschitz. It was also noted that, while the existing benchmark

problems seem to be well-designed for concrete instances of rules and reasoning patterns, they are not suitable for more abstract ones, such as those encountered in deontic reasoning, for example.

- **Explainability vs Rationality:** An important discussion point that was raised is the one of explainability vs. rationality. One of the driving forces, and indeed, one of the strongest selling points of NMR, is to design and build systems exhibiting rational behaviour that can be explicitly explained. But it needs to be borne in mind that explainability doesn't necessarily imply rationality. A simple example of this would be laws of nature, which are explainable, but not rational.

4.3.3 Next Steps

- **Follow-up seminar within a year:** There needs to be a follow-up seminar within a year to keep the momentum going. The focus of the follow-up seminar should be the link between NMR and Cognition. An important component of this seminar should be a series of tutorials (on NMR and Cognition) aimed at educating members of the two communities about the state of the art in the other community. Next steps after the follow-up seminar could be an event at NMR 2020 (in Patras, Greece) and a Dagstuhl Seminar in the longer run.
- **Collaborative Site:** There is an urgent need for a collaborative site to be established for this sub-topic. The immediate need is for a repository and a collaborative platform. In the short term it was proposed to use the Dagstuhl Wiki for this. In the longer term it should be hosted on an NMR site, which is to be established. The establishment of the NMR site is a point that should be taken up with the new chairs of the NMR Steering Committee, Gabriele Kern-Isberner and Renata Wassermann.

4.4 Working Group on NMR Engineering

Renata Wassermann (University of Sao Paulo, BR)

While there are still many interesting theoretical issues to be solved in the realm of NMR, we feel that experimenting with real applications and having real world problems in mind can benefit the area. In this working group, the participants discussed possible strategies for increasing the visibility of existing implementations and applications, as well as the need to encourage new contributions to this "engineering" part of NMR.

A first challenge reported by those who are involved in implementations is to gather reliable benchmarks for the different branches of NMR. For some sub-areas of NMR, there are system competitions that take place regularly and could serve as an inspiration for devising benchmarks for the other areas. For example, in Answer Set Programming (ASP), the ASP Challenge (https://sites.google.com/view/aspcomp2019) happens every two years since 2007. The International Conference on Computational Models of Argument (COMMA – http://comma.csc.liv.ac.uk), features a track on system demonstrations. During the meeting, we have listed a series of other sources for inspiration:

- The Competition on Legal Information Extraction and Entailment[5] (COLIEE), which provides examples for legal reasoning.
- The International General Game Playing Competition[6] (IGGPC).
- The SAT competition[7]
- Examples from the International Workshop on Principles of Diagnosis Series[8] (DX).
- The SATLIB[9] and TPTP[10] repositories provide propositional knowledge bases for SAT problems.
- Several biomedical ontologies and knowledge bases that can be found on the web, for example, in the BioPortal[11].

It is important to remark that in order to have good benchmarks, it is not enough to collect knowledge bases, we have to generate interesting problems using them. For example, in belief revision, we must also select input sentences which give rise to interesting contraction or revision problems.

Another topic of discussion was the visibility of existing tools and applications, both for researchers within NMR, that could benefit from what colleagues are doing as for the Artificial Intelligence community, that still sees NMR as a purely theoretical area.

As short term actions that were discussed in the meeting, we can mention:

- Create a web page and a repository, possibly linked to the page of the NMR Workshop series.
- Collect for the repository existing implementations and tools, with their description.
- Collect also examples of "real world" applications.
- Collect tutorials, overview papers, slides on various aspects of NMR.
- Edit the Wikipedia pages on NMR and related topics:
  - https://en.wikipedia.org/wiki/Non-monotonic_logic
  - https://en.wikipedia.org/wiki/Defeasible_reasoning
  - https://en.wikipedia.org/wiki/Belief_revision
- Create or link repository for Knowledge Bases in propositional logic and Description Logics.
- Define guidelines for creating reasoning problems from the knowledge bases.
- Define simple forms for researchers to submit new tools or benchmarks sets to the repository.

[5] https://sites.ualberta.ca/~miyoung2/COLIEE2018
[6] http://ggp.stanford.edu/iggpc/index.php
[7] http://www.satcompetition.org
[8] https://dx-workshop.org/dx-series
[9] https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html
[10] http://www.tptp.org
[11] https://bioportal.bioontology.org

Participants

- Zied Bouraoui
Artois University – Lens, FR
- Arina Britz
University of Stellenbosch, ZA
- Giovanni Casini
University of Luxembourg, LU
- Célia da Costa Pereira
Laboratoire I3S – Sophia Antipolis, FR
- Luc De Raedt
KU Leuven, BE
- James P. Delgrande
Simon Fraser University – Burnaby, CA
- Thomas Eiter
TU Wien, AT
- Eduardo Fermé
University of Madeira – Funchal, PT
- Laura Giordano
University of Eastern Piedmont – Alessandria, IT
- Andreas Herzig
University of Toulouse, FR
- Anthony Hunter
University College London, GB
- Gabriele Kern-Isberner
TU Dortmund, DE
- Sebastien Konieczny
Artois University – Lens, FR
- Maria Vanina Martinez
University of Buenos Aires, AR
- Thomas Meyer
University of Cape Town, ZA
- Abhaya Nayak
Macquarie University – Sydney, AU
- Odile Papini
University of Marseille, FR
- Marco Ragni
Universität Freiburg, DE
- Gavin Rens
KU Leuven, BE
- Hans Rott
Universität Regensburg, DE
- Ken Satoh
National Institute of Informatics – Tokyo, JP
- Guillermo R. Simari
National University of the South – Bahía Blanca, AR
- Ivan José Varzinczak
Artois University – Lens, FR
- Serena Villata
Laboratoire I3S – Sophia Antipolis, FR
- Renata Wassermann
University of Sao Paulo, BR
- Emil Weydert
University of Luxembourg, LU

Report from Dagstuhl Seminar 19081

Verification and Synthesis of Human-Robot Interaction

Edited by

Rachid Alami[1], Kerstin I. Eder[2], Guy Hoffman[3], and Hadas Kress-Gazit[4]

1 **LAAS – Toulouse, FR,** rachid.alami@laas.fr
2 **University of Bristol, GB,** kerstin.eder@bristol.ac.uk
3 **Cornell University – Ithaca, US,** hoffman@cornell.edu
4 **Cornell University – Ithaca, US,** hadaskg@cornell.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19081 "Verification and Synthesis of Human-Robot Interaction". This seminar brought together researchers from two distinct communities – Formal Methods for Robotics, and Human-Robot Interaction – to discuss the path towards creating safe and verifiable autonomous systems that are compatible with humans.

Seminar February 17–22, 2019 – http://www.dagstuhl.de/19081
2012 ACM Subject Classification Computer systems organization → Robotic autonomy, Human-centered computing → Human computer interaction (HCI), Software and its engineering → Formal methods
Keywords and phrases Formal Methods, Human-Robot Interaction
Digital Object Identifier 10.4230/DagRep.9.2.91
Edited in cooperation with Ross A. Knepper

1 Executive Summary

Hadas Kress-Gazit (Cornell University – Ithaca, US)
Rachid Alami (LAAS – Toulouse, FR)
Kerstin I. Eder (University of Bristol, GB)
Guy Hoffman (Cornell University – Ithaca, US)

There is a growing trend in robotics moving from industrial robots that work physically separated from people to robots that collaborate and interact with people in the workplace and the home. The field of human-robot interaction (HRI) studies such interactions from the computational, design and social points of view. At the same time, there is growing interest in research regarding the safety, verification and automated synthesis of behaviors for robots and autonomous systems. The fields of formal methods and testing, which focus on verification and synthesis of systems, aim to model systems and define and prove specifications over these systems; in the context of robotics, these techniques take into account the robot dynamics and its interaction with its changing and uncertain environment.

However, a human collaborating with a robot is not just part of the robot's environment, but an autonomous agent with intentions, beliefs, and actions that mesh with those of the robotic agent. This raises new research questions related to verification and synthesis including what appropriate models for human-robot interaction would be; whether and how algorithms for HRI can enable verification; how to take the human into account in automatic synthesis of robotic systems; and what (if any) guarantees can be provided with a human in the loop.

To date, very little work has explored questions of verification, safety guarantees and automated synthesis in the context of Human-Robot Interaction. HRI has modeled humans computationally but not from a verification point of view and without providing guarantees. Furthermore, there are rarely any formal specifications in the computational HRI literature; validated objective metrics for evaluation are also scarce. The verification and synthesis community has mostly focused on the robot's autonomous behavior and its environment, and not paid much attention to the integral presence of the human or the interaction, including the psychological, social, and intentional aspects of human activity.

In this seminar we bring together experts in computational HRI, verification of autonomous systems, formal methods, and cognitive and social psychology to exchange ideas, define research directions, and foster collaborations toward a new theory and practice of verifiable HRI.

2 Table of Contents

Working groups

3 Overview of Talks

3.1 Education in HRI

Henny Admoni (Carnegie Mellon University – Pittsburgh, US)

This short talk provides an overview of the field of HRI for education. It covers research from this field broadly at an introductory level by providing examples of a selection of projects. There is an overview of the benefits, models, and challenges of HRI for education.

3.2 Computational approaches in HRI: Examples and Discussion

Rachid Alami (LAAS – Toulouse, FR) and Guy Hoffman (Cornell University – Ithaca, US)

Brief review of design objectives for the implementation of robots that engage in collaboration with human. Teamwork and joint action: collaboration and coordination. Presentation of a set of examples.

3.3 Human-Robot Handovers

Maya Cakmak (University of Washington – Seattle, US)

Handovers are an essential capability for personal robots that are intended to assist humans or collaborate with humans in different environments, such as homes, hospitals, factories. This talk characterizes the problem of human-robot handovers, identifying dimensions in which handovers can vary and ways in which we can measure the success and quality of a handover.

3.4 Two examples of how we used verification techniques in HRI

Kerstin I. Eder (University of Bristol, GB) and Dejanira Araiza-Illan (ARTC – Singapore, SG)

These slides show how we used a combination of different verification techniques to gain confidence in the correctness of autonomous systems that interact with humans, i.e. HRI. The first investigates the front-end of the system development process, where specifications are developed/translated into designs, e.g. in Simulink. The important question is whether these Simulink designs preserve the intent of the specification. This is so important because if the final coded system does not work, it is very useful to know whether this is because there is a coding bug, or whether there is a bug in the design. As such, verification should be done as early as possible, including in particular the design level. We propose the introduction of assertions, written in Simulink, and added directly to the Simulink model. Whether the design satisfies these assertions can then be determined using two techniques. Some assertions can be tested, i.e. those that are ground. Others, i.e. those that contain variables, can be checked using theorem proving, ideally automatic theorem proving techniques. In a different context, we developed a coverage-driven simulation-based verification technique (CDV) for a robot to human hand-over task. We present the different components of a CDV testbench, the System Under Test, i.e. the robotic code, the test generator, the checker and coverage collector. In particular, we show some of the requirements used to encode assertion monitors to flag issues during simulation as well as the coverage models we used. Associated papers are linked into the slides. Please contact us for further information.

3.5 A Simple Model

Rüdiger Ehlers (Clausthal-Zellefeld, DE)

During the seminar, we identified a lack of compatibility between the models of human behavior in the field of Human Robot Interaction (HRI) and the models used by the formal methods community. This short talk provided some initial ideas on how the latter type of model can be brought closer to the former one. One concrete Markov Decision Process (MDP) formulation of a scenario in which a human and a robot pass by each other in a narrow hallway was given. The MDP already captures *some* aspects from the talk given by Marc Hanheide (Section 3.6). Starting from this formulation, the idea is to add additional HRI aspects one by one. Along the way, the necessary concepts to increase the scalability of analyzing the MDPs of the particular resulting shape can be researched.

3.6 Social Navigation: Problem Statement and (Some) Solutions

Marc Hanheide (University of Lincoln, GB)

Social navigation constitutes an area of research in robot navigation, to explicitly respect social norms, to model human intent in navigation, and ultimately provide robots with the skill to safely and appropriately navigate in the presence of humans in a shared space. First, two main contexts are evident in literature: One is to avoid and circumvent humans in order to ensure safe passage and increase acceptability and perceived safety (trustworthiness) of robots for humans. Second, it studies suitable ways to approach humans to facilitate and support further interaction (e.g. dialogues, object hand-overs, and other forms of engagement). Literature reports on different approaches to accomplish these objectives, such as social cost- or utility maps, joined multi-agent planning explicitly modelling possible human behaviour in response to robots, and approaches employing computational models of games theory or explicit discrete state transitions to predict human-robot spatial interaction patterns. Some of these models lend themselves quite readily to further investigation in the context of form synthesis and verification.

3.7 HRI Specifications and Natural Language

Christoffer R. Heckman (University of Colorado – Boulder, US)

Formal human-robot interaction requires the creation of some specification for which there are many different forms. They might include specification of safety that consider some behaviors proximal to humans that can include restrictions of state space, or some more amorphous social contracts that cannot be broken. In this short talk, I consider how natural language grounded in the spatial environment might also be considered as a form of specification. On the one hand, this mode is natural and is the form in which we as humans find simplest to define instructions (e.g. "go over there and hang out for a bit"). On the other, natural language has an enormous prior information set from which it draws and is also variably dependent on the environment. I consider some techniques in machine vision and dynamical systems that have been in some sense brought to heel through the success and proliferation of data-driven techniques, but I also identified a few remaining challenges related to how one might define specifications in natural language. Finally, I give some considerations for how this might work in the future through joint vector embeddings and ontological grounding.

3.8 Iterative Design of Verifiable Human Models

Guy Hoffman (Cornell University – Ithaca, US)

I propose a human-centered process of iteratively designing formal models for human-robot interaction. The framework is inspired by iterative practices in human-centered design (HCD), often conceptualized as a cycle of four phases: observation, ideation, prototyping, and testing. I propose mapping these onto existing practices in formal methods in robotics, such as model-building, testing, and verification. In the proposed process, samples of human participants interact in human-participant studies with robots synthesized from intermediate models and specifications. Verification methods are used to test and update these models and specifications, leading to newly synthesized controllers that are then tested with an additional sample of human interactants. The proposed approach suggests a number of research questions, including how to optimally sample human interactants, and whether and how to update models from outcomes of the interaction studies.

3.9 Verification of Autonomous Robots, a Roboticist Bottom Up Approach

Felix Ingrand (LAAS – Toulouse, FR)

Complete validation and verification of the software of an autonomous robot (AR) is out of reach for now. Still, this should not prevent us from trying to verify some components and their integration. There are many approaches to consider for the V&V of AR software, e.g. write high level specifications and derive them in correct implementations, deploy and develop new or modified V&V formalisms to program robotics components, etc. We propose an approach which rely on an existing robotics specification/implementation framework (GenoM) to deploy robotics functional components, to which we harness existing well known formal V&V framework (UPPAAL, BIP, FIACRE/TINA). GenoM was originally developed by roboticists and software engineers, who wanted to clearly and precisely specify how a reusable, portable, middleware independent, functional component should be written and implemented. Many complex robotic experiments have been developed and deployed using GenoM and it is only recently that its designers realized that the rigorous specification, a clear semantic of the implementation and the template mechanism to synthesize code opens the door to automatic formal model synthesis and formal V&V (offline and online). This bottom up approach, which starts from components implementation, may be more modest than the top down ones which aim at a larger and mode global view of the problem. Yet, it gives encouraging results on real implementations on which one can build more complex high level properties to be then V&V.

3.10 Multimodal Dialog in HRI

Ross A. Knepper (Cornell University, US)

Multimodal dialog exploits natural human communicative abilities to mediate human interaction in service of a joint activity or shared task. Modalities include any combination of speech, gesture, facial expression, eye gaze, body language, and gross body motion. Dialog involves a back and forth exchange, in which previous communicative acts become context in which to interpret later ones. The multiple modalities within a single communicative act also serve as context for understanding one another, which provides redundancy and makes the problem more tractable. This presentation walks you through the problem statement for grounding multimodal acts in a symbolic basis.

3.11 Task-Agnostic HRI

Ross A. Knepper (Cornell University, US)

There is an unbounded quantity of HRI tasks and scenarios for which domain-specific details are needed to specify and model correct behavior. In contrast, there is a set of behaviors that transcend the details of task and focus on the establishment of maintenance of a cooperative team, consensus around goals and intentions within the task, and understanding of the partition between shared and individual decision making. By formalizing and verifying these behaviors, many different HRI tasks benefit indirectly.

3.12 Testing, Verification, Synthesis Tutorial

Hadas Kress-Gazit (Cornell University – Ithaca, US) and Kerstin I. Eder (University of Bristol, GB)

This tutorial provided a basic introduction to terminology used in the context of testing, verification and synthesis with example applications in HRI.

3.13 Toward a human model – assessing perception-action coupling

Ute Leonards (University of Bristol, GB) and Kerstin I. Eder (University of Bristol, GB)

A key challenge for certification, verification and validation of robots interacting with people outside factory settings is how to ensure safety for all possible situations the robot might encounter. Ultimately, human behaviour remains unpredictable as there are far too many ways people could potentially interact with robots over and above those intended and accounted for within the design process. Efforts to solve this challenge include the application of increasingly complex cognitive models in the robot to predict human behaviour, including such aspects as theory of mind and other psychological theories on human social interaction, complex learning rules and so on; models that loose flexibility and processing speed.

Instead, we propose to go back to a very basic model of human action prediction, essentially a type of sanity checks. The model is based on the observation that human beings are embodied, and every task (action) involves a motor response that is usually tightly coupled to sensory input and thus the environment it is performed in (see Gibson's affordance models, 1979). Such perception-action coupling, be it for gait, hand, eye movements or a combination of these, is predictive for any given individual. Any deviation from this basic perception-action coupling in human behaviour, e.g. to perform an action in a different way to that expected or to perform a different action, leads to a delay, i.e. noticeable hesitation, to account for the required decision making time under cognitive control and the change in motor planning. In other words, the sanity check here would be a time-critical but simple check of whether the predicted action is performed: any decision making for predefined actions occurs within a small, predefined time window and is restricted to a small number of movement alternatives. Any temporal delay means therefore that the person is most likely not performing the task the robot is expecting them to perform, or at least not in the way the robot is expecting them to (e.g. the person is distracted); and safety measures should be taken.

The clear advantage of such a basic model would be increased usability as the model is quite generic. Research into developing a basic, safety-focused model for a variety of application domains would provide us with new insights into the feasibility and limitations of this approach. Compared to the myriad of application-specific models of increasing complexity that are currently available or under development, this human-centric approach promises simpler models, flexibility of use and computational efficiency.

3.14 Robots in Therapy

Shelly Levy-Tzedek (Ben Gurion University – Beer Sheva, IL)

There is an unmet need in therapy for clinician hours. Specifically, in post-stroke rehabilitation, patients need to preform repetitive practice; but without an accompanying therapist, compliance is low. One way to fill this “care gap” is to enlist robots. There are physically assistive robots, which can help the patient perform the task by moving their impaired limb, and there are socially assistive robots, which can help people by motivating them to perform the exercise and giving them feedback. If a therapist is not present during the exercise, however, there is potential damage that can ensue. Thus, the robotic system should be able to model, and respond to, the person’s affect (e.g., work by Jamy Li), intent (e.g., work by Hennt Admoni), and motor performance (e.g., work by Shelly Levy-Tzedek).

3.15 Physical HRI

Todd Murphey (Northwestern University – Evanston, US)

This talk briefly described some of the needs and challenges associated with using robots to physically assist and train people. Several key points about how physical Human Robot Interaction differs from other kinds of HRI were made. These include that the person and robot interact with each other through forces, and these forces have both mechanical effects–e.g., they can stabilize and destabilize someone–and communication effects–they can help someone learn from physical interaction. Several examples of what verification might look like in the context of physical assistance and rehabilitation were discussed, including the need to keep someone safe while avoiding overassistance.

