DAGSTUHL
REPORTS

Volume 7, Issue 1, January 2017

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
June, 2017

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.7.1.i

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

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

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

Editorial Office
Marc Herbstritt *(Managing Editor)*
Jutka Gasiorowski *(Editorial Assistance)*
Dagmar Glaser *(Editorial Assistance)*
Thomas Schillo *(Technical Assistance)*

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

Report from Dagstuhl Seminar 17021

Functoriality in Geometric Data

Edited by

Mirela Ben-Chen[1], Frédéric Chazal[2], Leonidas J. Guibas[3], and Maks Ovsjanikov[4]

1 **Technion – Haifa, IL,** mirela@cs.technion.ac.il
2 **INRIA Saclay – Île-de-France, FR,** frederic.chazal@inria.fr
3 **Stanford, USA,** guibas@cs.stanford.edu
4 **Ecole Polytechnique – Palaiseau, FR,** maks@lix.polytechnique.fr

Abstract

This report provides an overview of the talks at the Dagstuhl Seminar 17021 "Functoriality in Geometric Data". The seminar brought together researchers interested in the fundamental questions of *similarity* and *correspondence* across geometric data sets, which include collections of GPS traces, images, 3D shapes and other types of geometric data. A recent trend, emerging independently in multiple theoretical and applied communities, is to understand *networks* of geometric data sets through their relations and interconnections, a point of view that can be broadly described as exploiting the *functoriality* of data, which has a long tradition associated with it in mathematics. Functoriality, in its broadest form, is the notion that in dealing with any kind of mathematical object, it is at least as important to understand the transformations or symmetries possessed by the object or the family of objects to which it belongs, as it is to study the object itself. This general idea has led to deep insights into the structure of various geometric spaces as well as to state-of-the-art methods in various application domains. The talks spanned a wide array of subjects under the common theme of functoriality, including: the analysis of geometric collections, optimal transport for geometric datasets, deep learning applications and many more.

Seminar January 8–13, 2017 – http://www.dagstuhl.de/17021
1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling, I.4 Image Processing and Computer Vision
Keywords and phrases computational geometry, data analysis, geometry processing
Digital Object Identifier 10.4230/DagRep.7.1.1

1 Summary

Mirela Ben-Chen
Frédéric Chazal
Leonidas J. Guibas
Maks Ovsjanikov

Across science, engineering, medicine and business we face a deluge of data coming from sensors, from simulations, or from the activities of myriads of individuals on the Internet. The data often has a geometric character, as is the case with 1D GPS traces, 2D images, 3D scans, and so on. Furthermore, the data sets we collect are frequently highly correlated, reflecting information about the same or similar entities in the world, or echoing semantically

Functoriality in Geometric Data, *Dagstuhl Reports*, Vol. 7, Issue 01, pp. 1–18
Editors: Mirela Ben-Chen, Frédéric Chazal, Leonidas J. Guibas, and Maks Ovsjanikov
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

important repetitions/symmetries or hierarchical structures common to both man-made and natural objects.

A recent trend, emerging independently in multiple theoretical and applied communities is to understand geometric data sets through their relations and interconnections, a point of view that can be broadly described as exploiting the *functoriality* of data, which has a long tradition associated with it in mathematics. Functoriality, in its broadest form, is the notion that in dealing with any kind of mathematical object, it is at least as important to understand the transformations or symmetries possessed by the object or the family of objects to which it belongs, as it is to study the object itself. This general idea been successfully applied in a large variety of fields, both theoretical and practical, often leading to deep insights into the structure of various objects as well as to elegant and efficient methods in various application domains, including computational geometry, computer vision and computer graphics.

This seminar brought together researchers and practitioners interested in notions of *similarity*, *correspondence* and, more generally, *relations* across geometric data sets. Mathematical and computational tools for the construction, analysis, and exploitation of such relational networks were the central focus of this seminar.

2 Table of Contents

Summary

Overview of Talks

3 Overview of Talks

3.1 Output sensitive algorithms for approximate incidences and their applications

Dror Aiger (Google Israel – Tel Aviv, IL), Haim Kaplan (Tel Aviv University, IL), and Micha Sharir (Tel Aviv University, IL)