3.16 Formal Specification Patterns for Sanity Checking

Kristin Yvonne Rozier (Iowa State University, US)

As a community, we have identified choosing the right (safe, progressing) next-action as one of the biggest challenges in human-robot interaction; doing this sufficiently quickly adds to the challenge. Sanity checking provides a tractable answer to this challenge. We exemplify sanity checks over the mission of the ExoMars Schiaparelli Lander and then generalize the patterns we so often encode for autonomous spacecraft, aircraft, and robots. Sanity checks are unsatisfying and unintuitive: we are unaccustomed to encoding common sense in the form of requirements and unsatisfied by the lack of diagnosis they provide. The necessity of choosing a next-action in real time, within the limitations of embedded computation requires us to sacrifice the satisfaction of diagnosing why and how a particularly interesting and complex error occurred in favor of the just-in-time determination that we need to switch to "safe mode." Motivated by the goal of choosing correctly from among the small, finite set of possible next-actions any automated system can execute, sanity checks provide a promising way forward, that we can monitor and enforce on-board via runtime verification, e.g., with R2U2 (http://temporallogic.org/research/R2U2/).

3.17 Interactive Autonomy: a human-centered approach for safe interactions

Dorsa Sadigh (Stanford University, US)

Reward functions are formal specifications used to describe how a robot should act or interact with humans. Similar to specifications, coming up with reward functions can be challenging too. We would like to learn these specifications either from demonstrations or preferences. However, teleoperating robots with high degrees of freedom is quite challenging so learning reward functions from demonstrations can be limiting. Instead, we propose an approach to actively generate new scenarios and query humans in order to learn their preferences from a combination of pariwise comparisons and limited expert demonstrations.

4 Working groups

4.1 Physical Human-Robot Interaction Discussion Group

Brenna D. Argall, Kerstin I. Eder, Christopher R. Heckmann, Ute Leonards, Todd Murphey, Kristin Yvonne Rozier

This breakout group discussion focused on how and when verification techniques may be used to increase trustworthiness of a physical human-robot system. Trustworthiness is a mutual effect in physical Human-Robot Interaction (pHRI)–the human must trust the automation and the automation must trust the human (often with the automation regulating the exchange of decision authority). Parts of the system may be verifiable in a classical formal methods sense–for instance, the computing elements–but the group agreed that it seems unlikely that the human is a classically verifiable component. The discussion primarily built up a model of where and how verification methods can be applied. The resulting model included many interacting components: the human, the robot, the forceful interactions between them, the computed combined model of the human and robot, observations of the combined system, online and offline machine learning algorithms responsible for building models of the combined system, the computing elements, and physics/psychophysics that constrain possible behaviors. These interactions form a complex network of interdependent components, where each component of the network could be formally modeled and verified and the interactions between them could be modeled and verified. An important insight is that a person or multiple people can always choose to undermine a pHRI system, so guarantees of safety are necessarily limited to making sure that the rest of the automation is not responsible for failure.

4.2 An Iterative Workflow for HRI Model Repair

Frank Broz, Jan Kretinsky, Nils Jansen, Hadas Kress-Gazit, Guy Hoffman

In this breakout session, we discussed an iterative workflow enabling model repair for human-robot interaction tasks. Our approach is anchored in a probabilistic joint human-robot interaction model drawing on separate human and robot state transition graphs. The workflow includes learning initial parameters for a human model from human-human interaction datasets, and evaluating these vis-a-vis specifications drawn from the social psychology literature. A synthesized robot controller is then composed with the human model and put to test in a human participant experiment. The outcomes of this experiment is used to both study guarantees on the interaction and refine the model as part of the iterative improvement cycle. As technique for model refinement we plan to explore model repair under temporal logic constraints. We decided to begin by implementing this workflow on mutual gaze and handover tasks.

4.3 Synthesis-Aided End-User Programming of Interactive Robots

Maya Cakmak, Ivan Gavran, Jana Tumova, Aurelie Clodic, Shelly Levy-Tzedek, Laurel Riek, Hadas Kress-Gazit, Marc Hanheide, Rüdiger Ehlers

Programming interactive robots to create new applications is challenging, even for experienced software developers, due to the complexity of concurrently handling multiple input channels while generating actions across multiple modalities (speech, sounds, gesture, gaze, text on screen, facial expression, motion). Program synthesis can enable writing better programs in less time and with less prior expertise by automating parts of the programming process. However, it is currently unclear how what parts of the programming process can be automated, what user input can be captured and translated into specifications that can be used for synthesis, and what synthesis methods are appropriate. The goal of this project is to identify opportunities for applying synthesis to improve the process of programming interactive robots.

4.3.1 Specific Outcomes

- Choose domains/tasks focused on robots that socially interact with people, e.g. storytelling robot, language tutor robot, stress support robot
- Clarify what we mean by program/controller, e.g. finite state machines
- Identify prior formats of specifications, e.g. partial programs, correctness properties, interaction traces, sketches, etc
- Identify modalities for capturing user input, e.g. natural language, demonstration (in different ways), visual programming environments, text, etc
- Identify methods for translating user input to specs
- Identify methods for combining different combinations of specs to synthesize robot programs with certain properties
- Work out a running example for all of the above based on chosen tasks
- Make figures to communicate created knowledge/ideas
- Outline position/framing paper to communicate created knowledge/ideas
- Identify low-hanging novel research and discuss possible collaborative paper opportunities

4.4 A Framework for Synthesis-Aided End-User Programming of Interactive Robots

Maya Cakmak, Jana Tumova, Laurel Riek, Shelly Levy-Tzedek, Ivan Gavran, Aurelie Clodic, Hadas Kress-Gazit, Rüdiger Ehlers

Programming interactive robots to create new applications is challenging, even for experienced software developers, due to the complexity of concurrently handling multiple input channels while generating actions across multiple modalities (speech, sounds, gesture, gaze, text on screen, facial expression, motion). Program synthesis can enable writing better programs in less time and with less prior expertise by automating parts of the programming process. However, it is currently unclear how what parts of the programming process can be automated, what user input can be captured and translated into specifications that can be used for synthesis, and what synthesis methods are appropriate. The goal of this project is to identify opportunities for applying synthesis to improve the process of programming interactive robots.

4.5 HRI Skill: Social Navigation (Elaboration)

Marc Hanheide, Rüdiger Ehlers, Jana Tumova, Felix Ingrand, Kerstin I. Eder, Satoru Satake

This is an elaboration group from Tuesday with the goal of reaching more detailed and thought-through outcomes.

4.5.1 Specific Outcomes

- More detailed models, including a comparison between them
- Low hanging fruit for specific research projects
  - Simulation test generation
  - Basic proven runtime guards
- Technical and intellectual challenges
- Defining metrics for success
- Human models (approach, pass)
- Defining contexts of the area

4.6 Formalizing Flexible Collaboration in HRI

Ross A. Knepper, Morteza Lahijanian, Henny Admoni, Rachid Alami, Shanee Honig, Satoru Satake, Victor Fernández Castro

There is an unbounded quantity of HRI tasks and scenarios for which domain-specific details are needed to specify and model correct behavior. In contrast, there is a set of behaviors that transcend the details of task and focus on the establishment of maintenance of a cooperative team, consensus around goals and intentions within the task, and understanding of the partition between shared and individual decision making. By formalizing and verifying these behaviors, many different HRI tasks benefit indirectly.

4.6.1 Specific Outcomes

- Define a few simple, diverse exemplar tasks that we could implement
- Identify formalisms and models
- Provide a minimal necessary set of capabilities for a working model
- Identify a list of abstract capabilities that the system would ideally have
- Plan data collection study in each task

4.7 Model Repair for Models of Human

Jan Kretinsky, Frank Broz, Hadas Kress-Gazit, Guy Hoffman

Given a wrong model of a human a true property that does not hold in the model, how to fix the model to reflect it.

4.8 HRI Application Area: Healthcare and Therapy (Elaboration)

Jamy Jue Li, Erika Abraham, Victor Fernández Castro

This is an elaboration group from Tuesday with the goal of reaching more detailed and thought-through outcomes. Specific Outcomes:
- More detailed models, including a comparison between them
- Low-hanging fruit for specific research projects
- Technical and intellectual challenges
- Defining metrics for success
- Human models
- Defining contexts of the area

4.8.1 Rationale for HRI Therapy + Models

1. Human therapist will not remember all the details about the past because they have so many people they work with, thus a system model that tracks and adapts to many details that the human therapist simply cannot track over time will be helpful to use to adapt and personalize therapy to an individual in a way that a human cannot.
2. To maintain the therapist involvement and expertise, the therapist may be helped by receiving advice (i.e., recommendations) from the system based on information that the therapist may not remember or even perceive (such as minute but consistent changes in response times)
3. To be able to look at a wide variety of system parameters that could affect a metric of success (for example, the user's improvement in performance in an educational test), the model of the system could test many free parameters of the system that have been identified by the researcher and refined through discussion with the therapist

4.8.2 Overall Process

- *Example application overview.*
  Robot playing an activity with an autistic child where an adult therapist is leading the interaction. The adult therapist presses buttons on their tablet to initiate one of multiple activities for the robot to play, and to initiate specific actions within those games like the robot displaying a "happy" face, "sad" face, etc. in the order the therapist chooses. The child can also initiate actions on their tablet (such as them choosing for the robot to make a happy face) and also respond to the robot's questions using the tablet.

- *Data collection of interactions of users with the system*
  An example of data collection in HRI studies is 50 participants are run and interactions are videotaped, then annotated by 3 trained coders for participant affect (emotion), engagement and performance in a therapeutic task. The annotation is created by coders who view a 15 minute video clip of a child interacting with the robot and code at each 5 second time point the child's affect and the child's engagement. This timeline of codes will be matched with the time-stamped record of the system's log, which contains both the timing of the robot's actions and the timing of the human therapist's initiated actions for the robot.
- *Data collection is needed*
  What is annotated in the videos is a key consideration for the model, because it may be valuable to annotate a lot of additional variables (preferably automated annotation) that do not directly have a hypothesis around those variables because a model may be able to find a possible new relationship in the data.
- *Which types of model to use?*
  - A model for prediction is useful in this application (synthesizing a controller that is most likely to lead to success metric in the model)
  - A model for verification of the controller doesn't make much sense because it is hard to verify a human's behavior (however, it may be able to verify a limited subset of the human's behavior defined as the person's inputs with the system, which could be active input like tablet presses or passive input like smiles or eye gaze)
- *How to learn the model?*
  - Teaching-based isn't possible
  - Adaptation could be possible
- *How should the robot behave?*
  Personalization: Coupling of either two robot responses (robot responds with both a smile "you're correct" and a visual light) or two robot stimuli/prompts (delivery of both robot emotion, e.g., happy face, and another perceptual cue in tandem)

4.8.3 Operational Ideas

- What to do with the model?
  - Option 1: Find patterns in data executions In a large dataset of interactions that have been annotated, it could be possible to determine patterns in the data leading to insights of the relationship between the robot's and human's indicators/properties and the desired goal metric
  - Option 2: Give advice to therapist. In many situations where the therapist is involved in the therapy process with the technical system and their expertise is used to monitor the user's behavior and help the user, giving the therapist advice rather than autonomous adaptation (i.e., the system simply deciding what action to take) is a more acceptable strategy that respects the preference for the human therapist to always make the final call. Whether the human therapist follows the advice or not is a variable that the system can track. Advice-giving for therapists requires some level of explanation to the therapist about why each advised option is being presented in order to increase transparency to the therapist, which still being minimal cognitive load to distract from the therapist's main task of assisting the user. One way in which a graphical interface could present this to the therapist is by explaining what purpose (i.e., model strategy) each advice action corresponds to, such as: 1) system judges the success chance is highest with advised action #1; 2) system picks action #2 to improve ambiguity

in the model by testing this action; or 3) system judges action #3 will led to high success change for alternative outcome that could be desired (such as to increase child happiness) Types of advice: Parameters of therapy vs. robot behavior parameters
- Option 3: Make predictions about child behavior. In a large dataset of interactions that have been annotated, it could be possible to identify precursors to success metrics (e.g., child presses on tablet within 5 seconds) or child behaviors (e.g., child stands up and leaves) using characteristics captured by the system.
- Option 4: Optimize model learning to improve the model. In the event that the model cannot disambiguate between how two paths in the model affect the success metric, the model can be used to generate multiple test cases corresponding to each of the paths to the model. The resulting group of executions could then be used to refine the model to determine which of the paths leads to a better success metric result.

Model construction.

Variables to capture
- Order of the trials in a task (e.g., order of the emotions that the game goes through)
- Collect the parameters of robot appearance and behavior that can be varied, then run these by the therapist
- Usability test with some of the parameters
- Design of executions
- Vary parameters in multiple executions
- For example, using multiple variable time delays in an interaction with a robot between [0,25 s – 4 s] at each verbal response of the robot can provide a dataset of variable time delays to try to identify which time delay results in the best human performance (or other measure) of the system
- Another example is using multiple embodiments in the learning activity, such as a photo of a robot, a screen image of a robot, an actual robot

4.8.4 Other Considerations

Time needed to be interdisciplinary
Takes time to model data in the correct form, to collect the data, how to present the advice to the therapist.

4.9 HRI Application Area: Healthcare and Therapy

Jamy Jue Li, Erika Abraham, Victor Fernández Castro, Shelly Levy-Tzedek

In a scenario where a human therapist uses a robot to monitor and help a user (for example, in physical therapy or social skills therapy), a model could provide advice to the therapist on how the robot should behave while explaining the strategy for the advice (for example, highest predicted primary outcome, highest predicted secondary outcome or disambiguating how two potential paths in the model affect outcomes). User parameters to be collected for prediction could include order of task trials in an activity, type of trials (e.g., reach task vs push task; emotion task vs informational task), and design characteristics of the robot.

Design executions could also be autonomously generated to explore how new robot parameters (for example, latency in robot response) affect outcomes. These may be personalised per individual. A key challenge may be the time needed to construct the model, collect data and design how to present advice to the therapist.

4.10 HRI Application Area: Industrial Assembly (Elaboration)

Björn Matthias, Michael Gienger, Daniele Magazzeni, Alessandro Cimatti, Dejanira Araiza-Illan

This is an elaboration group from Tuesday with the goal of reaching more detailed and thought-through outcomes.

Specific Outcomes

- More detailed models, including a comparison between them
- Low-hanging fruit for specific research projects
- Technical and intellectual challenges
- Defining metrics for success
- Human models
- Defining contexts of the area

4.11 Verification Modeling for Physical HRI

Todd Murphey, Kristin Yvonne Rozier, Kerstin I. Eder, Brenna D. Argall, Ute Leonards, Christopher R. Heckman

This is a continuation of the Tuesday group, focusing on how and when verification techniques may be used to increase trustworthiness of a physical human-robot system. Trustworthiness is a mutual effect in physical HRI—the human must trust the automation and the automation must trust the human (often with the automation regulating the exchange of decision authority). Parts of the system may be verifiable in a classical formal methods sense, but it at least seems unlikely that the human is a classically verifiable component. Nevertheless, in the context of an otherwise formally understood system, the human model could be falsified. The purpose of this discussion will be to build up a model of where and how verification methods can be applied.

Specific Outcomes

- Data-driven modeling and its properties
- Learning and active learning of specification semantics from continuous time/space physical HRI
- Which pieces of a physical HRI system can be formally verified using theory?

- Among the pieces that can be theoretically verified based on models, how can those verified pieces be composed?
- What are a list of pHRI applications that are both important and have reasonable decompositions into analyzable pieces? Semi-rigid exoskeletons are probably an example, but soft-body exoskeletons may not be. Understanding what constitutes the division would be helpful.
- What is at least one concrete project that could be accomplished in a 2-3 year period?
- Diagram describing key needs and challenges

Participants

- Erika Abraham
RWTH Aachen, DE
- Henny Admoni
Carnegie Mellon University – Pittsburgh, US
- Rachid Alami
LAAS – Toulouse, FR
- Dejanira Araiza-Illan
ARTC – Singapore, SG
- Brenna D. Argall
Northwestern University – Evanston, US
- Frank Broz
Heriot-Watt University – Edinburgh, GB
- Maya Cakmak
University of Washington – Seattle, US
- Alessandro Cimatti
Bruno Kessler Foundation – Povo, IT
- Aurelie Clodic
LAAS – Toulouse, FR
- Kerstin I. Eder
University of Bristol, GB
- Rüdiger Ehlers
Clausthal-Zellefeld, DE
- Victor Fernandez Castro
ENS – Paris, FR
- Ivan Gavran
MPI-SWS – Kaiserslautern, DE
- Michael Gienger
Honda Research Europe – Offenbach, DE
- Marc Hanheide
University of Lincoln, GB
- Christoffer R, Heckman
University of Colorado – Boulder, US
- Guy Hoffman
Cornell University – Ithaca, US
- Shanee Honig
Ben Gurion University – Beer Sheva, IL
- Felix Ingrand
LAAS – Toulouse, FR
- Nils Jansen
Radboud University Nijmegen, NL
- Ross A. Knepper
Cornell University, US
- Hadas Kress-Gazit
Cornell University – Ithaca, US
- Jan Kretinsky
TU München, DE
- Morteza Lahijanian
University of Colorado – Boulder, US
- Ute Leonards
University of Bristol, GB
- Shelly Levy-Tzedek
Ben Gurion University – Beer Sheva, IL
- Jamy Jue Li
University of Twente, NL
- Daniele Magazzeni
King's College London, GB
- Björn Matthias
ABB AG Forschungszentrum – Ladenburg, DE
- Todd Murphey
Northwestern University – Evanston, US
- Laurel Riek
University of California – San Diego, US
- Kristin Yvonne Rozier
Iowa State University, US
- Dorsa Sadigh
Stanford University, US
- Maha Salem
Google UK – London, GB
- Satoru Satake
ATR – Kyoto, JP
- Jana Tumova
KTH Royal Institute of Technology – Stockholm, SE

Report from Dagstuhl Seminar 19082

AI for the Social Good

Edited by

Claudia Clopath[1], Ruben De Winne[2], Mohammad Emtiyaz Khan[3], and Tom Schaul[4]

1 **Imperial College London, GB,** c.clopath@imperial.ac.uk
2 **Oxfam Novib – The Hague, NL,** ruben_de_winne@hotmail.com
3 **RIKEN – Tokyo, JP,** emtiyaz.khan@riken.jp
4 **Google DeepMind – London, GB,** schaul@google.com

Abstract

Artificial intelligence (AI) and machine learning (ML) have made impressive progress in the last few years. Long-standing challenges like Go have fallen and the technology has entered daily use via the vision, speech or translation capabilities in billions of smartphones. The pace of research progress shows no signs of slowing down, and demand for talent is unprecedented. AI for Social Good in general is trying to ensure that the social good does not become an afterthought, but that society benefits as a whole. In this Dagstuhl seminar, we brought together AI and machine learning researchers with non-governmental organisations (NGOs), as they already pursue a social good goal, have rich domain knowledge, and vast networks with (non)-governmental actors in developing countries. Such collaborations benefit both sides: on the one hand, the new techniques can help with prediction, data analysis, modelling, or decision making. On the other hand, the NGOs' domains contain many non-standard conditions, like missing data, side-effects, or multiple competing objectives, all of which are fascinating research challenges in themselves. And of course, publication impact is substantially enhanced when a method has real-world impact.

In this workshop, researchers and practitioners from diverse areas of machine learning joined stakeholders from a range of NGOs to spend a week together. We first pursued an improved understanding of each side's challenges and established a common language, via presentations and discussion groups. We identified ten key challenges for AI for Social Good initiatives. To make matters concrete, we organised a hackathon around some existing technical questions within the NGOs to scope the applicability of AI methods and seed collaborations. Finally, we defined guidelines and next steps for future AI for Social Good initiatives.

Seminar February 17–22, 2019 – http://www.dagstuhl.de/19082
2012 ACM Subject Classification Computing methodologies → Machine learning
Keywords and phrases Machine Learning, Artificial Intelligence, Social Good, NGO, sustainable development goals, Non-governmental organisation
Digital Object Identifier 10.4230/DagRep.9.2.111

1 Executive Summary

Ruben De Winne (Oxfam Novib – The Hague, NL, ruben_de_winne@hotmail.com*)*

The purpose of Dagstuhl Seminar 19082: AI for the Social Good was to bring together researchers in artificial intelligence (AI) and machine learning (ML) with non-governmental organisations (NGOs) to explore if and how AI and ML could benefit the social good. Indeed, AI and ML have made impressive progress in the last few years. Long-standing challenges like Go have fallen and the technology has entered daily use via the vision, speech or translation capabilities in billions of smartphones. The pace of research progress shows no signs of slowing down, and demand for talent is unprecedented. But as part of a wider AI for Social Good trend, this seminar wanted to contribute to ensuring that the social good does not become an afterthought in the rapid AI and ML evolution, but that society benefits as a whole.

The five-day seminar brought together AI and ML researchers from various universities and industry research labs with representatives from NGOs based in Somalia, Rwanda, Uganda, Belgium, United Kingdom and The Netherlands. These NGOs all pursue various social good goals, such as increasing access to justice for vulnerable people, promoting human rights & protecting human rights defenders, and defeating poverty. On these topics, NGOs have rich domain knowledge, just like they have vast networks with (non-)governmental actors in developing countries. Mostly, NGOs have their finger on the pulse of the challenges that the world & especially its most vulnerable inhabitants are facing today, and will be facing tomorrow. The objective of the seminar was to look at these challenges through an AI and ML lens, to explore if and how these technologies could help NGOs to address these challenges. The motivation was also that collaborations between AI and ML researchers and NGOs could benefit both sides: on the one hand, the new techniques can help with prediction, data analysis, modelling, or decision making. On the other hand, the NGOs' domains contain many non-standard conditions, like missing data, side-effects, or multiple competing objectives, all of which are fascinating research challenges in themselves. And of course, publication impact is substantially enhanced when a method has real-world impact.

The seminar facilitated the exploration of possible collaborations between AI and ML researchers and NGOs through a two-pronged approach. This approach combined high-level talks & discussions on the one hand with a hands-on hackathon on the other hand. High-level talks & discussions focused first on the central concepts and theories in AI and ML and in the NGOs' development work, before diving into specific issues such as privacy & anonymity, data quality, intellectual property, accessibility and ethical issues. These talks and discussions allowed all participants – in a very short timeframe – to reach a sufficient level of understanding of each other's work. This understanding was the basis to then start investigating jointly through a hackathon how AI and ML could help addressing the real-world challenges presented by the NGOs. At the start of the hackathon, an open marketplace-like setting allowed AI and ML researchers and NGOs to find the best match between technological supply and demand. When teams of researchers and NGOs were established, their initial objective was not to start coding, but to define objectives, assess scope and feasibility. Throughout the hackathon, group membership was fluid, as some projects finished early, were deemed out of scope, or needed to wait for data. Some groups managed to build a viable initial prototype, others established the seeds for future collaborations, and a few were proposed as full summer projects within the "Data Science of Social Good summer school". The projects' aims were diverse. They included better seeds for farmers, modelling cognitive age and decline, scalable legal assistance and scalable citizen feedback. As a result of the hackathon, all NGOs could take concrete results home – some to build on further, some as mature solutions.

Finally, a result of the seminar that is relevant for the entire AI for Social Good community are the ten key challenges for AI for Social Good initiatives that participants identified:

1. the importance of deep, long-term partnerships,
2. clear and well-defined goals and use cases,

3. bias towards simpler solutions,
4. data readiness,
5. setting expectations with regards to both impact and the pace at which technology can be applied,
6. ensuring privacy and security of data,
7. inclusivity and ethics of the applications,
8. factoring in the limitations of both communities,
9. challenges in overcoming the barriers to NGOs utilising the potential of AI/ML, and
10. the relative cost of AI/ML for social good.

2 Table of Contents

3 Overview of Talks

3.1 Introduction to the workshop

Claudia Clopath (Imperial College London, GB)

Claudia Clopath introduced the notion of "AI for Social Good" and the motivation for the workshop. In particular, she listed the following goals: 1) Bring NGO participants and AI researchers together, 2) Create awareness, 3) Make sure incentives are aligned, 4) New innovations coming out of the newly formed collaborations.

The potential outcomes of the workshops were: 1) Understanding the link between the two worlds: Machine Learning and NGOs, 2) Understanding the ways to make AI for Social Good initiatives work, 3) Build Prototypes during the hackathon time, 4) Build collaborations, create a network, make friends, 5) and prepare a set of follow-up plans. Finally, she defined a set of rules for the workshop: Connect, Respect, Bottom-Up, Brain-storm, Fun.

3.2 Successful examples of AI for Social Good

Julien Cornebise (Element AI- London, GB)

The movement of "AI for Good" is undergoing a great gain of interest, with the United Nations AI for Good Summit being in its third iteration and several "AI for Good" workshops proposed for NeurIPS 2018. Yet it is still pretty recent, and as with any young domain, this might cause some teething problems. The current hype around Artificial Intelligence and Machine Learning can also induce some measure of "tech saviour syndrome". In the field of "for good" however, resources are often scarce and lives can be at stake, bringing a very high cost to any such early mistakes.

In this talk, we draw parallels with three decades of ICT4D (Information and Communications Technology for Development), to learn some of key lessons and avoid repeating mistakes from the past. We start by suggesting a way to navigate the obvious question of "What is good?" by partnering with actual domain experts. We show some of the key factors of success and explain how to navigate some of the pitfalls that can rise in such projects. We illustrate the point with two concrete successful projects over the last three years, which brought together domain experts from Amnesty International, and machine learners. We finally open the conversation with a wider reflection on how the current specialists in machine learning, mainly employed in the private sector, may be channelled to help with much larger social problems that we are facing as a society and as a species.

3.3 Using Machine Learning to Address Global Challenges

Shakir Mohamed (Google DeepMind – London, GB)

The sustainable development goals (SDG) give a broad framework for addressing global challenges that have been agreed to by the governments of the world, and we now have the opportunity to think of the ways in which machine learning can contribute to addressing this global agenda. To achieve this, we will discuss approaches for doing this using the integrated transformations given by the SDGs, the need for high-frequency and diverse data sources, and the central role of multi-disciplinary teams. We'll use the example of satellite imagery and look at how they provide a data source to look at global challenges in poverty prediction, deforestation, global fishing monitoring, and the deployment of domestic solar panels. We will also explore other data sources and areas in weather and early warning, energy, healthcare and conservation.

3.4 Introduction to Machine Learning I

Mohammad Emtiyaz Khan (RIKEN – Tokyo, JP)

In this talk, I will present an initial introduction to Machine Learning (ML). I will start with the historical context of the relationship of ML with AI, and formulate a general definition to differentiate it from other fields such as statistics, data mining etc. I will summarise some of the recent successes, but also talk about some documented failures. I will conclude by mentioning subfields within machine learning and some of the open problems.

3.5 Introduction to Machine learning II

Yee Whye Teh (University of Oxford, GB)

Yee Whye Teh introduced the difference between unsupervised, supervised and reinforcement learning. He explained the different levels of difficulty and what is required to make them work. He explained–in simple language for the NGO participants–the advantages and disadvantages of the three different methods and illustrated these points with examples.

3.6 Introduction to Reinforcement Learning – chalk talk

Tom Schaul (Google DeepMind – London, GB)

Reinforcement learning (RL) is both one of the most general types of AI and one of the more difficult ones to apply. It permits the automation of complex sequential decision making processes, maximising any well-defined objective. It learns by interacting, trying things out and learning from its successes and mistakes, and it thus needs only minimal knowledge about the task. To work well, however, it requires a lot of direct interaction, thus making many potentially costly/harmful mistakes while learning is still going on. The second drawback for practical applications is the fact that defining the reward/success can be complicated or ambiguous–yet, clarity of objective is a desirable thing in itself, so framing a problem as an RL problem often leads to more clarity.

4 Working groups

Cracking each other's code

We broke up into small sessions and discuss what each participant is going on an everyday basis at work and what are the terms we commonly use. This was a way to get to know each other, understand each other's work and challenges and understand each other's jargon. One important point was to outline what defines success in their work.

Topics for discussion

We dedicated one session to eliciting a broad range of topics of discussion, surrounding AI for Social Good, from our highly diverse and interdisciplinary participants. On the basis of this list (see below) we grouped and prioritised, and those resulted in the break-out groups for the following days–but of course, not all of them could be discussed within the limited time.

- Trust and transparency
- Privacy and confidentiality
- Equality
- Ethics and principles
- Affordability, funding
- Talent, capacity, matchmaking
- Intellectual property
- Access, usability, bandwidth, illiteracy
- Who is the user?
- SDG alignment
- Urgency, relevance
- Maintenance, technology transfer, integration, sustainability
- Robustness, verifiability, open data
- Fake claims, adversarial actors, exploiters, aid shopping
- Text (Docs + informal), NLP (natural language processing), dialogue
- Voice

- Privacy, anonymisation, safety
- Data quality (scarcity, missing data, sample size, confidence in the data)
- Data safety
- Sensors
- Visualisation
- Predictive modeling
- Causality, outliers, classification similarities
- Performance evaluation
- Auto ML
- Transfer + generalisation
- Bias, interpretability
- Continual learning, adaptation
- Measurement models
- Feature design
- On-line versus off-line

Theory of Change

Theories of change are an important technical term in civil society and the NGO world. They capture a problem specific understanding of causal structure. Most government, institutions, and grant making institutions use theories of change to structure interventions. Causality, as studied by machine learners, can be a complementary tool to theories of change.

Urgency, Relevance and Goals

NGOs have a tension between quality and quantity of impact. For example, donors would like to see big numbers of beneficiaries, but deep individual stories are important for private supporters. This leads to a risk of cherry-picking if the strategy is only going for quality, which in turn can be perceived as unfair in the community of the beneficiaries. NGOs need to find an empty niche where they can avoid competing with other NGOs. But importantly, the relevance of the projects needs to be aligned with their mission, needs to reflect the donor's priorities and needs to be centred on the beneficiaries. NGOs have tend to not plan long-term, as funding cycles are short, and many funders avoid taking risks. However, the reality is that long-term investments tend to help in many ways, including the urgent issues (e.g., education is useful for addressing climate change or epidemics). We discussed ways to go around that, namely 1) using for-profit companies, 2) by investing in long-term monitoring like sensors for example, 3) exerting political pressure to change the culture and constraints. Finally, we discussed the problem of "free riders" who don't contribute to solving the problem but take advantage of the solution.

Privacy & anonymity

We discuss the ways to encrypt data both for storing and sharing. We discuss the tools to anonymise data, e.g., removing fields with personally identifiable information (PII) or using tools to obtain k-anonymity. Multi-party computation permits collaborating without sharing data. If we do linear data analysis, adding noise makes sure individuals cannot be identified. Finally, we talked about how to find and remove biases in data and the risk of fake data.

Use of Auto-ML

Auto-ML is a tool to democratise ML so that non-experts can use ML technology. NGOs could start using it, but they at least need a software engineer to begin with. We discussed the fact the the model is only as strong as the data, so if the data is not good, Auto-ML cannot do miracles. We talked about the issue of interpretability: we can actually understand what the machine is doing. This is particularly important if one needs to be accountable for the conclusions the machine is making.

Data quality

We discuss of concept of "Garbage in, garbage out": if the data is not good, the ML won't be good either. We discussed ways to clean the data both on the NGOs side and on the ML side, and acknowledged the fact that data cleaning is typically a large part of the whole project. We discussed the different possible data formats: tabular, speech, text. Finally, we talked about predictive models and that fact that each project separately needs to define what is the minimum viable solution, and what are requirements in terms of fairness and beyond raw performance. Finally, we presented the "Data Science Hierarchy of Needs", where there are 6 levels, from the bottom to the top:

1. Collect data: instrumentation, logging, sensors, external data, user generated content.
2. Move and Store: reliable data flow, infrastructure, pipelines, data storage.
3. Explore and transform: cleaning the data, detect anomalies, etc.
4. Aggregate and label: analytics, metrics, segments, aggregates, features, training data.
5. Learning/optimise: A/B testing, experimentation, simple ML algorithms.
6. AI and Deep learning.

Only when the 5 first levels are established, does it make sense to move on to the 6th level and invest in state-of-the-art AI methods.

Intellectual property

We first discussed the how important it is for NGOs to know their level of maturity in terms of data availability, cleanliness and usability. It is important that intellectual property (IP) is taken into account throughout the project. In particular, NGOs need to understand the value of their data and also use that to pitch it better to their donor: if the data Is better/richer/more abundant, the results will be more interesting.

Accessibility

We first discussed the accessibility to the most vulnerable, namely in terms of language barriers, disabilities, minority groups, access to devices, internet connectivity and infrastructure. We talked about the fact that technology should be "accessible by design". We then talked about usability both from the NGOs side and from the beneficiaries side. These principles should be aligned with the SDGs to ensure inclusiveness. In order to achieve that, we should look at the feedback from users and how we can correct the system to ensure accessibility. We should think of the eligibility criteria for the users and staff, and about training them.

Ethical issues

As this topic is so complex and multi-faceted, we had multiple separate break-out groups on the topic of ethics. Among the issues that were raised were the following:

- Involving the voice of affected people versus not endangering them
- Transparency versus privacy
- Transparency versus solving the problem
- Meaningful informed consent, building trust, buy-in from all stakeholders
- Gradual/iterative deployment versus quick impact with risks
- Job loss risks versus efficiency gains
- Safe exploration of private data
- Framing of risks, perception, awareness and public relations
- Responsibility to identify the most vulnerable
- Who is responsible for the decisions/mistakes the ML system makes?
- Mitigating risks
- Delegating to the machine versus helping humans to make better decisions

On the solution side, we discussed the Toronto framework on responsibly dealing with data, and the Data4Development criteria.

5 Hackathon

Two days of the workshop were dedicated to a hackathon. After project presentations by the NGO participants and tool presentations by the ML participants, we had a couple of match-making rounds to form initial mixed groups. The initial objective for each group was not to start coding, but to define objectives, assess scope and feasibility. Throughout the hackathon, group membership was fluid, as some projects finished early, were deemed out of scope, or needed to wait for data. Some groups managed to build a viable initial prototype, others established the seeds for future collaborations, and a few were proposed as full summer projects within the "Data Science of Social Good summer school". Among the projects were the following:

Better seeds for farmers. We built a clustering algorithm that matches the similarity in score for (standardised) crop varieties and traits, so that users of the application can view which varieties of a specific crop display similar traits based on their own scoring of a certain variety.

Modelling cognitive age and decline. We built a prototype to distinguish between groups of people with different levels of cognitive decline. We used publicly available data.

Scalable legal assistance. Legal advice is typically accessible to only a certain subset of people. Here we tried to improve an existing process of providing legal recommendations to the least privileged beneficiaries. In particular, a recommendation system prototype (based on past responses) was designed to making some of that process less labour-intensive, so that the same number of lawyers can serve more requests.

Competent judges. Another way of improving outcomes in the justice system is to improve the quality and consistency of judgements, by training judges and automatic monitoring. In the workshop, we defined steps towards enabling a such a project and how to gather the necessary data.

Scalable citizen feedback. Two projects were centred around how to effectively handle large amounts of citizen feedback, but they differed in the data modality (voice messages versus tabular). The voice message feedback service service looked promising at first, but we realised that there was very little clean and annotated data, and there do not currently exist out-of-the-box tools for voice recognition in languages like Somali.

Spaces for trusted discussions. We assessed the feasibility to study the effect of moderation on social media data.

6 Guidelines / Takeaways

The seminar highlighted ten key insights amongst domain experts from NGOs and technical specialists from machine learning, in relation to:

1. the importance of deep, long-term partnerships,
2. clear and well-defined goals and use cases,
3. bias towards simpler solutions,
4. data readiness,
5. setting expectations with regards to both impact and the pace at which technology can be applied,
6. ensuring privacy and security of data,
7. inclusivity and ethics of the applications,
8. factoring in the limitations of both communities,
9. challenges in overcoming the barriers to NGOs utilising the potential of AI/ML, and
10. the relative cost of AI/ML for social good.

7 Follow-up actions

In the last part of the seminar, we discussed ways forward for AI for Social Good initiatives in general, and with the collaborations we started during the week. We discussed the following action plan:

- We plan to write a manuscript with the take-home messages that we have learned during the workshop. The audience targeted are people who are running or want to run AI for Social Good initiatives.
- We plan to write a Global Challenges manuscript that relates examples of AI for Social Good projects to the Sustainable Developmental Goals.
- We plan to write a third document with the key principles of AI for Social Good projects, to serve as a call to action.
- We plan to identify, organise and share the different sources of funding that could be used for AI for Social Good projects.
- We listed possible future meetings and workshops.
- We finally made sure that the collaborations we started at the seminar will continue. To that end, we defined multiple milestones and commitments, we built working groups, etc.

Participants

- Gerald Abila
BarefootLaw – Kampala, UG
- Hisham Almiraat
Justice and Peace Netherlands – The Hague, NL
- Hiromi Arai
RIKEN – Tokyo, JP
- Danielle C. M. Belgrave
Imperial College London, GB
- Fanny Cachat
ICJ – Brussels, BE
- Claudia Clopath
Imperial College London, GB
- Bec Connelly
RNW Media – Hilversum, NL
- Julien Cornebise
Element AI- London, GB
- Wilfried de Wever
SEMA – Kampala, UG
- Ruben De Winne
Oxfam Novib – The Hague, NL
- Daphne Ezer
University of Warwick, GB
- Tobias Glasmachers
Ruhr-Universität Bochum, DE
- Frank Hutter
Universität Freiburg, DE
- Mohammad Emtiyaz Khan
RIKEN – Tokyo, JP
- Shakir Mohamed
DeepMind, GB
- Frank Mugisha
Chemonics International Inc. – Kigali, RW
- Mihoko Otake
RIKEN – Tokyo, JP
- Mustafa Othman
Shaqodoon Organization – Hargeisa, SO
- Angela Picciariello
Oxfam – Oxford, GB
- Julia Proskurnia
Google Switzerland – Zürich, CH
- Tom Schaul
Google DeepMind – London, GB
- Kyle Snyder
RNW Media – Hilversum, NL
- Yee Whye Teh
University of Oxford, GB
- Nenad Tomasev
Google DeepMind – London, GB

Report from Dagstuhl Seminar 19092

Beyond-Planar Graphs: Combinatorics, Models and Algorithms

Edited by

Seok-Hee Hong[1], Michael Kaufmann[2], János Pach[3], and Csaba D. Tóth[4]

1 **The University of Sydney, AU,** seokhee.hong@sydney.edu.au
2 **Universität Tübingen, DE,** mk@informatik.uni-tuebingen.de
3 **EPFL – Lausanne, CH,** pachjanos@gmail.com
4 **California State University – Northridge, US,** csaba.toth@csun.edu

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19092 "Beyond-Planar Graphs: Combinatorics, Models and Algorithms" which brought together 36 researchers in the areas of graph theory, combinatorics, computational geometry, and graph drawing. This seminar continued the work initiated in Dagstuhl Seminar 16452 "Beyond-Planar Graphs: Algorithmics and Combinatorics" and focused on the exploration of structural properties and the development of algorithms for so-called beyond-planar graphs, i.e., non-planar graphs that admit a drawing with topological constraints such as specific types of crossings, or with some forbidden crossing patterns. The seminar began with four talks about the results of scientific collaborations originating from the previous Dagstuhl seminar. Next we discussed open research problems about beyond planar graphs, such as their combinatorial structures (e.g., book thickness, queue number), their topology (e.g., simultaneous embeddability, gap planarity, quasi-quasiplanarity), their geometric representations (e.g., representations on few segments or arcs), and applications (e.g., manipulation of graph drawings by untangling operations). Six working groups were formed that investigated several of the open research questions. In addition, talks on related subjects and recent conference contributions were presented in the morning opening sessions. Abstracts of all talks and a report from each working group are included in this report.

Seminar February 24 – March 1, 2019 – http://www.dagstuhl.de/19092

2012 ACM Subject Classification Computing methodologies → Combinatorial algorithms, Mathematics of computing → Graph algorithms, Theory of computation → Computational geometry, Human-centered computing → Graph drawings

Keywords and phrases combinatorial geometry, geometric algorithms, graph algorithms, graph drawing, graph theory, network visualization

Digital Object Identifier 10.4230/DagRep.9.2.123

Edited in cooperation with Henry Förster

1 Executive Summary

Seok-Hee Hong (The University of Sydney, AU)
Michael Kaufmann (Universität Tübingen, DE)
János Pach (EPFL – Lausanne, CH)
Csaba D. Tóth (California State University – Northridge, US)

Most big data sets are relational, containing a set of objects and relations between the objects. This is commonly modeled by graphs, with the objects as the vertices and the relations as the edges. A great deal is known about the structure and properties of special types of graphs, in particular *planar graphs* which are fundamental for both Graph Theory, Graph Algorithms and Automatic Layout. Structural properties of planar graphs can often be expressed, for example, in terms of excluded minors, low density, and small separators. These properties lead to efficient algorithms; consequently a number of fundamental algorithms for planar graphs have been discovered. As many of the characteristic properties of planar graphs have been generalized (e.g., graph minor theory, topological obstructions, χ-boundedness), these algorithms also extend in various directions to broad families of graphs.

Typical real world graphs, such as social networks and biological networks, are *nonplanar*. In particular, the class of scale-free networks, which can be used to model web-graphs, social networks and many kinds of biological networks, are sparse nonplanar graphs, with globally sparse and locally dense structure. To analyze and visualize such real world networks, we need to formulate and solve fundamental mathematical and algorithmic research questions on *sparse nonplanar* graphs. Sparsity, in most cases, is explained by properties that generalize those of planar graphs: in terms of topological obstructions or forbidden intersection patterns among the edges. These are called *beyond-planar graphs*. Important beyond-planar graph classes include the following:

- *k-planar graphs*: graphs that can be drawn with at most k crossings per edge;
- *k-quasi-planar graphs*: graphs which can be drawn without k mutually crossing edges;
- *k-gap-planar graphs*: graphs that admit a drawing in which each crossing is assigned to one of the two involved edges and each edge is assigned at most k of its crossings;
- *RAC (Right Angle Crossing) graphs*: graphs that have straight-line drawings in which any two crossing edges meet in a right angle;
- *bar k-visibility graphs*: graphs whose vertices are represented as horizontal segments (bars) and edges are represented as vertical lines connecting bars, intersecting at most k bars;
- *fan-crossing-free graphs*: graphs which can be drawn without fan-crossings; and
- *fan-planar graphs*: graphs which can be drawn such that every edge is crossed only by pairwise adjacent edges (fans).

Compared to the first edition of the seminar, we planned to focus more on aspects of computational geometry. Therefore, we included one new organizer as well as some more participants from this field.

Thirty-six participants met on Sunday afternoon for a first informal get-together and reunion since the last workshop which took place more than two years ago. From that event, the four working groups nearly all have completed and published subsequent work. We decided to build on the achievements of the previous meeting and scheduled short talks recalling the previous seminar's results. On Monday afternoon, we held an engaging open problems session and formed new working groups. Notably, this time, more problems related to computational geometry as well as questions from combinatorics have been proposed. Open problems included questions about the combinatorial structures (e.g, book thickness, queue number), the topology (e.g., simultaneous embeddability, gap planarity, quasi-quasiplanarity), the geometric representations (e.g., representations on few segments or arcs), and applications (e.g., manipulation of graph drawings by untangling operations) of beyond-planar graphs.

In the opening session of every morning, we have drawn inspiration from additional talks, fresh conference contributions on related topics (see abstracts). An impressive session on the last day was devoted to progress reports that included plans for publications and follow-up projects among researchers that would have been highly unlikely without this seminar. From our personal impression and the feedback of the participants, the seminar has initiated collaboration and lead to new ideas and directions.

We thank all the people from Schloss Dagstuhl for providing a positive environment and hope to repeat this seminar, possibly with some new focus, for a third time.

2 Table of Contents

3 Overview of Talks

3.1 On the relationship between k-planar and k-quasiplanar graphs

Patrizio Angelini (Universität Tübingen, DE)

Joint work of Patrizio Angelini, Michael A. Bekos, Franz J. Brandenburg, Giordano Da Lozzo, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, Ignaz Rutter, Michael Hoffmann, Csaba Tóth
Main reference Patrizio Angelini, Michael A. Bekos, Franz J. Brandenburg, Giordano Da Lozzo, Giuseppe Di Battista, Walter Didimo, Giuseppe Liotta, Fabrizio Montecchiani, Ignaz Rutter: "On the Relationship Between k-Planar and k-Quasi-Planar Graphs", in Proc. of the Graph-Theoretic Concepts in Computer Science – 43rd International Workshop, WG 2017, Eindhoven, The Netherlands, June 21-23, 2017, Revised Selected Papers, Lecture Notes in Computer Science, Vol. 10520, pp. 59–74, Springer, 2017.
URL https://doi.org/10.1007/978-3-319-68705-6_5
Main reference Michael Hoffmann, Csaba D. Tóth: "Two-Planar Graphs Are Quasiplanar", in Proc. of the 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 – Aalborg, Denmark, LIPIcs, Vol. 83, pp. 47:1–47:14, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2017.
URL http://dx.doi.org/10.4230/LIPIcs.MFCS.2017.47

In the area of beyond planarity, the two most studied families of graph classes are those of k-planar and k-quasiplanar graphs. A graph is k-planar if it admits a drawing in the plane so that no edge is crossed by more than k edges, while it is k-quasiplanar if it admits a drawing that contains no set of pairwise crossing edges.

We are interested in inclusion relationships between the classes belonging to these two families. Clearly, every k-planar graph is $(k + 1)$-planar, and every k-quasiplanar graph is $(k + 1)$-quasiplanar, and hence the two families define proper hierarchies. On the other hand, the relationship between these two hierarchies is not well established yet. The only result, which follows from the definitions, is that every k-planar graph is $(k + 2)$-quasiplanar.

In this work we prove that every k-planar graph is also $(k + 1)$-quasiplanar. This result is obtained by a rerouting technique that solves all sets of $k + 1$ pairwise crossing edges without introducing new ones. The question whether every k-planar graph is also k-quasiplanar, for $k > 2$, remains open.

3.2 Z_2-genus of graphs and minimum rank of partial symmetric matrices

Radoslav Fulek (IST Austria – Klosterneuburg, AT)

Joint work of Radoslav Fulek, Jan Kynčl
Main reference Radoslav Fulek, Jan Kynčl: "Z_2-Genus of Graphs and Minimum Rank of Partial Symmetric Matrices", in Proc. of the 35th International Symposium on Computational Geometry, SoCG 2019, LIPIcs, Vol. 129, pp. 39:1–39:16, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.
URL http://dx.doi.org/10.4230/LIPIcs.SoCG.2019.39

The *genus* $\mathrm{g}(G)$ of a graph G is the minimum g such that G has an embedding on the orientable surface M_g of genus g. A drawing of a graph on a surface is *independently even* if every pair of nonadjacent edges in the drawing crosses an even number of times. The $\mathbb{Z}_2$-*genus* of a graph G, denoted by $\mathrm{g}_0(G)$, is the minimum g such that G has an independently even drawing on M_g. By a result of Battle, Harary, Kodama and Youngs from 1962, the graph genus is additive over 2-connected blocks. In 2013, Schaefer and Štefankovič proved that the

$\mathbb{Z}_2$-genus of a graph is additive over 2-connected blocks as well, and asked whether this result can be extended to so-called 2-amalgamations, as an analogue of results by Decker, Glover, Huneke, and Stahl for the genus. We give the following partial answer. If $G = G_1 \cup G_2$, G_1 and G_2 intersect in two vertices u and v, and $G - u - v$ has k connected components (among which we count the edge uv if present), then $|\mathrm{g}_0(G) - (\mathrm{g}_0(G_1) + \mathrm{g}_0(G_2))| \leq k + 1$. For complete bipartite graphs $K_{m,n}$, with $n \geq m \geq 3$, we prove that $\frac{\mathrm{g}_0(K_{m,n})}{\mathrm{g}(K_{m,n})} = 1 - O(\frac{1}{n})$. Similar results are proved also for the Euler $\mathbb{Z}_2$-genus. We express the $\mathbb{Z}_2$-genus of a graph using the minimum rank of partial symmetric matrices over $\mathbb{Z}_2$; a problem that might be of independent interest.

3.3 Planar Graphs of Bounded Degree have Bounded Queue Number

Henry Förster (Universität Tübingen, DE), Michael Bekos (Universität Tübingen, DE), Martin Gronemann (Universität Köln, DE), Tamara Mchedlidze (KIT – Karlsruher Institut für Technologie, DE), Fabrizio Montecchiani (University of Perugia, IT), Chrysanthi Raftopoulou (National Technical University of Athens, GR), and Torsten Ueckerdt (KIT – Karlsruher Institut für Technologie, DE)

Main reference Michael Bekos, Henry Förster, Martin Gronemann, Tamara Mchedlidze, Fabrizio Montecchiani, Chrysanthi Raftopoulou, and Torsten Ueckerdt: "Planar Graphs of Bounded Degree Have Bounded Queue Number", in Proc. of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC'19), June 23–26, 2019, Phoenix, AZ, USA. ACM, New York, NY, USA, 9 pages.
URL https://doi.org/10.1145/3313276.3316324

A *queue layout* of a graph consists of a *linear order* of its vertices and a partition of its edges into *queues*, so that no two independent edges of the same queue are nested. The *queue number* of a graph is the minimum number of queues required by any of its queue layouts.

A long-standing conjecture by Heath, Leighton and Rosenberg states that the queue number of planar graphs is bounded. This conjecture has been partially settled in the positive for several subfamilies of planar graphs (most of which have bounded treewidth). In this talk, we present a new important step towards settling this conjecture. We prove that planar graphs of bounded degree (which may have unbounded treewidth) have bounded queue number.

A notable implication of this result is that every planar graph of bounded degree admits a three-dimensional straight-line grid drawing in linear volume. Further implications are that every planar graph of bounded degree has bounded track number, and that every k-planar graph (i.e., every graph that can be drawn in the plane with at most k crossings per edge) of bounded degree has bounded queue number.

3.4 Orthogonal and Smooth Orthogonal Layouts of 1-Planar Graphs with Low Edge Complexity

Chrysanthi Raftopoulou (National Technical University of Athens, GR)

Joint work of Chrysanthi Raftopoulou, Evmorfia Argyriou, Sabine Cornelsen, Henry Förster, Michael Kaufmann, Martin Nöllenburg, Yoshio Okamoto, Alexander Wolff
Main reference Evmorfia N. Argyriou, Sabine Cornelsen, Henry Förster, Michael Kaufmann, Martin Nöllenburg, Yoshio Okamoto, Chrysanthi N. Raftopoulou, Alexander Wolff: "Orthogonal and Smooth Orthogonal Layouts of 1-Planar Graphs with Low Edge Complexity", 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. 509–523, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-04414-5_36

While orthogonal drawings have a long history, smooth orthogonal drawings have been introduced only recently. So far, only planar drawings or drawings with an arbitrary number of crossings per edge have been studied. Recently, a lot of research effort in graph drawing has been directed towards the study of beyond-planar graphs such as 1-planar graphs, which admit a drawing where each edge is crossed at most once. In this talk, we consider graphs with a fixed embedding. For 1-planar graphs, we present algorithms that yield orthogonal drawings with optimal edge complexity and smooth orthogonal drawings with small edge complexity. For the subclass of outer-1-planar graphs, which can be drawn such that all vertices lie on the outer face, we achieve optimal edge complexity for both, orthogonal and smooth orthogonal drawings.

3.5 Inserting an Edge into a Geometric Embedding

Ignaz Rutter (Universität Passau, DE)

Joint work of Marcel Radermacher, Ignaz Rutter
Main reference Marcel Radermacher, Ignaz Rutter: "Inserting an Edge into a Geometric Embedding", 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. 402–415, Springer, 2018.
URL https://doi.org/10.1007/978-3-030-04414-5_29

The algorithm to insert an edge e in linear time into a planar graph G with a minimal number of crossings on e [1], is a helpful tool for designing heuristics that minimize edge crossings in drawings of general graphs. Unfortunately, some graphs do not have a geometric embedding Γ such that $\Gamma + e$ has the same number of crossings as the embedding $G + e$. This motivates the study of the computational complexity of the following problem: Given a combinatorially embedded graph G, compute a geometric embedding Γ that has the same combinatorial embedding as G and that minimizes the crossings of $\Gamma + e$. We give polynomial-time algorithms for special cases and prove that the general problem is fixed-parameter tractable in the number of crossings. Moreover, we show how to approximate the number of crossings by a factor $(\Delta - 2)$, where Δ is the maximum vertex degree of G.

References

1 Gutwenger, C., Mutzel, P., Weiskircher, R.: Inserting an Edge into a Planar Graph. Algorithmica **41**(4), 289–308 (2005). 10.1007/s00453-004-1128-8

3.6 A crossing lemma for multigraphs

Géza Tóth (Alfréd Rényi Institute of Mathematics – Budapest, HU) and János Pach (EPFL – Lausanne, CH)

Joint work of Géza Tóth, János Pach
Main reference János Pach, Géza Tóth: "A Crossing Lemma for Multigraphs", in Proc. of the 34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary, LIPIcs, Vol. 99, pp. 65:1–65:13, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2018.
URL http://dx.doi.org/10.4230/LIPIcs.SoCG.2018.65

Let G be a drawing of a graph with n vertices and $e > 4n$ edges, in which no two adjacent edges cross and any pair of independent edges cross at most once. According to the celebrated Crossing Lemma of Ajtai, Chvátal, Newborn, Szemerédi and Leighton, the number of crossings in G is at least $c\frac{e^3}{n^2}$, for a suitable constant $c > 0$. In a seminal paper, Székely generalized this result to multigraphs, establishing the lower bound $c\frac{e^3}{mn^2}$, where m denotes the maximum multiplicity of an edge in G. We get rid of the dependence on m by showing that, as in the original Crossing Lemma, the number of crossings is at least $c'\frac{e^3}{n^2}$ for some $c' > 0$, provided that the "lens" enclosed by every pair of parallel edges in G contains at least one vertex. This settles a conjecture of Bekos, Kaufmann, and Raftopoulou.

This work started at the Dagstuhl Seminar "Beyond-Planar Graphs: Algorithmics and Combinatorics", November 6-11, 2016, in a working group, together with Stefan Felsner, Michael Kaufmann, Vincenzo Roselli, Torsten Ueckerdt, and Pavel Valtr. We are very grateful to them for their valuable comments, suggestions, and for many interesting discussions.

3.7 The Number of Crossings in Multigraphs with No Empty Lens

Torsten Ueckerdt (KIT – Karlsruher Institut für Technologie, DE)

Joint work of Michael Kaufmann, János Pach, Géza Tóth, Torsten Ueckerdt
Main reference Michael Kaufmann, János Pach, Géza Tóth, Torsten Ueckerdt: "The Number of Crossings in Multigraphs with No Empty Lens", 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. 242–254, Springer, 2018.
URL http://dx.doi.org/10.1007/978-3-030-04414-5_17

Let G be a multigraph with n vertices and $e > 4n$ edges, drawn in the plane such that any two parallel edges form a simple closed curve with at least one vertex in its interior and at least one vertex in its exterior. Pach and Tóth [1] extended the Crossing Lemma of Ajtai *et al.* [2] and Leighton [3] by showing that if no two adjacent edges cross and every pair of nonadjacent edges cross at most once, then the number of edge crossings in G is at least $\alpha e^3/n^2$, for a suitable constant $\alpha > 0$. The situation turns out to be quite different if nonparallel edges are allowed to cross any number of times. It is proved that in this case the number of crossings in G is at least $\alpha e^{2.5}/n^{1.5}$. The order of magnitude of this bound cannot be improved.

This project initiated at the Dagstuhl seminar 16452 "Beyond-Planar Graphs: Algorithmics and Combinatorics," November 2016. We would like to thank all participants, especially Stefan Felsner, Vincenzo Roselli, and Pavel Valtr, for fruitful discussions.

References

1 J. Pach and G. Tóth. *A Crossing Lemma for Multigraphs.* In B. Speckmann and C. D. Tóth, editors, *34th International Symposium on Computational Geometry (SoCG 2018)*, volume 99 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 65:1–65:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

2 M. Ajtai, V. Chvátal, M. M. Newborn, and E. Szemerédi. *Crossing-free subgraphs. North-Holland Mathematics Studies*, 60(C):9–12, 1982.

3 T. Leighton. *Complexity issues in VLSI.* Foundations of computing series, 1983.

3.8 Every collinear set in a planar graph is free

Vida Dujmović (University of Ottawa, CA), Fabrizio Frati (Roma Tre University, IT), Günter Rote (FU Berlin, DE)

Joint work of Vida Dujmović, Fabrizio Frati, Daniel Gonçalves, Pat Morin, Günter Rote
Main reference Vida Dujmović, Fabrizio Frati, Daniel Gonçalves, Pat Morin, Günter Rote: "Every Collinear Set in a Planar Graph Is Free", in Proc. of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019, pp. 1521–1538, SIAM, 2019.
URL http://dx.doi.org/10.1137/1.9781611975482.92

We show that if a planar graph G has a plane straight-line drawing in which a subset S of its vertices are collinear, then for any set of points, X, in the plane with $|X| = |S|$, there is a plane straight-line drawing of G in which the vertices in S are mapped to the points in X. This solves an open problem posed by Ravsky and Verbitsky in 2008. In their terminology, we show that every collinear set is free.

This result has applications in graph drawing, including untangling, column planarity, universal point subsets, and partial simultaneous drawings.

Preprint of the full paper: http://arxiv.org/abs/1811.03432

4 Working groups

4.1 Traversing Edges

Eyal Ackerman (University of Haifa, IL), Stefan Felsner (TU Berlin, DE), Radoslav Fulek (IST Austria – Klosterneuburg, AT), Balázs Keszegh (Alfréd Rényi Institute of Mathematics – Budapest, HU), János Pach (EPFL – Lausanne, CH), Günter Rote (FU Berlin, DE), Csaba D. Tóth (California State University – Northridge, US), Géza Tóth (Alfréd Rényi Institute of Mathematics – Budapest, HU), and Torsten Ueckerdt (KIT – Karlsruher Institut für Technologie, DE)

A *geometric graph* is a graph drawn in the plane such that its vertices are distinct points in a general position (no three on a line) and its edges are straight-line segments. Two edges in a geometric graph are either adjacent, crossing or disjoint. Disjoint edges may be further classified as *avoiding* (or *parallel*) and nonavoiding, where two disjoint edges are called avoiding if their endpoints are in convex position. Define two edges to be *traversing* if they are crossing or they are disjoint and nonavoiding. In other words, two edges are traversing if at least one of them contains in its interior the intersection point of the two lines that contain the two edges.

It is a natural question to ask for the density of a geometric graph with no k pairwise conflicting edges, where ‘conflicting’ refers to one of the above-mentioned relations between two edges.[1] The case of no k pairwise adjacent edges is not interesting as it implies that the maximum degree is $k-1$. Considering geometric graphs with no k pairwise disjoint edges, it was first proved by Pach and Törőcsik [11] that they have linearly many edges. The best bound is due to G. Tóth [13]:

▶ **Theorem 1** ([13]). *An n-vertex geometric graph with no k pairwise disjoint edges has $O(k^2 n)$ edges.*

Valtr [15] proved a linear bound considering pairwise avoiding edges.

▶ **Theorem 2** ([15]). *An n-vertex geometric graph with no k pairwise avoiding edges has $O_k(n)$ edges.*

The case of pairwise crossing edges is a special case of a famous and rather old conjecture [6, 8] concerning the density of *k-quasi-planar* graphs.[2]

▶ **Conjecture 3.** *An n-vertex k-quasi-planar graph has $O_k(n)$ edges.*

This conjecture is known to hold for $k \leq 4$ [1, 4, 5] but for $k > 4$ it is open even for geometric graphs. The best bound is due to Valtr:

▶ **Theorem 4** ([14]). *An n-vertex geometric graph with no k pairwise crossing edges has $O_k(n \log n)$ edges.*

The main goal of our workgroup was to prove the following relaxed variant of Conjecture 3:

▶ **Conjecture 5** ([3]). *An n-vertex geometric graph with no k pairwise traversing edges has $O_k(n)$ edges.*

An n-vertex geometric graph with no pair of traversing edges is outerplanar and therefore has at most $2n-3$ edges (for $n > 1$). For $k \leq 4$ Conjecture 5 holds since Conjecture 3 holds. For $k > 4$ a possible approach to prove Conjecture 5 would have been to provide a positive answer to the following question.

▶ **Problem 6.** *Is it true that every set of m segments in the plane without k pairwise traversing segments contains a subset of $\Omega_k(m)$ segments no two of which are traversing?*

Indeed, if this question had an affirmative answer, then it would imply Conjecture 5 as follows. Given an n-vertex geometric graph with m edges, no k of which are pairwise traversing, one can slightly shorten each edge and obtain a set of m segments, no k of which are pairwise traversing. Suppose that this set contains $c_k m$ segments such that no two of them are traversing, for some $c_k > 0$. Then the corresponding edges of the graph are pairwise nontraversing and hence $c_k m \leq 2n-3$ and Conjecture 5 follows. Unfortunately, by modifying a construction by Pawlik et al. [12] and Walczak [16] we provide a negative answer to Problem 6.

[1] We consider k to be a fixed integer and use the notation $O_k(\cdot)$ to indicate that the constant hiding in the big O notation depends only on k.

[2] Recall that a graph is *k-quasi-planar* if it admits a drawing in which no k edges are pairwise crossing.

▶ **Theorem 7.** *There exist sets of m segments, no three of which are pairwise traversing, such that the maximum size of a pairwise nontraversing subset is $o(m)$.*

The maximum size of a subset with no two traversing segments in this construction is $O(m/\log\log m)$. It is an interesting problem to determine the maximum size of such a subset in any set of m segments no k of which are pairwise traversing. The best lower bound we were able to find was $\Omega(\sqrt{m})$

In the special case of a (bipartite) geometric graph G in which all edges cross a single line ℓ, we were able to prove Conjecture 5. In fact in this case, a linear upper bound is known even when no k edges pairwise cross [14]. However, for traversing edges we have devised a simpler proof: Denote by $\overline{e}$ the complement of an edge e on the line that supports e and observe that e_1 and e_2 are traversing if and only if $\overline{e_1}$ and $\overline{e_2}$ are disjoint. Therefore, as ℓ goes to infinity we obtain a graph with no k pairwise disjoint edges and the linear bound on its density follows from Theorem 1. This result, along with a standard divide-and-conquer argument, shows that an n-vertex geometric graph with no k pairwise traversing edges has $O_k(n \log n)$ edges, without relying on the same known bound for k-quasi-planar graphs.

Alas, we were unable to make any further progress on Conjecture 5. Still, to get a better understanding of the notion of traversing edges we reverted to simpler questions involving such edges. Recall that an embedded graph is k-plane if each of its edges is crossed at most k times. The maximum densities of n-vertex k-plane graphs for $k = 1, 2, 3, 4$ are known to be $4n-8$ [10], $5n-10$ [10], $5.5n-11$ [9], and $6n-O(1)$ [2], respectively. We considered analogue graphs with respect to traversing edges, that is, the density of *k-traversing* geometric graphs – graphs in which each edge is involved in at most k traversings. Since, by definition, these graphs are k-plane we are interested in exact bounds on their densities.

▶ **Theorem 8.** *Let G be an n-vertex 1-traversing geometric graph. Then $|E(G)| \leq \lfloor 2.5n \rfloor - 4$, if $n \geq 2$. This bound is tight.*

Note that there might be asymmetry when two edges e_1 and e_2 are traversing according to which of them contains the intersection point of the two supporting lines. Suppose that e_1 contains that point. Then we say that e_1 is *traversed* by e_2 and that e_2 is *traversing* e_1. Note that if e_1 and e_2 are crossing, then each of them is traversing and traversed by the other. Theorem 8 is in fact implied by each of following two variants.

▶ **Theorem 9.** *Let G be an n-vertex geometric graph in which each edge is traversing at most one edge. Then $|E(G)| \leq \lfloor 2.5n \rfloor - 4$, if $n \geq 2$. This bound is tight.*

▶ **Theorem 10.** *Let G be an n-vertex geometric graph in which each edge is traversed by at most one edge. Then $|E(G)| \leq \lfloor 2.5n \rfloor - 4$, if $n \geq 2$. This bound is tight.*

The upper bound $\lfloor 2.5n \rfloor - 4$ matches the maximum size of an n-vertex outer 1-plane graph [7] (an *outer* k-plane graph is a geometric k-plane graph in which the vertices are in convex position). Note that for a convex geometric graph the notions of crossing and traversing edges coincide. We only found one example of a nonconvex 1-traversing geometric graph with the maximum possible density, namely a nonconvex drawing of K_4. Call a k-traversing geometric graph *optimal* if there is no other k-traversing geometric graph with the same number of vertices and a greater number of edges.

▶ **Problem 11.** *Is it true that for every integer k there is an integer n_k such that every optimal k-traversing graph with more than n_k vertices is an outer k-plane graph?*

A possible way to provide a negative answer to this question would be to show that in some cases we get different maximum densities for the different notions of traversing. Perhaps an easier problem would be to show that the class of graphs that can be drawn such that every edge is traversing at most k other edges and the class of graphs that can be drawn such that every edge is traversed by at most k other edges are not the same.

References

1 E. Ackerman, On the maximum number of edges in topological graphs with no four pairwise crossing edges, *Discrete Comput. Geom.* **41** (2009), 365–375.

2 E. Ackerman, On topological graphs with at most four crossings per edge, *arXiv:* **1509.01932**, 2015.

3 E. Ackerman, N. Nitzan and R. Pinchasi, The maximum number of edges in geometric graphs with pairwise virtually avoiding edges, *Graphs and Combinatorics* **30** (2014), 1065–1072.

4 E. Ackerman and G. Tardos, On the maximum number of edges in quasi-planar graphs, *J. Combinatorial Theory, Ser. A.* **114** (2007), 563–571.

5 P. K. Agarwal, B. Aronov, J. Pach, R. Pollack, and M. Sharir, Quasi-planar graphs have a linear number of edges, *Combinatorica* **17** (1997), no. 1, 1–9.

6 P. Brass, W. Moser, J. Pach, *Research Problems in Discrete Geometry*, Springer, 2005.

7 W. Didimo, Density of straight-line 1-planar graph drawings, *Information Processing Letters* 113:7 (2013), 236–240.

8 J. Pach, Notes on geometric graph theory, In J. E. Goodman, R. Pollack and W. Steiger, editors, *Discrete and Computational Geometry: Papers from DIMACS special year*, volume 6 of *DIMACS series*, 273–285, AMS, Providence, RI, 1991.

9 J. Pach, R. Radoičić, G. Tardos, G. Tóth, Improving the crossing lemma by finding more crossings in sparse graphs, *Disc. Compu. Geometry*, 36:4 (2006), 527–552.

10 J. Pach and G. Tóth, Graphs drawn with few crossings per edge, *Combinatorica*, 17:3 (1997), 427–439.

11 J. Pach and J. Törőcsik, Some geometric applications of Dilworth's theorem, *Discrete Comput. Geom.* **12** (1994), no. 1, 1–7.

12 A. Pawlik, J. Kozik, T. Krawczyk, M. Lasoń, P. Micek, W. T. Trotter and B. Walczak, Triangle-free intersection graphs of line segments with large chromatic number, *J. Combinatorial Theory, Ser. B.* **105** (2014), 6–10.

13 G. Tóth, Note on geometric graphs, *J. Combinatorial Theory, Ser. A.* **89** (2000), no. 1, 126–132.

14 P. Valtr, Graph drawings with no k pairwise crossing edges, In G. D. Battista, editor, *Graph Drawing*, volume 1353 of *Lecture Notes in Computer Science*, 205–218, Springer, 1997.

15 P. Valtr, On geometric graphs with no k pairwise parallel edges, *Discrete Comput. Geom.* **19** (1998), no. 3, 461–469.

16 B. Walczak, Triangle-free geometric intersection graphs with no large independent sets, *Discrete Comput. Geom.* **53** (2015), no. 1, 221–225.

4.2 Variants of the Segment Number of a Graph

Carlos Alegría (National Autonomous University of Mexico, MX), Yoshio Okamoto (The University of Electro-Communications – Tokyo, JP), Alexander Ravsky (Pidstryhach Institute for Applied Problems of Mechanics and Mathematics, National Academy of Sciences of Ukraine – Lviv, UA), and Alexander Wolff (Universität Würzburg, DE)

When drawing a graph, a way to keep the visual complexity low is to use few geometric objects for drawing the edges. This idea is captured by the *segment number* of a graph, that is, the smallest number of line segments that together constitute a straight-line drawing of the given graph. The *arc number* of a graph is defined analogously with respect to circular-arc drawings. For a graph G, we denote its segment number by $\mathrm{seg}(G)$ and its arc number by $\mathrm{arc}(G)$. So far, both numbers have only been studied for planar graphs. Two obvious lower bounds for $\mathrm{seg}(G)$ are known [1]: the slope number of G and $\eta(G)/2$, where $\eta(G)$ is the number of odd-degree vertices of G. Dujmović et al. [1], who introduced slope and segment number, showed among others that trees can be drawn such that the optimum segment number and the optimum slope number are achieved simultaneously. In other words, any tree T admits a drawing with $\eta(T)/2$ segments and $\Delta(T)/2$ slopes, where $\Delta(T)$ is the maximum degree of T. Unfortunately, these drawings need exponential area. Therefore, Schulz [9] suggested to study the arc number of planar graphs. Among others, he showed that any n-vertex tree can be drawn on a polynomial-size grid ($O(n^{1.81}) \times n$) using at most $3n/4$ arcs.

Upper bounds for the segment number and the arc number (in terms of the number of vertices, n, ignoring constant additive terms) are known for series-parallel graphs ($3n/2$ vs. n), planar 3-trees ($2n$ vs. $11n/6$), and triconnected planar graphs ($5n/2$ vs. $2n$) [1, 9]. The upper bound on the segment number for triconnected planar graphs has been improved for the special cases of triangulations and 4-connected triangulations (from $5n/2$ to $7n/3$ and $9n/4$, respectively) by Durocher and Mondal [2]. Hültenschmidt et al. [4] provided bounds for segment and arc number under the additional constraint that vertices must lie on a polynomial-size grid. They also showed that n-vertex triangulations can be drawn with at most $5n/3$ arcs, which is better than the lower bound of $2n$ for the segment number on this class of graphs. For 4-connected triangulations, they need at most $3n/2$ arcs. Kindermann et al. [6] recently strengthened some of these results by showing that many classes of planar graphs admit non-trivial bounds on the segment number even when restricting vertices to a grid of size $O(n) \times O(n^2)$. For drawing n-vertex trees with at most $3n/4$ segments, they reduced the grid size to $n \times n$. Durocher et al. [3] showed that the segment number is NP-hard to compute, even in the special case of arrangement graphs. It is still open, however, whether the segment number is fixed-parameter tractable.

In this report, we consider several variants of the planar segment number seg that has been studied extensively. In particular, we study the *3D segment number* seg_3, which is the most obvious generalization of the planar segment number. It is the smallest number of straight-line segments needed for a crossing-free straight-line drawing of a given graph in 3D. We also study the *crossing segment number* $\mathrm{seg}_\times$ in 2D, where edges are allowed to cross, but they are not allowed to overlap or to contain vertices in their interiors. Finally, for planar graphs, we study the *bend segment number* $\mathrm{seg}_\angle$ in 2D, which is the smallest number of straight-line segments needed for a crossing-free polyline drawing of a given graph in 2D. For a given polyline drawing δ of a graph in 2D or 3D, let $\mathrm{seg}(\delta)$ be the number of straight-line segments of which the drawing δ consists.

Table 1 Overview over our results for cubic graphs. The lower and upper bounds depend on the vertex connectivity γ of the given n-vertex graph G. Note that seg and $\mathrm{seg}_\angle$ are defined only for planar graphs.

γ	$\mathrm{seg}(G)$	$\mathrm{seg}_3(G)$	$\mathrm{seg}_\angle(G)$	$\mathrm{seg}_\times(G)$
1	$\geq 5n/6$ (Prop. 4)	$\geq 5n/6$ (Prop. 4)	$\geq 5n/6$ (Prop. 4)	$\geq 5n/6$ (Prop. 4)
2		$\leq 5n/4 + 1/2$ (Thm. 5)	$\leq n+1$ (Prop. 6)	
	$\geq 3n/4$ (Prop. 8)	$\geq 5n/6$ (Prop. 7)	$\geq 3n/4$ (Prop. 8)	$\geq 3n/4$ (Prop. 8)
3	$n/2 + 3$ [5, 8]	$\leq n$ (except $K_{3,3}$; Thm. 9)		
		$\geq 9n/14$ (Prop. 10)	$\mathrm{seg}_\angle(G) = \mathrm{seg}(G)$	

Table 1 gives an overview over our results for connected ($\gamma = 1$), biconnected ($\gamma = 2$), and triconnected ($\gamma = 3$) cubic graphs. We sketch some of the proofs in Section 4.2.2. First, however, we establish some relationships between the variants of the segment number; see Section 4.2.1.

4.2.1 Relationships Between Variants of the Segment Number

▶ **Proposition 1.** *For any graph G it holds that $\mathrm{seg}_\times(G) \leq \mathrm{seg}_3(G)$.*

Proof. Let δ be a (crossing-free) straight-line drawing of G in 3D with $\mathrm{seg}(\delta) = \mathrm{seg}_3(G)$. For each triple u, v, w of three distinct vertices of G in δ let $P(u, v, w)$ be a plane spanned by the vectors $u - v$ and $w - v$ and let $\mathcal{P}$ be the set of all such planes. Choose a point A in $\mathbb{R}^3 \setminus \bigcup \mathcal{P}$ that does not lie in the xy-plane. Let δ' be the drawing that results from projecting δ parallel to the vector OA onto the xy-plane. Due to the choice of our projection, δ' may contain crossings, but no edge contains a vertex it is not incident to and no two edges overlap. Hence, $\mathrm{seg}_\times(G) \leq \mathrm{seg}_3(G)$. ◀

▶ **Proposition 2.** *There is an infinite family of planar graphs $(T_i)_{i\geq 4}$ such that T_i has i vertices and the ratios $\mathrm{seg}(T_i)/\mathrm{seg}_3(T_i)$, $\mathrm{seg}(T_i)/\mathrm{seg}_\angle(T_i)$, and $\mathrm{seg}(T_i)/\mathrm{seg}_\times(T_i)$ all tend to 2 with increasing i.*

Proof sketch. We construct the graph T_i starting from a triangulation with maximum degree 6 and $t_i = i$ vertices (and, hence, $3i - 6$ edges and $2i - 4$ faces). For example, take two triangular grids and glue their boundaries. We assume that i is even. To each vertex v of the triangulation, we attach an *i-fan*, that is, a path of length i each of whose vertices is connected to v. Now the idea of the proof is that, for every i-fan that must be drawn inside one of the interior faces, we need roughly i segments if we cannot bend edges, use crossings, or exploit 3D. Otherwise, we need only about $i/2$ segments. ◀

4.2.2 Cubic Graphs

Now we turn to cubic graphs. Consider a straight-line drawing δ of a cubic graph (in 2D or 3D). Note that there are two types of vertices; those where exactly one segment ends and those where three segments end. We call these vertices *flat vertices* and *tripods*, respectively. Let $f(\delta)$ be the number of flat vertices, and let $t(\delta)$ be the number of tripods in δ.

▶ **Lemma 3.** *For any straight-line drawing δ of a cubic graph with n vertices, $\mathrm{seg}(\delta) = 3n/2 - f(\delta) = n/2 + t(\delta)$.*

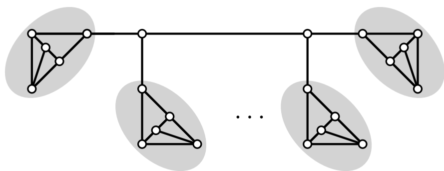


Figure 1 The graph G_k (here $k = 4$) is a caterpillar with $k - 2$ inner vertices of degree 3 where each leaf has been replaced by a copy of the 5-vertex graph H (shaded gray).

Proof. The number of "segment ends" is $3t(\delta) + f(\delta) = 3n - 2f(\delta) = n + 2t(\delta)$. The claim follows since every segment has two ends. ◀

▶ **Proposition 4.** *There is an infinite family $(G_k)_{k\geq 1}$ of connected cubic graphs such that G_k has $n_k = 6k - 2$ vertices and $\mathrm{seg}(G_k) = \mathrm{seg}_3(G_k) = \mathrm{seg}_\angle(G_k) = \mathrm{seg}_\times(G_k) = 5k - 1 = 5n_k/6 + 2/3$.*

Proof sketch. Consider the graph G_k depicted in Fig. 1 (for $k = 4$). In each gray-shaded subgraph, at least two vertices are tripods. Hence, for any drawing δ of G, $t(\delta) \geq 2k$. Now Lemma 3 yields that $\mathrm{seg}(\delta) \geq 5k - 1$. For the drawing in Fig. 1, the bound is tight. ◀

Every biconnected cubic graph G admits an st-ordering, that is, an ordering $\langle v_1, \ldots, v_n\rangle$ of the vertex set $\{v_1, \ldots, v_n\}$ of G such that for every $j \in \{2, n-1\}$ vertex v_j has at least one predecessor (that is, a neighbor v_i with $i < j$) and at least one successor (that is, a neighbor v_k with $k > j$). Using an st-ordering of the given graph, we can construct a straight-line drawing of the graph in 3D and bound the number of segments in the drawing as follows.

▶ **Theorem 5.** *For any biconnected cubic graph G with n vertices, $\mathrm{seg}_3(G) \leq 5n/4 + 1/2$.*

▶ **Proposition 6.** *For any biconnected planar cubic graph G with n vertices, it holds that $\mathrm{seg}_\angle(G) \leq n + 1$. A corresponding drawing can be found in linear time.*

Proof. We draw G using the algorithm of Liu et al. [7] that draws any planar biconnected cubic graph except the tetrahedron orthogonally with at most one bend per edge and at most $n/2 + 1$ bends in total. It remains to count the number of segments in this drawing. In any vertex exactly one segment ends; in any bend exactly two segments end. In total, this yields at most $n + 2 \cdot (n/2 + 1) = 2n + 2$ segment ends and at most $n + 1$ segments.

Concerning the special case of the tetrahedron (K_4), note that it can be drawn with five segments when bending one of its six edges. ◀

▶ **Proposition 7.** *There is an infinite family of cubic graphs $(H_k)_{k\geq 3}$ such that H_k has $n_k = 6k$ vertices, $\mathrm{seg}_3(H_k) = 5k = 5n_k/6$, and $\mathrm{seg}_\times(H_k) = 4k = 2n_k/3$.*

Proof sketch. Consider the graph H_k depicted in Fig. 2 (for $k = 4$). It is a k-cycle where each vertex is replaced by a copy of a 6-vertex graph K ($K_{3,3}$ minus an edge). The graph H_k has $n_k = 6k$ vertices and is not planar. In any 2D drawing with crossings at least one vertex in each copy of K is a tripod; in 3D at least two vertices in each copy are tripods. Now Lemma 3 yields that $\mathrm{seg}_\times(H_k) \geq 4k$ and $\mathrm{seg}_3(H_k) \geq 5k$.

Figure 2 shows that $\mathrm{seg}_\times(H_k) \leq 4k$ and, by lifting in each copy of K the white vertex that is not on the convex hull out of the drawing plane, that $\mathrm{seg}_3(H_k) \leq 5k$. ◀

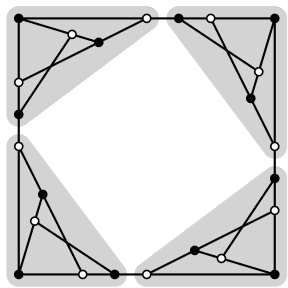


Figure 2 The cubic graph H_k (here $k = 4$) is a k-cycle whose vertices are replaced by the subgraphs in the gray shaded regions ($K_{3,3}$ minus an edge). The graph H_k has $n_k = 6k$ vertices, $\text{seg}_3(H_k) = 5n_k/6$, and $\text{seg}_\times(H_k) = 2n_k/3$.

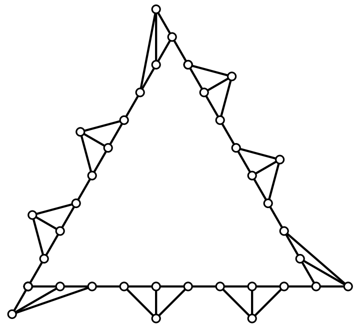


Figure 3 The planar Hamiltonian cubic graph I_k (here $k = 9$) is a k-cycle whose vertices are replaced by copies of K_4 minus an edge. The graph I_k has $n_k = 4k$ vertices and $\text{seg}(I_k) = \text{seg}_3(I_k) = \text{seg}_\angle(I_k) = \text{seg}_\times(I_k) = 3n_k/4$.

▶ **Proposition 8.** *There is an infinite family of planar cubic Hamiltonian graphs $(I_k)_{k\geq 3}$ such that I_k has $n_k = 4k$ vertices and $\text{seg}(I_k) = \text{seg}_3(I_k) = \text{seg}_\angle(I_k) = \text{seg}_\times(I_k) = 3k = 3n_k/4$.*

Proof sketch. Consider the graph I_k depicted in Fig. 3 (for $k = 9$). The proof is similar to that of the crossing case in Proposition 7. ◀

▶ **Theorem 9.** *Every triconnected cubic n-vertex graph admits a straight-line drawing in 3D with at most n segments – except $K_{3,3}$, which needs seven segments.*

Proof sketch. Partition the given graph into a perfect matching and a collection of pairwise disjoint cycles. Treat each cycle separately and draw it on a copy of the moment curve. ◀

▶ **Proposition 10.** *There is an infinite family of triconnected cubic graphs $(F_k)_{k\geq 4}$ such that F_k has $n_k = 14k$ vertices and $\text{seg}_3(F_k) = 9k = 9n_k/14$.*

Proof sketch. Let K' be the graph that results from removing one edge from $K_{3,3}$ and subdividing another edge. Now take any triconnected cubic graph with $2k$ vertices and replace each of its vertices by a copy of the 7-vertex graph K'. The resulting graph F_k has $n_k = 14k$ vertices and is not planar.

The proof that $\text{seg}_3(F_k) = 9k$ is similar to that of the 3D case in Proposition 7. ◀

Acknowledgments. We thank Günter Rote and Martin Gronemann for asking interesting questions that led to some of this research.

References

1 V. Dujmović, D. Eppstein, M. Suderman, and D. R. Wood. Drawings of planar graphs with few slopes and segments. *Comput. Geom. Theory Appl.*, 38(3):194–212, 2007. `doi:10.1016/j.comgeo.2006.09.002`.

2 S. Durocher and D. Mondal. Drawing plane triangulations with few segments. In *Proc. Canad. Conf. Comput. Geom. (CCCG'14)*, pages 40–45, 2014. URL: http://cccg.ca/proceedings/2014/papers/paper06.pdf.

3 S. Durocher, D. Mondal, R. Nishat, and S. Whitesides. A note on minimum-segment drawings of planar graphs. *J. Graph Alg. Appl.*, 17(3):301–328, 2013. `doi:10.7155/jgaa.00295`.

4 G. Hültenschmidt, P. Kindermann, W. Meulemans, and A. Schulz. Drawing planar graphs with few geometric primitives. In H. L. Bodlaender and G. J. Woeginger, editors, *Proc. 43rd Int. Workshop Graph-Theoretic Concepts Comput. Sci. (WG'17)*, volume 10520 of *LNCS*, pages 316–329. Springer, 2017. doi:10.1007/978-3-319-68705-6_24.
5 A. Igamberdiev, W. Meulemans, and A. Schulz. Drawing planar cubic 3-connected graphs with few segments: Algorithms & experiments. *J. Graph Alg. Appl.*, 21(4):561–588, 2017. doi:10.7155/jgaa.00430.
6 P. Kindermann, T. Mchedlidze, T. Schneck, and A. Symvonis. Drawing planar graphs with few segments on a polynomial grid. arXiv report, 2019. URL: https://arxiv.org/abs/1903.08496.
7 Y. Liu, P. Marchioro, and R. Petreschi. At most single-bend embeddings of cubic graphs. *Appl. Math.*, 9(2):127–142, 1994. doi:10.1007/BF02662066.
8 D. Mondal, R. I. Nishat, S. Biswas, and M. S. Rahman. Minimum-segment convex drawings of 3-connected cubic plane graphs. *J. Comb. Optim.*, 25(3):460–480, 2013. doi:10.1007/s10878-011-9390-6.
9 A. Schulz. Drawing graphs with few arcs. *J. Graph Alg. Appl.*, 19(1):393–412, 2015. doi:10.7155/jgaa.00366.

4.3 Simultaneous Graph Embedding Beyond Planarity

Patrizio Angelini (Universität Tübingen, DE), Henry Förster (Universität Tübingen, DE), Michael Hoffmann (ETH Zürich, CH), Michael Kaufmann (Universität Tübingen, DE), Stephen G. Kobourov (University of Arizona – Tucson, US), Giuseppe Liotta (University of Perugia, IT), and Maurizio Patrignani (Roma Tre University, IT)

Abstract. Simultaneous Graph Embedding asks the question whether a set of graphs $\mathcal{G}$ with shared vertex set V can be embedded in the plane such that each graph in $\mathcal{G}$ is drawn planar. We study this problem in the beyond planarity framework by allowing the graphs in $\mathcal{G}$ to have crossings between their edges as long as they respect certain crossing configurations. We call this setting `Beyond-Simultaneous`. In addition, we also study a setting called `Beyond-Union`, where we require the union of all graphs in $\mathcal{G}$ to fulfill restrictions on the crossing configurations.

We show that in setting `Beyond-Simultaneous` two planar graphs and a tree can always be realized such that each of the graphs is drawn quasiplanar, we also prove that the same holds for a 1-planar graph and a planar graph. Further, we show that in setting `Beyond-Union`, a path and a matching cannot always be embedded such that their union is k-planar for a fixed k whereas five cycles cannot always be drawn such that their union is quasiplanar.

4.3.1 Introduction

Simultaneous Graph Embedding is a family of problems where you are given a set of graphs $\mathcal{G} = \{G_1, \ldots, G_k\}$ with shared vertex set V and you are required to produce drawings $\{\Gamma_1, \ldots, \Gamma_k\}$ of them in such a way that each vertex has the same position in every Γ_i and each Γ_i satisfies certain readability properties. Usually, the readability property that is pursued while searching for a simultaneous embedding is planarity and a large body of research has been dedicated to the complexity of deciding whether a set of graphs admits such simultaneous embeddings or to determine if such embeddings always exist given the number and the types of the input graphs; for a survey refer to [6].

Simultaneous Graph Embedding has been studied both from a geometric point of view (Geometric Simultaneous Embedding – `GSE`) [5, 10] and from a topological point of view (Simultaneous Embedding with Fixed Edges – `SEFE`) [7, 8]. In particular, in `GSE`, the edges are required to be straight-line segments while in `SEFE` they can be drawn as topological curves, but the edges shared between two graphs G_i and G_j have to be drawn in the same way in Γ_i and Γ_j. In the following, we focus on the topological setting unless otherwise specified.

We study two variants of the simultaneous embedding problem in the beyond planarity framework by allowing the graphs in $\mathcal{G}$ to be drawn non-planar. In the first problem, we only restrict the crossings in each of the graphs $G \in \mathcal{G}$.

► **Problem 1** (`Beyond-Simultaneous`). *Is it possible to simultaneously embed a set of graphs $\mathcal{G}$ with shared vertex set V in the plane such that each graph $G \in \mathcal{G}$ is drawn k-(quasi)planar?*

Recall that in a k-planar drawing, each edge is crossed at most k times whereas in a k-quasiplanar drawing, there is no k-tuple of pairwise intersecting edges. Also recall, that 3-quasiplanar is often referred to as quasiplanar. In the second problem, we additionally restrict the crossings in the union of all graphs in $\mathcal{G}$.

► **Problem 2** (`Beyond-Union`). *Is it possible to simultaneously embed a set of graphs $\mathcal{G}$ with shared vertex set V in the plane such that the union graph $G_\cup = \bigcup_{G \in \mathcal{G}} G$ is drawn k-(quasi)planar?*

Note that in setting `Beyond-Union` we could also ask each $G \in \mathcal{G}$ to satisfy stronger restrictions on the crossing configurations.

In the remainder of this report, we first present preliminary results in Section 4.3.2 which will be used in our proofs. Then, we investigate the more restricted `Beyond-Union` setting in Section 4.3.3 and show very restrictive negative results. Afterwards, we show positive results in the `Beyond-Simultaneous` setting in Section 4.3.4. We conclude the report by listing open problems in Section 4.3.5.

4.3.2 Preliminaries

We make use of a result on the partially embedded planarity problem (`PEP`) which is defined as follows.

► **Problem 3** (`PEP`). *Let G be a planar graph, H a subgraph of G and $\mathcal{H}$ an embedding of H. Can G be embedded in the plane such that H is drawn with embedding $\mathcal{H}$?*

Problem `PEP` has been introduced and studied in [4] where a linear-time algorithm is presented. In particular, this algorithm is based on a characterization that we will exploit in the following.

► **Lemma 4** ([4]). *Let $(G, H, \mathcal{H})$ be an instance of `PEP` and let $\mathcal{G}$ be a planar embedding of G. $\mathcal{G}$ is a solution for $(G, H, \mathcal{H})$ if and only if the following conditions hold:*
1. *for every vertex $v \in V$, the edges incident to v in H appear in the same cyclic order in the rotation schemes of v in $\mathcal{H}$ and in $\mathcal{G}$; and*
2. *for every cycle C of H, and for every vertex v of $H \setminus C$, we have that v lies in the interior of C in $\mathcal{G}$ if and only if it lies in the interior of C in $\mathcal{H}$.*

Another important tool that we will exploit is the following theorem due to Pach and Wenger [12].

▶ **Theorem 5** ([12]). *Every planar graph on n vertices admits a planar embedding which maps each vertex to an arbitrarily prespecified distinct location and each edge to a polygonal curve with $O(n)$ bends. Further, there exists a path, whose vertices are mapped to a point set in convex position, such that in any embedding of this graph that respects the mapping of vertices to points there exists one edge with a linear number of bends.*

4.3.3 Setting `Beyond-Union`

Here, we first attempt to maintain k-planarity for a fixed k. Unfortunately, this already fails for a path and a matching.

▶ **Theorem 6.** *There exists a family of paths $\mathcal{P}$ and a family of matchings $\mathcal{M}$ such that $P \in \mathcal{P}$ and $M \in \mathcal{M}$ on n shared vertices cannot be simultaneously embedded such that their union is k-planar for any $k \in o(n/\log^2 n)$.*

Proof. To prove the theorem, we exploit a family of 3-regular graphs which is known to be not k-planar for any $k \in o(n/\log^2 n)$ [3]. Consider the hypercube graph $\mathcal{H}_d$ of dimension d. Let v be a vertex of $\mathcal{H}_d$ and let $u_1, \ldots, u_d$ be its neighbors. We replace v by a cycle $(v_1, \ldots, v_d)$ such that v_i is connected to u_i for $1 \leq i \leq d$. By repeating this procedure for all vertices of $\mathcal{H}_d$, we obtain the cube connected cycle graph CCC_d of dimension d which is a cubic graph on $n = d \cdot 2^d$ vertices.

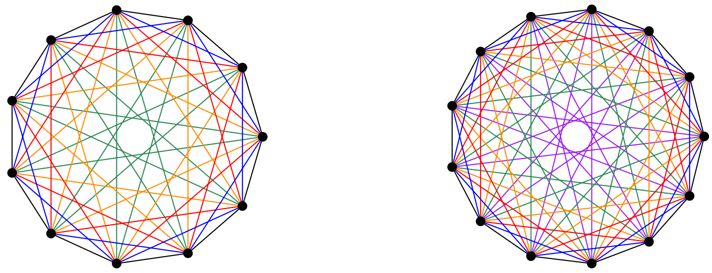
It is known that the crossing number $cr(CCC_d) = \Omega(4^d)$ [13]. Hence, the average number of crossings per edge is $\Omega(2^d/d) = \Omega(n/\log^2 n)$. Further, it is known, that CCC_d is a Hamiltonian graph [11]. Hence, CCC_d is composed of a cycle (the Hamiltonian cycle) and a matching. To obtain the the statement of the theorem, it is possible to show that removing one edge does not alter the arguments. ◀

In addition, when further restricting each of the subgraphs to be drawn planar, there exist even two paths that cannot be drawn with a sublinear number of crossings per edge. We state this fact in the following Theorem, which can be proved with the same reasoning used to prove Lemma 10 and Theorem 8 of a recent manuscript [9]. We repeat the argument for completeness.

▶ **Theorem 7.** *There exist two families of paths $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $P_1 \in \mathcal{P}_1$ and $P_2 \in \mathcal{P}_2$ on n shared vertices cannot be simultaneously embedded such that their union is k-planar for any $k \in o(n)$ if P_1 and P_2 are embedded planar.*

Proof. Assume for contradiction that every two paths P_1 and P_2 on n shared vertices admit a simultaneous embedding such that both are drawn planar and that their union is $o(n)$-planar. Since we have a simultaneous embedding we can construct a drawing on a point set so that P_1 is drawn monotone and straight-line and each edge of P_2 has as many bends as it has intersections with P_1. In such a drawing, P_1 describes a convex point set for P_2. Hence, every path P_2 admits a planar drawing on every point set such that each of its edges is only bent $o(n)$ times. This is a contradiction to Theorem 5. ◀

In the next step, we shift our attention to quasiplanar embeddings of unions of graphs. Since the union of two planar graphs has thickness two, two planar graphs can always be simultaneously embedded such that their union is quasiplanar [12]. We show however, that even for a few cycles quasiplanarity cannot be maintained:



(a)

(b)

Figure 4 (a) K_{11} is the union of five cycles. (b) K_{13} is the union of six cycles.

▶ **Theorem 8.** *There exist five cycles $C_1 = (V, E_1)$, $C_2 = (V, E_2)$, $C_3 = (V, E_3)$, $C_4 = (V, E_4)$ and $C_5 = (V, E_5)$ on $|V| = 11$ vertices which cannot be simultaneously embedded such that their union is simple quasiplanar. In addition, there exist six cycles $C'_1 = (V', E'_1), \ldots, C'_6 = (V', E'_6)$ on $|V'| = 13$ vertices which cannot be simultaneously embedded such that their union is quasiplanar.*

Proof. Consider K_{11}. It has $\binom{11}{2} = 55$ edges. Since simple quasiplanar graphs have density $6.5n - 20$ [2], K_{11} cannot be quasiplanar. Further K_{11} is the union of the following five cycles:

- $C_1 = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11})$
- $C_2 = (v_1, v_3, v_5, v_7, v_9, v_{11}, v_2, v_4, v_6, v_8, v_{10})$
- $C_3 = (v_1, v_4, v_7, v_{10}, v_2, v_5, v_8, v_{11}, v_3, v_6, v_9)$
- $C_4 = (v_1, v_5, v_9, v_2, v_6, v_{10}, v_3, v_7, v_{11}, v_4, v_8)$
- $C_5 = (v_1, v_6, v_{11}, v_5, v_{10}, v_4, v_9, v_3, v_8, v_2, v_7)$

For an illustration, refer to Fig. 4a.

Similar arguments for K_{13} apply for the non-simple case; see Fig. 4b. ◀

4.3.4 Setting `Beyond-Simultaneous`

▶ **Theorem 9.** *Let $G_1 = (V, E_1)$ and $G_3 = (V, E_3)$ be planar graphs and $T_2 = (V, E_2)$ be a tree with shared vertex set V. Then G_1, T_2 and G_3 can be simultaneously embedded in the plane such that G_1 and T_2 are drawn planar and G_3 is drawn quasiplanar.*

Proof. Our strategy is to construct first a simultaneous embedding of G_1 and T_2 and then of the resulting graph with G_3. When constructing a simultaneous embedding of two graphs, we consider the graph induced by their common edges as a subgraph for which we want to satisfy the conditions of Lemma 4. Since this subgraph is always a forest due to the fact that T_2 is a tree, Condition 2 is always satisfied. For Condition 1, we already take into account the conditions imposed by the planar embedding of G_3 to the embedding of T_2 while constructing the simultaneous embedding of G_1 and T_2. Namely, we first embed G_1 in the plane such that G_1 is planar. Then, we add the edges $E_2 \setminus E_1$ without intersecting an edge of $E_2 \cap E_1$. Finally, we draw $G'_3 = (V, E_3 \setminus E_1)$ planar. Hence, edges of G'_3 can only intersect edges of G_3 which are part of G_1 resulting in a quasiplanar drawing of G_3.

We draw the remaining edges of T_2 without intersecting $E_2 \cap E_1$ as follows: We observe, that $(V, E_2 \cap E_1)$ is a planar drawn subforest of T_2. Since T_2 is a tree any of its embeddings is planar. Hence, Condition 1 stated in Lemma 4 is trivially fulfilled. Moreover, for edges

$E_3 \cap (E_2 \setminus E_1)$, we can chose an ordering around each vertex such that it corresponds to a planar embedding $\mathcal{G}'_3$ of planar graph G'_3. The remaining edges of T_2 can be arbitrarily embedded.

When embedding G'_3, we already have embedded edges $E_3 \cap (E_2 \setminus E_1)$. Since we have chosen the embedding of these edges such that they respect the proper planar embedding $\mathcal{G}'_3$ of G'_3, Condition 1 stated in Lemma 4 is again fulfilled. Thus, we can extend the partial embedding of G'_3 to planar embedding $\mathcal{G}'_3$. ◀

▶ **Corollary 10.** *Let $G_1 = (V, E_1)$ be a 1-planar graph and $G_2 = (V, E_2)$ be a planar graph. Then G_1 and G_2 can be simultaneously embedded in the plane such that both G_1 and G_2 are drawn quasiplanar.*

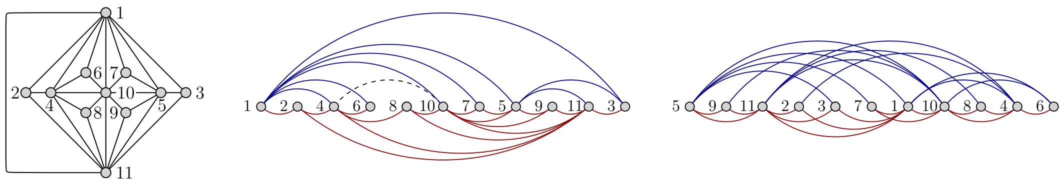
Proof. Since G_1 is 1-planar, it is the union of a planar graph G'_1 and a forest F_1 with shared vertex set [1]. By Theorem 9, there exists a simultaneous embedding of G'_1, F_1, and G_2 such that G'_1 and F_1 are drawn planar and G_2 is drawn quasiplanar. Since the union of two planar drawings with same vertex set is quasiplanar, G_1 is drawn quasiplanar, as well. ◀

4.3.5 Open Problems

- Our results show that asking for k-planarity is too restrictive in setting `Beyond-Union`, while for quasiplanarity we have a counterexample for a set of five cycles. What about the quasiplanarity of the union of a small set of paths (e.g. 3 or 4)?
- In the setting `Beyond-Simultaneous`, we ask what is the smallest set of graph families which cannot be always simultaneously embedded so that each graph is quasiplanar. In particular, can three planar graphs (or two 1-planar graphs, or four paths) always be simultaneously embedded such that each one is drawn quasiplanar?
- How difficult is it to test whether a given set of graphs admits a `Beyond-Union` or `Beyond-Simultaneous` embedding?

References

1. E. Ackerman. A note on 1-planar graphs. *Discrete Applied Mathematics*, 175:104–108, 2014.
2. E. Ackerman and G. Tardos. On the maximum number of edges in quasi-planar graphs. *Journal of Combinatorial Theory, Series A*, 114(3):563 – 571, 2007.
3. P. Angelini, M. A. Bekos, M. Kaufmann, and T. Schneck. Low-degree graphs beyond planarity. In T. Biedl and A. Kerren, editors, *Proc. of 26th International Symposium on Graph Drawing and Network Visualization (GD 2018)*, volume 11282 of *LNCS*, pages 630–632, 2018.
4. P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, Apr. 2015.
5. P. Angelini, M. Geyer, M. Kaufmann, and D. Neuwirth. On a tree and a path with no geometric simultaneous embedding. *J. Graph Algorithms Appl.*, 16(1):37–83, 2012.
6. T. Bläsius, S. G. Kobourov, and I. Rutter. Simultaneous embedding of planar graphs. In R. Tamassia, editor, *Handbook on Graph Drawing and Visualization.*, pages 349–381. Chapman and Hall/CRC, 2013.
7. T. Bläsius and I. Rutter. Simultaneous pq-ordering with applications to constrained embedding problems. *ACM Trans. Algorithms*, 12(2):16:1–16:46, 2016.
8. P. Braß, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. *Comput. Geom.*, 36(2):117–130, 2007.



(a) **(b)** **(c)**

Figure 5 (a) The Goldner-Harary graph; (b) a 3-stack layout; and (c) a 2-queue layout.

9 E. Di Giacomo, L. Gasieniec, G. Liotta, and A. Navarra. On the curve complexity of 3-colored point-set embeddings. Submitted Manuscript.

10 A. Estrella-Balderrama, E. Gassner, M. Jünger, M. Percan, M. Schaefer, and M. Schulz. Simultaneous geometric graph embeddings. In S. Hong, T. Nishizeki, and W. Quan, editors, *Graph Drawing, 15th International Symposium, GD 2007, Sydney, Australia, September 24-26, 2007. Revised Papers*, volume 4875 of *Lecture Notes in Computer Science*, pages 280–290. Springer, 2007.

11 L.-H. Hsu, T.-Y. Ho, Y.-H. Ho, and C.-W. Tsay. Cycles in cube-connected cycles graphs. *Discrete Applied Mathematics*, 167:163 – 171, 2014.

12 J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17(4):717–728, Dec 2001.

13 O. Sýkora and I. Vrťo. On crossing numbers of hypercubes and cube connected cycles. *BIT Numerical Mathematics*, 33(2):232–237, Jun 1993.

4.4 On Linear Layouts of Planar and k-Planar Graphs

Michael Bekos (Universität Tübingen, DE), Giordano Da Lozzo (Roma Tre University, IT), Vida Dujmović (University of Ottawa, CA), Fabrizio Frati (Roma Tre University, IT), Martin Gronemann (Universität Köln, DE), Tamara Mchedlidze (KIT – Karlsruher Institut für Technologie, DE), Fabrizio Montecchiani (University of Perugia, IT), Martin Nöllenburg (TU Wien, AT), Sergey Pupyrev (Facebook – Menlo Park, US), and Chrysanthi Raftopoulou (National Technical University of Athens, GR)

4.4.1 Introduction and Related Work

A *linear layout* of a graph G consists of a linear order of the vertices of G and of a partition of the edges of G that satisfies a certain property and whose size is given. In what follows, we study two well-known types of linear layouts, namely stack and queue layouts. Moreover, we consider linear layouts in which these two types are mixed.

4.4.1.1 Stack Layouts

We first consider stack layouts, also known as *book embeddings*, which form a fundamental problem in graph theory (see, e.g., [7] for an overview). In a stack layout, the edge partition is such that no two edges of the same part, which is called *stack*, cross; see Figure 5b. The *stack number*, or *book thickness*, of a graph is the smallest number of stacks that are required by any stack layout of the graph.

Problems on stack layouts are mainly classified into two categories based on whether the graph to be embedded is planar or not. For non-planar graphs, it is known that there exist graphs on n vertices that have stack number $\Theta(n)$, e.g., the stack number of the complete graph K_n is $\lceil n/2 \rceil$ [6]. Sublinear stack number is achieved by graphs with, e.g., subquadratic number of edges [28], subquadratic genus [27] or sublinear treewidth [13]. Constant stack number is achieved by graphs that are, e.g., in a minor-closed family [8] or in a bounded-treewidth family [21]. Another class of non-planar graphs that was proved to have constant stack number is the class of 1-planar graphs [4].

For planar graphs, a remarkable result is due to Yannakakis, who back in 1986 proved that for any planar graph four stacks suffice [34]. However, more restricted subclasses of planar graphs allow layouts with fewer stacks. Bernhart and Kainen [6] showed that the graphs which can be embedded using a single stack are the outerplanar graphs, while the graphs which can be embedded using two stacks are the subhamiltonian ones.

It is known that not all planar graphs are subhamiltonian and the corresponding decision problem whether a maximal planar graph is Hamiltonian (and therefore admits a 2-stack layout) is $\mathcal{NP}$-complete [32]. However, several subclasses of planar graphs are known to be Hamiltonian or subhamiltonian, see, e.g., [3, 9, 10, 22, 25, 29].

4.4.1.2 Queue Layouts

A *queue layout* is a linear layout such that no two independent edges that are assigned to the same part, which is called a *queue*, are nested [24]; see Figure 5c for an illustration. The *queue number* of a graph G is the minimum number of queues in any queue layout of G.

It is known that there exist non-planar graphs on n vertices with $\Theta(n)$ queue number, for example, the queue number of the complete graph K_n is $\lfloor n/2 \rfloor$ [24]. Moreover, there exist graphs of bounded degree that may require arbitrarily many queues [33]. Among the graphs having sublinear queue number are those with a subquadratic number of edges [23], and those that belong to any minor-closed graph family [17]. Bounded queue number is achieved by all graphs of bounded treewidth [16]. In particular, a graph with treewidth w has queue number $\mathcal{O}(2^w)$ [31]. Improved bounds (linear in the parameter) are known for graphs of bounded pathwidth [16], bounded track number [19], bounded bandwidth [23], or bounded layered pathwidth [2]; for a survey we refer the reader to [17].

A rich body of literature focuses on planar graphs. In fact, it is known that the graphs that admit queue layouts with only one queue are the arched-level planar graphs [24], which are planar graphs with at most $2n-3$ edges over n vertices (note that testing whether a graph is arched-level planar is $\mathcal{NP}$-complete [23]). Trees are arched-level planar and therefore have queue number one [24]. Outerplanar graphs have queue number at most two [23], Halin graphs and series-parallel graphs have queue number at most three [20, 30], and planar 3-trees have queue number at most five [1]. Back in 1992, Heath, Leighton and Rosenberg [23] conjectured that every planar graph has bounded queue number. Notably, this conjecture has been an open problem for almost three decades. Recently, the conjecture was settled in the positive first for planar graphs with bounded degree [5, 18], and subsequently for general planar graphs [15], thus improving the previous logarithmic and poly-logarithmic upper bounds [2, 11, 14]. On the other hand, the best-known lower bound is due to a family of planar 3-trees that require four queues [1].

4.4.2 Problems and Progress

In what follows we give a high-level description of the problems we studied and of the progress we made for them. In particular, we mainly focused on two research problems: linear layouts of directed planar graphs and nonplanar graphs that can be drawn with few crossings per edge.

4.4.2.1 Upward Planar Graphs

An *upward stack (queue) layout* of a directed graph G is a stack (queue) layout of G such that the linear ordering of the vertices is a linear extension of the partial order induced by the directions on the edges of G; that is, for any edge directed from a vertex u to a vertex v, we have that u precedes v in the linear ordering. We consider upward planar graphs, that is, planar directed graphs that can be drawn without crossings and such that each edge is a y-monotone curve from its source to its target. It is a longstanding open question to determine the asymptotic behavior of the upward stack number of upward planar graphs. Surprisingly, the best known bounds are only the trivial ones, $\mathcal{O}(n)$ and $\Omega(1)$. Contrastingly, it is known that the upward queue number of upward planar graphs is $\Theta(n)$ in the worst case.

During the Dagstuhl seminar, we proved that every n-vertex upward planar graph has a *mixed* layout with $\mathcal{O}(\sqrt{n})$ stacks and $\mathcal{O}(\sqrt{n})$ queues. We proved that this bound is tight if the vertex ordering is fixed in advance. We also proved that $\mathcal{O}(\log n)$ stacks are enough to construct stack layouts of n-vertex upward outerplanar graphs. Constant bounds can be achieved for upward outerplanar st-graphs and upward outerplanar single-source graphs.

4.4.2.2 k-Planar Graphs

A graph is *k-planar*, for a positive integer k, if it can be drawn in the plane such that each edge is crossed at most k times (see [12, 26] for surveys). Recall that every 1-planar graph admits a stack layout with a constant number of stacks [4]. Moreover, for a fixed value of k, every k-planar graph admits a queue layout with a constant number of queues [15].

During the Dagstuhl seminar, we sketched a proof that every graph that admits a drawing in the plane such that the uncrossed edges form a biconnected planar drawing in which each face has length at most ℓ admits a stack layout with a number of stacks that depends polynomially in ℓ and that does not depend on the size of the graph. Observe that any such a graph is also k-planar, where $k \leq \frac{\ell^2}{4}$.

4.4.3 Open Problems

The main objectives for our research are the following open problems.

- What is the asymptotic behavior of the upward stack number of n-vertex upward planar graphs? The question is interesting even for n-vertex upward planar graphs without transitive edges.
- What is the largest integer k such that every directed acyclic graph whose underlying graph has treewidth at most k has upward stack number in $O(1)$? We proved that $k \leq 2$; further, it is known that $k \geq 1$. We conjecture that $k = 2$; this strengthens a conjecture of Heath, Pemmaraju and Trenk on the upward stack number of directed outerplanar graphs. The above question is interesting even for upward planar graphs whose underlying graph has treewidth at most k, where we are not aware of any upper bound on k.

- Establish a worst-case optimal upper bound for the stack number of general k-planar graphs, ideally $\mathcal{O}(k)$.
- Establish upper bounds for the stack number of other families of nonplanar graphs, such as fan-planar graphs, fan-crossing-free graphs and k-quasiplanar graphs (see [12] for definitions and results about these families of graphs).

References

1 Jawaherul Md. Alam, Michael A. Bekos, Martin Gronemann, Michael Kaufmann, and Sergey Pupyrev. Queue layouts of planar 3-trees. In Therese C. Biedl and Andreas Kerren, editors, *Graph Drawing and Network Visualization*, volume 11282 of *LNCS*, pages 213–226, Cham, 2018. Springer. doi:10.1007/978-3-030-04414-5_15.

2 Michael J. Bannister, William E. Devanny, Vida Dujmović, David Eppstein, and David R. Wood. Track layouts, layered path decompositions, and leveled planarity. *Algorithmica*, Jul 2018. doi:10.1007/s00453-018-0487-5.

3 M. Bekos, M. Gronemann, and C. N. Raftopoulou. Two-page book embeddings of 4-planar graphs. In *STACS*, volume 25 of *LIPIcs*, pages 137–148. Schloss Dagstuhl, 2014.

4 M. A. Bekos, T. Bruckdorfer, M. Kaufmann, and C. N. Raftopoulou. The book thickness of 1-planar graphs is constant. In N. Bansal and I. Finocchi, editors, *ESA*, volume 9294 of *LNCS*, pages 130–141. Springer, 2015.

5 Michael A. Bekos, Henry Förster, Martin Gronemann, Tamara Mchedlidze, Fabrizio Montecchiani, Chrysanthi N. Raftopoulou, and Torsten Ueckerdt. Planar graphs of bounded degree have constant queue number. *CoRR*, abs/1811.00816, 2018. Accepted at STOC 2019. URL: http://arxiv.org/abs/1811.00816.

6 Frank Bernhart and Paul C. Kainen. The book thickness of a graph. *J. Comb. Theory, Series B*, 27(3):320–331, 1979.

7 T. Bilski. Embedding graphs in books: a survey. *IEEE Proceedings of Computers and Digital Techniques*, 139(2):134–138, 1992.

8 R. Blankenship. *Book Embeddings of Graphs*. PhD thesis, Louisiana State University, 2003.

9 Fan R. K. Chung, Frank Thomson Leighton, and Arnold L. Rosenberg. Embedding graphs in books: A layout problem with applications to VLSI design. *SIAM J. Discrete Math.*, 8(1):33–58, 1987.

10 G. Cornuéjols, D. Naddef, and W. Pulleyblank. Halin graphs and the travelling salesman problem. *Mathematical Programming*, 26(3):287–294, 1983.

11 Giuseppe Di Battista, Fabrizio Frati, and János Pach. On the queue number of planar graphs. *SIAM J. Comput.*, 42(6):2243–2285, 2013. doi:10.1137/130908051.

12 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. URL: https://dl.acm.org/citation.cfm?id=3301281.

13 V. Dujmović and D. Wood. Graph treewidth and geometric thickness parameters. *Discrete Computational Geomentry*, 37(4):641–670, 2007.

14 Vida Dujmović. Graph layouts via layered separators. *J. Comb. Theory, Ser. B*, 110:79–89, 2015. doi:10.1016/j.jctb.2014.07.005.

15 Vida Dujmović, Gwenaël Joret, Piotr Micek, Pat Morin, Torsten Ueckerdt, and David R. Wood. Planar Graphs have Bounded Queue-Number. *CoRR*, abs/1904.04791, 2019. URL: http://arxiv.org/abs/1904.04791.

16 Vida Dujmović, Pat Morin, and David R. Wood. Layout of graphs with bounded tree-width. *SIAM J. Comput.*, 34(3):553–579, 2005. doi:10.1137/S0097539702416141.

17 Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-closed graph classes with applications. *J. Comb. Theory, Ser. B*, 127:111–147, 2017. doi:10.1016/j.jctb.2017.05.006.

18 Vida Dujmović, Pat Morin, and David R. Wood. Queue layouts of graphs with bounded degree and bounded genus. *CoRR*, abs/1901.05594, 2019. URL: http://arxiv.org/abs/1901.05594.

19 Vida Dujmović and David R. Wood. Stacks, queues and tracks: Layouts of graph subdivisions. *Discrete Math. Theor. Comput. Sci.*, 7(1):155–202, 2005. URL: http://dmtcs.episciences.org/346.

20 Joseph L. Ganley. Stack and queue layouts of Halin graphs, 1995. Manuscript.

21 Joseph L. Ganley and Lenwood S. Heath. The pagenumber of k-trees is $O(k)$. *Discrete Applied Mathematics*, 109(3):215–221, 2001.

22 Lenwood S. Heath. *Algorithms for Embedding Graphs in Books*. PhD thesis, University of N. Carolina, 1985.

23 Lenwood S. Heath, Frank Thomson Leighton, and Arnold L. Rosenberg. Comparing queues and stacks as machines for laying out graphs. *SIAM J. Discrete Mathematics*, 3(5):398–412, 1992.

24 Lenwood S. Heath and Arnold L. Rosenberg. Laying out graphs using queues. *SIAM J. Comput.*, 21(5):927–958, 1992. doi:10.1137/0221055.

25 Paul C. Kainen and Shannon Overbay. Extension of a theorem of Whitney. *Applied Mathematics Letters*, 20(7):835–837, 2007.

26 Stephen G. Kobourov, Giuseppe Liotta, and Fabrizio Montecchiani. An annotated bibliography on 1-planarity. *Computer Science Review*, 25:49–67, 2017. doi:10.1016/j.cosrev.2017.06.002.

27 S. Malitz. Genus g graphs have pagenumber $O(\sqrt{q})$. *Journal of Algorithms*, 17(1):85–109, 1994.

28 S. Malitz. Graphs with e edges have pagenumber $O(\sqrt{E})$. *Journal of Algorithms*, 17(1):71–84, 1994.

29 T. Nishizeki and N. Chiba. *Planar Graphs: Theory and Algorithms*, chapter 10. Hamiltonian Cycles, pages 171–184. Dover Books on Mathematics. Courier Dover Publications, 2008.

30 S. Rengarajan and C. E. Veni Madhavan. Stack and queue number of 2-trees. In *COCOON*, volume 959 of *LNCS*, pages 203–212, Berlin, Heidelberg, 1995. Springer. doi:10.1007/BFb0030834.

31 Veit Wiechert. On the queue-number of graphs with bounded tree-width. *Electr. J. Comb.*, 24(1):P1.65, 2017. URL: http://www.combinatorics.org/ojs/index.php/eljc/article/view/v24i1p65.

32 Avi Wigderson. The complexity of the Hamiltonian circuit problem for maximal planar graphs. Technical Report TR-298, EECS Department, Princeton University, 1982.

33 David R. Wood. Bounded-degree graphs have arbitrarily large queue-number. *Discrete Math. Theor. Comput. Sci.*, 10(1), 2008. URL: http://dmtcs.episciences.org/434.

34 Mihalis Yannakakis. Embedding planar graphs in four pages. *J. Comput. Syst. Sci.*, 38(1):36–67, 1989. doi:10.1016/0022-0000(89)90032-9.

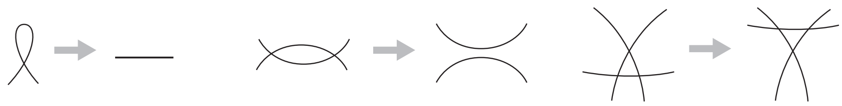


Figure 6 Homotopy moves $1 \to 0$, $2 \to 0$, and $3 \to 3$.

4.5 Monotone Untangling of Graph Drawings

María del Pilar Cano Vila (UPC – Barcelona, ES), Peter Eades (The University of Sydney, AU), Radoslav Fulek (IST Austria – Klosterneuburg, AT), Seok-Hee Hong (The University of Sydney, AU), Linda Kleist (TU Braunschweig, DE), Anna Lubiw (University of Waterloo, CA), Günter Rote (FU Berlin, DE), Ignaz Rutter (Universität Passau, DE), Leonie Ryvkin (Ruhr-Universität Bochum, DE), and Csaba D. Tóth (California State University – Northridge, US)

Abstract. Given a planar graph drawn in the plane with edge crossings, our goal is to untangle it to a crossing-free drawing using a sequence of moves that never increase the number of crossings. We consider two types of moves: continuous *homotopy moves*; and the more general *edge moves* that remove and redraw one edge at a time. We call a move *monotone* if it does not increase the number of crossings. Thus our goal is to untangle the graph drawing using monotone homotopy moves or monotone edge moves.

4.5.1 Our Results

1. With homotopy moves, if the tangled drawing has been created from a planar drawing with homotopy moves that never move an edge across a vertex, then the drawing can be untangled using monotone homotopy moves that never move an edge across a vertex.
2. With monotone edge moves we can untangle any drawing of a cactus graph, and we can untangle any drawing of a *banana cactus* graph if the drawing has a planar rotation system.
3. Not every drawing of a planar graph can be untangled with monotone edge moves.

4.5.2 Background and Concepts

In one well-studied version of untangling a straight-line graph drawing, the goal is to move as few vertices as possible in order to get a planar straight-line drawing, see [9, 1] and references therein. In this version, the vertices are re-positioned all at once. By contrast, we fix the vertices and consider a sequence of incremental changes to the curves representing the edges.

There is considerable work on untangling a curve or a set of curves using incremental homotopy moves. Homotopy moves, which are the "shadows" of the classical Reidemeister moves, are defined as follows:

These moves are monotone, but the reversals of $1 \to 0$ and $2 \to 0$ are not.

Our preliminary understanding of the relevant background work is as follows. An algorithm to simplify any planar closed curve using at most $O(n^2)$ monotone homotopy moves is implicit in Steinitz's proof [10, 11] that every 3-connected planar graph is the 1-skeleton of a convex polyhedron. For more information, see [5, 7, 4, 2]. Chang and Erickson [2] improved the number of moves to a tight bound of $\Theta(n^{3/2})$ but at the expense of losing monotonicity. For monotone moves, no bound better than $O(n^2)$ is known.

One of Steinitz's basic ideas was extended by Hass and Scott [6, 7] to Theorem 3 (stated below), which has been used in many subsequent works and which we will also use.

For generalizations to multiple curves, tangles, and other surfaces (with boundary and/or of higher genus), see [2] and the references therein. Chang's thesis [3] is an excellent resource.

Homotopy moves, and in particular, the result of Hass and Scott, have been applied to graph drawings, for example in the work by Kynčl [8] on simple realizability of complete abstract topological graphs. Proofs of versions of the Hanani-Tutte theorem may also be relevant.

4.5.3 Untangling via Monotone Homotopy Moves

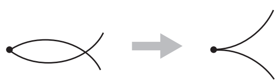
We may interpret a drawing of the graph G as a set of curves on the punctured plane, where each curve starts and ends at a boundary component. With this interpretation, two curves are homotopic if one can be continuously deformed to the other in the punctured plane (i.e., this replaces the condition that we never move an edge across a vertex). In order to deal with multiple curves (edges), we allow the homotopy move shown in Figure 7a.

▶ **Theorem 1.** *Let D and D^* be two drawings of a planar graph G such that D^* is plane, the vertex positions coincide, and for each edge e of G the curve representing e in D is homotopic to the curve representing e in D^*.*

Then D can be transformed into D^* by a sequence of monotone homotopy moves. The number of moves is at most $k + \frac{1}{4}k^2$ where k is the number of crossings in D.

We start with some definitions: A curve is *simple* if it has no self-intersections. A *loop* is a section of a curve such that its endpoints coincide and it has no other self-intersections, i.e., it is simple.

A *lens* consists of two sections of curves (either two disjoint sections of the same curve or sections of distinct curves) such that each is simple and connects two distinct points; moreover, between the two sections there exist no other intersections. See Fig. 7b.



(a) A $1 \to 0$ homotopy move in the presence of a vertex.

(b) An empty loop and two empty lenses.

Figure 7

Note that a loop or lens forms a simple closed curve which has an interior and an exterior. We say that a loop is *empty* if neither its interior nor the loop itself contains a vertex; a lens is *empty* if its interior does not contain any vertex and the lens is incident to at most one vertex placed on an intersection point of the two sections (of two different curves) of the lens. We call a loop or a lens *clean* with respect to a set of curves if it is not involved in any crossing (except for the one crossing of a loop and the two crossings of the lens by definition). Note that a clean lens does not need to be empty; it may contain many vertices in its interior.

The idea of our procedure is to untangle empty lenses until we arrive at a crossing-free drawing. To do so, we first show that a non-simple curve (edge) guarantees the existence of an empty loop or empty lens (Lemma 2). Moreover, two simple and intersecting curves guarantee the existence of an empty lens (Lemma 4). Finally, we show that given an empty lens or empty loop, there exists a sequence of monotone homotopy moves that removes the empty lens or loop and reduces the number of crossings (Lemma 5).

▶ **Lemma 2.** *If a curve in D representing an edge of G (a curve homotopic to a simple curve) is not simple then it contains an empty loop or an empty lens (not incident to any vertex).*

Lemma 2 follows from a theorem of Hass and Scott. We state their theorem in the original language of the paper: an *embedded 1-gon* is an empty loop and an *embedded 2-gon* is an empty lens; two arcs between vertices are *homotopic rel boundary* if they are homotopic (in the punctured plane).

▶ **Theorem 3** (Hass and Scott [6], Theorem 2.1)**.** *Let f be a general position arc on a surface F such that f is homotopic rel boundary to a simple arc g on F, but f is not simple. Then, the arc f has an embedded 1-gon or 2-gon.*

When we overlay D and D^*, every edge is represented by two curves which together form a closed curve that is incident to two vertices. We call the curves from D *curvy* and the curves from D^* *straight*.

▶ **Lemma 4.** *Let C_1 and C_2 be two simple closed curves that contain no vertex and such that each curve C_i is incident to two distinct vertices that split it into two parts, the* straight *and the* curvy *part. If C_1 and C_2 intersect but their straight parts do not intersect, then (parts of) C_1 and C_2 form an empty lens (that may be incident to one vertex).*

Proof Sketch. Let e be a part of C_1 that intersects C_2. We keep track of the sequence of intersections with the parts of C_2 by a word over the alphabet $\{s, c\}$, where s represents an intersection with the straight part and c an intersection with the curvy part of C_2. First we consider the case that the vertices of C_1 and C_2 are distinct. Thus, in particular, e starts and ends outside of C_2. Note that a subword ss or cc represents a lens (since both curves are simple) which is empty (since it is contained in the interior of C_2 which contains no vertices). Assume for the sake of a contradiction that any two consecutive letters are different in the word. Note that the word has even length, otherwise C_2 contains a vertex of e. Thus the word has the form $(sc)^k$ or $(cs)^k$ for some $k \in \mathbb{N}$. Since e intersects s, it must be the curvy part of C_1. Consequently, $C_1 - e$ is the straight part of C_1, and does not intersect s by assumption.

Analogous argument for C_2 implies that all intersections between C_1 and C_2 are in their curvy parts. Consequently, the word representing the intersections of e is s^k for some $k \in \mathbb{N}$, which is a contradiction.

Now, we consider the case that C_1 and C_2 share a vertex v. If both edges of C_1 start and end outside of C_2, the above argument yields an empty lens. Thus, at least one edge e of C_1 starts inside C_2 at v. Clearly, it ends outside of C_2 since its endvertex is not incident to C_2 and C_2 does not contain any vertex. The first intersection point on e from v certifies an empty lens; as before, it is a lens since the curves are simple and it is empty since it is contained in C_2. ◀

Remark: Note that the fact that no two straight parts intersect is necessary for the existence of a lens, see Fig. 8a.

(a) An example without a lens. However, the two straight parts intersect.

(b) A 1-0-move is necessary to change the rotation system.

Figure 8

▶ **Lemma 5.** *If there exists an empty loop or an empty lens (in the union of D and D^*) of one or two curves (possibly with one vertex), then there exists a sequence of monotone homotopy moves to decrease the number of crossings of the set of curves.*

Note that a monotone homotopy move never introduces a new pair of crossing curves; hence even the (multi-)set of crossings is monotonically non-increasing. Thus artificial/imaginary crossings between curves of D and D^* will never make up for real crossings. The proof of Lemma 5 is similar to a result by Hass and Scott [7, Lemma 2.6], but we also handle the case that one crossing of the lens is a vertex.

Proof Sketch. Suppose we are given an empty loop. We may assume that it contains neither a loop nor a lens in its interior; otherwise we consider the minimal such loop or lens, which would be empty and not contain any vertex on the boundary. Thus this loop is clean and the loop can be removed with a 1-0-move.

Suppose we are given an empty lens. Without loss of generality we consider the case with precisely one vertex (formed by different curves); the case with no vertex can be handled by inserting an artificial vertex on one of the two intersection points. We may assume that the lens contains no loop; otherwise we take a minimal loop which is empty and has no vertex on the boundary. We may further assume that it contains no lens, since every contained lens is empty and not incident to the vertex since its sections belong to different curves (or the vertex is artificial). By definition, the lens consists of two parts which meet at a (artificial) vertex v and a further intersection point p. Note that every section of a curve intersecting the interior of the lens connects the two parts of the lens. We *clean* the lens by moving the crossings inside the lens outside, by 3-3-moves. (Here we use the fact that any arrangement of chords in a circle where at least two chord cross contains at least two triangles incident to the circle. One of these triangles is not incident to the vertex on the loop and a crossing can be moved outside the loop by a 3-3-move. Consequently, there is a linear order of the edges of the arrangement in the interior of the lens. They can be moved outside the lens one by one with 3-3-moves over p in this order. Thus, we have a clean and empty lens, which can be removed via a 2-0 or 1-0 move. ◀

With these lemmas at hand, we are ready to prove the theorem.

Proof of Theorem 1. As long as the drawing has crossings, Lemmas 2 and 4 guarantee the existence of an empty loop or lens. Given an empty loop or lens, there exists a sequence of monotone homotopy moves to reduce the number of crossings by Lemma 5. Thus, in the end, we have transformed $D + D^*$ into a crossing-free drawing $\tilde{D} + \tilde{D}^*$. Since $\tilde{D} + \tilde{D}^*$ is crossing-free, $\tilde{D}$ and $\tilde{D}^+$ have the same rotation system and are thus equivalent drawings. Moreover, since the set of crossing is monotonically decreasing, the drawings D'^* and D^+ have the same rotation system. Consequently, we have transformed D into D^* by a sequence of monotone homotopy moves.

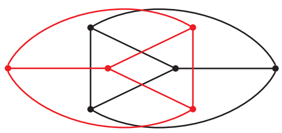
Number of moves: Note that the number of moves to clean an empty lens is upper bounded by the number of remaining crossings. Untangling the lens by a 2-0 move reduces the number of crossings by 2. Recall that a minimal loop is already clean and hence, it takes one move to reduce the number of crossings by one. Consequently, the number of homotopy moves is upper bounded by $k + \frac{1}{4}k^2$. ◀

4.5.4 Untangling via Edge Moves

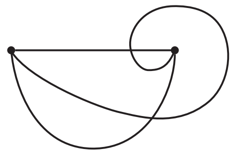
In this section we consider a more general move called an *edge move* that removes one edge and then redraws it. Note that the vertices remain fixed (as for homotopy moves). We examine the power of monotone edge moves to untangle some special graphs. Firstly, we observe that

▶ **Theorem 6.** *Not every drawing of a planar graph can be untangled with monotone edge moves.*

Proof. We show that the statement holds for two interlaced K_4's as depicted in Fig. 9a. Note that every edge is involved in exactly one crossing and redrawing it in a non-equivalent way introduces at least two crossings. ◀



(a) A drawing of two interlaced K_4's that cannot be untangled with monotone edge moves, nor, consequently, with monotone homotopy moves.



(b) The rotation system is that of a planar drawing, but it must change during homotopy moves.

Figure 9

A *cactus* graph is a graph such that every 2-connected component is an edge or a cycle. A *banana* graph is a graph with two vertices joined by an edge that has multiplicity at least 2. A *banana cactus* is a cactus graph in which each edge may be replaced by a banana. We prove:

▶ **Lemma 7.** *The following can be untangled using monotone edge moves:*
- *Any drawing of a cactus graph.*
- *Any drawing of a banana graph.*
- *Any drawing of a banana cactus in which the rotation system of the drawing (the cyclic order of edges incident to each vertex) belongs to a planar drawing.*

References

1. Javier Cano, Csaba D. Tóth, and Jorge Urrutia. Upper bound constructions for untangling planar geometric graphs. *SIAM Journal on Discrete Mathematics*, 28(4):1935–1943, 2014.
2. Hsien-Chih Chang and Jeff Erickson. Untangling planar curves. *Discrete & Computational Geometry*, 58(4):889–920, 2017.
3. Hsien-Chih Chang, Jeff Erickson, David Letscher, Arnaud De Mesmay, Saul Schleimer, Eric Sedgwick, Dylan Thurston, and Stephan Tillmann. Tightening curves on surfaces via local moves. In *Proc. 29th ACM-SIAM Symposium on Discrete Algorithms*, pages 121–135. SIAM, 2018.

4 Edward B. Curtis and James A. Morrow. *Inverse Problems for Electrical Networks*, volume 13 of *Series on Applied Mathematics*. World Scientific, 2000. p. 129 ff.
5 Branko Grünbaum. *Convex Polytopes*. Springer, 1967.
6 Joel Hass and Peter Scott. Intersections of curves on surfaces. *Israel Journal of Mathematics*, 51(1):90–120, 1985.
7 Joel Hass and Peter Scott. Shortening curves on surfaces. *Topology*, 33(1):25–43, 1994.
8 Jan Kynčl. Simple realizability of complete abstract topological graphs simplified. In *International Symposium on Graph Drawing*, pages 309–320. Springer, 2015.
9 János Pach and Gábor Tardos. Untangling a polygon. In *International Symposium on Graph Drawing*, pages 154–161. Springer, 2001.
10 Ernst Steinitz. Polyeder und Raumeinteilungen. In *Encyklopädie der Mathematischen Wissenschaften mit Einschluss ihrer Anwendungen III.AB*, volume 12, pages 1–139. 1916.
11 Ernst Steinitz and Hans Rademacher. *Vorlesungen über die Theorie der Polyeder: unter Einschluss der Elemente der Topologie*, volume 41 of *Grundlehren der Mathematischen Wissenschaften*. Springer, 1934.

4.6 Gap Planarity

Stefan Felsner (TU Berlin, DE), Ignaz Rutter (Universität Passau, DE), and Csaba D. Tóth (California State University – Northridge, US)

Recently, Bae et al. [2] defined k-gap-planar graphs, for $k \in \mathbb{N}$, that admit drawings in which each edge is "responsible" for up to k crossings. By Hall's matching theorem, it is equivalent to the following.

► **Definition 1.** A graph $G = (V, E)$ is *k-gap-planar* if it has a drawing in the plane so that for every subgraph $G = (V', E')$ there are at most $k|E'|$ crossings between the edges in E'.

Ossona de Mendez, Oum, and Wood [3] introduced a similar definition.

► **Definition 2.** A graph $G = (V, E)$ is *k-close-to-planar* if every subgraph $G' = (V', E')$ has a drawing in the plane with at most $k|E'|$ crossings (i.e., $\mathrm{cr}(G') \le k|E'|$).

It is clear that every k-gap-planar graph is k-close-to-planar. Is the converse true?

Answer: No. We show that graph $K_{6,6}$ is a counterexample. Bachmaier, Rutter, and Stumpf [1] show that $K_{6,6}$ is not 1-gap-planar. We claim that $K_{6,6}$ is 1-close-to-planar.

First, $\mathrm{cr}(K_{6,6}) = 36 = |E(K_{6,6})|$; see Fig. 10 for a crossing-minimal drawing of $K_{6,6}$. In the drawing in Fig. 10 the set of edges with precisely 4 crossings contains both adjacent and independent pairs of edges. By symmetry, we have $\mathrm{cr}(K_{6,6} - e) \le 32$ for any edge e, and $\mathrm{cr}(K_{6,6} - \{e, f\}) \le 28$ for any pair of distinct edges e, f. In particular, the crossing number of $K_{6,6} - e$ and $K_{6,6} - \{e, f\}$, resp., is clearly less than the number of edges in these graphs. It follows that any subgraph $G' = (V', E')$ obtained by removing 3 or more edges from $K_{6,6}$ satisfies $\mathrm{cr}(G') \le 28$. Hence, if $\mathrm{cr}(G') > |E'|$ it follows that $|E'| \le 28$, and hence $|E \setminus E'| \ge 8$.

An easy counting argument shows that $E \setminus E'$ contains a set A of three edges that are incident to a common vertex or a set B of three edges, two of which are adjacent and the third is independent from the other two. However, Figure 11 shows that both $K_{6,6} - A$ and $K_{6,6} - B$ are 1-gap planar, and hence also 1-close-to-planar. Both drawings are based on a 1-gap-planar drawing of $K_{5,6}$ by Bae et al. [2].

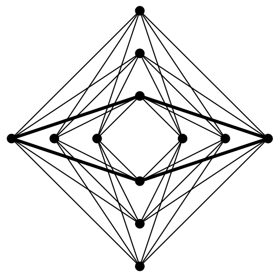


Figure 10 Crossing-minimal drawing of $K_{6,6}$. The bold edges are pairwise noncrossing and they each have four crossings, which shows that removing any two of these edges (and by symmetry any two edges) decreases the crossing number by at least 8.

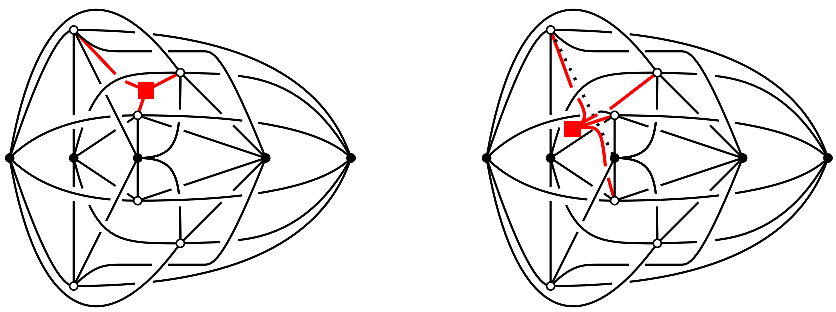


Figure 11 1-gap-planar drawings of $K_{6,6} - A$ and $K_{6,6} - B$, respectively. In the left drawing, A contains three edges incident to the square vertex. In the right drawing, B contains two edges incident to the square vertex and the dashed edge.

Open Problem. The negative answer raises the following question. Is there a function $f : \mathbb{N} \to \mathbb{N}$ such that every k-close-to-planar graph is $f(k)$-gap-planar?

References

1. Christian Bachmaier, Ignaz Rutter, and Peter Stumpf. 1-Gap planrity of complete bipartite graphs (Poster). *Proc. 26th Symposium on Graph Drawing and Network Visualization*, LNCS 11282, Springer, Cham (2018)
2. Sang Won Bae, Jean-Francois Baffier, Jinhee Chun, Peter Eades, Kord Eickmeyer, Luca Grilli, Seok-Hee Hong, Matias Korman, Fabrizio Montecchiani, Ignaz Rutter, and Csaba D. Tóth. Gap-planar graphs. *Theor. Comput. Sci.* 745:36–52 (2018)
3. Patrice Ossona de Mendez, Sang-Il Oum, and David R. Wood. Defective colouring of graphs excluding a subgraph or minor. *Combinatorica* 1–34, to appear (online in 2018).

Participants

- Eyal Ackerman
University of Haifa, IL
- Carlos Alegría Galicia
National Autonomous University of Mexico, MX
- Patrizio Angelini
Universität Tübingen, DE
- Michael Bekos
Universität Tübingen, DE
- María del Pilar Cano Vila
UPC – Barcelona, ES
- Giordano Da Lozzo
Roma Tre University, IT
- Vida Dujmović
University of Ottawa, CA
- Peter Eades
The University of Sydney, AU
- Stefan Felsner
TU Berlin, DE
- Henry Förster
Universität Tübingen, DE
- Fabrizio Frati
Roma Tre University, IT
- Radoslav Fulek
IST Austria – Klosterneuburg, AT
- Martin Gronemann
Universität Köln, DE
- Michael Hoffmann
ETH Zürich, CH
- Seok-Hee Hong
The University of Sydney, AU
- Michael Kaufmann
Universität Tübingen, DE
- Balázs Keszegh
Alfréd Rényi Institute of Mathematics – Budapest, HU
- Linda Kleist
TU Braunschweig, DE
- Stephen G. Kobourov
University of Arizona – Tucson, US
- Giuseppe Liotta
University of Perugia, IT
- Anna Lubiw
University of Waterloo, CA
- Tamara Mchedlidze
KIT – Karlsruher Institut für Technologie, DE
- Fabrizio Montecchiani
University of Perugia, IT
- Martin Nöllenburg
TU Wien, AT
- Yoshio Okamoto
The University of Electro-Communications – Tokyo, JP
- János Pach
EPFL – Lausanne, CH
- Maurizio Patrignani
Roma Tre University, IT
- Sergey Pupyrev
Facebook – Menlo Park, US
- Chrysanthi Raftopoulou
National Technical University of Athens, GR
- Günter Rote
FU Berlin, DE
- Ignaz Rutter
Universität Passau, DE
- Leonie Ryvkin
Ruhr-Universität Bochum, DE
- Csaba D. Tóth
California State University – Northridge, US
- Géza Tóth
Alfréd Rényi Institute of Mathematics – Budapest, HU
- Torsten Ueckerdt
KIT – Karlsruher Institut für Technologie, DE
- Alexander Wolff
Universität Würzburg, DE