An ϵ-approximate incidence between a point and some geometric object (line, circle, plane, sphere) occurs when the point and the object lie at distance at most ϵ from each other. Given a set of points and a set of objects, computing the approximate incidences between them is a major step in many database and web-based applications in computer vision and graphics. Two important such applications are robust model fitting, where we want to find models (lines, planes, etc.) that lie near many "interesting" points of an image (or a 3D point cloud), and point pattern matching, where we are given two sets of points A and B, and we want to find a large subset A of A for which there exists a rigid transformation which places each point of A ϵ-close to some point of B. Typically, approximate incidences are used to find candidate transformations (between model and data or between A and B), which are then tested against the whole input to filter those that have a large match. In a typical approximate incidence problem of this sort, we are given a set P of m points in two or three dimensions, a set S of n objects (lines, circles, planes, spheres), and an error parameter $\epsilon > 0$, and our goal is to report all pairs $(p, s) \in P \times S$ that lie at distance at most ϵ from one another. We present efficient output-sensitive approximation algorithms for several cases, including points and lines or circles in the plane, and points and planes, spheres, lines, or circles in three dimensions. Some of these cases arise in the applications mentioned above. Our algorithms report all pairs at distance $\leq \epsilon$, but may also report additional pairs, all of which are guaranteed to be at distance at most $\alpha\epsilon$, for some constant $\alpha > 1$. Our algorithms are based on simple primal and dual grid decompositions and are easy to implement. We analyze our algorithms and prove guaranteed upper bounds on their running time and on the "distortion" parameter α. Furthermore, we present experimental results on real and random data that demonstrate the advantage of our methods compared to other methods and to the naive implementation, commonly used in practice in these applications.

3.2 Joint denoising and distortion correction of atomic scale scanning transmission electron microscopy images

Benjamin Berkels (RWTH Aachen, DE) and Benedikt Wirth (Universität Münster, DE)

Main reference B. Berkels, B. Wirth, "Joint denoising and distortion correction of atomic scale scanning transmission electron microscopy images", arXiv:1612.08170v1 [cs.CV], 2016.
URL https://arxiv.org/abs/1612.08170v1

Nowadays, modern electron microscopes deliver images at atomic scale. The precise atomic structure encodes information about material properties. Thus, an important ingredient in the image analysis is to locate the centers of the atoms shown in micrographs as precisely as possible. Here, we consider scanning transmission electron microscopy (STEM), which acquires data in a rastering pattern, pixel by pixel. Due to this rastering combined with the magnification to atomic scale, movements of the specimen even at the nanometer scale lead to random image distortions that make precise atom localization difficult. Given a series of STEM images, we derive a Bayesian method that jointly estimates the distortion in each image and reconstructs the underlying atomic grid of the material by fitting the atom bumps with suitable bump functions. The resulting highly non-convex minimization problems are solved numerically with a trust region approach. Well-posedness of the reconstruction method and the model behavior for faster and faster rastering are investigated using variational techniques. The performance of the method is finally evaluated on both synthetic and real experimental data.

3.3 Consistent discretization and minimization of the L1 norm on manifolds

Alex M. Bronstein (Technion – Haifa, IL)

The L1 norm has been tremendously popular in signal and image processing in the past two decades due to its sparsity-promoting properties. More recently, its generalization to non-Euclidean domains has been found useful in shape analysis applications. For example, in conjunction with the minimization of the Dirichlet energy, it was shown to produce a compactly supported quasi-harmonic orthonormal basis, dubbed as compressed manifold modes. The continuous L1 norm on the manifold is often replaced by the vector l1 norm applied to sampled functions. We show that such an approach is incorrect in the sense that it does not consistently discretize the continuous norm and warn against its sensitivity to the specific sampling. We propose two alternative discretizations resulting in an iteratively-reweighed l2 norm. We demonstrate the proposed strategy on the compressed modes problem, which reduces to a sequence of simple eigendecomposition problems not requiring non-convex optimization on Stiefel manifolds and producing more stable and accurate results.

3.4 Geometric deep learning

Michael M. Bronstein (University of Lugano, CH)

Many scientific fields study data that have an underlying structure that is non-Euclidean space. Some examples include social networks in computational social sciences, sensor networks in communications, functional networks in brain imaging, regulatory networks in genetics, and meshed surfaces in computer graphics. In many applications, such geometric data are large and complex (in the case of social networks, on the scale of billions), and are natural targets for machine learning techniques. In particular, we would like to use deep neural networks, which have recently proven to be powerful tools for a broad range of problems from computer vision, natural language processing, and audio analysis. However, these tools have been most successful on data with an underlying Euclidean or grid-like structure, and in cases where the invariances of these structures are built into networks used to model them. I will discuss “Geometric deep learning”, a class of emerging techniques attempting to generalize deep neural models to non-Euclidean domains such as graphs and manifolds. I will show applications from the domains of vision, graphics, and network analysis.

3.5 A convex representation for the Elastica functional.

Antonin Chambolle (Ecole Polytechnique – Palaiseau, FR)

Joint work of Antonin Chambolle, Thomas Pock

In this talk, we have described a joint (ongoing) work with Thomas Pock (TU Graz, Austria) where we show how to design convex relaxation to curvature-dependent functionals such as $\Gamma \mapsto \int_\Gamma f(\kappa)ds$ where κ is the curvature and f is a convex function, with $f(t) \geq \sqrt{1+t^2}$. Our relaxation involves a lifting of the curve into the roto-translation group, and is easily extended as an energy of scalar functions in the plane. We can show that it is tight when Γ is the boundary of a C^2 set, and that it is a variant of a recent relaxation proposed by Bredies, Pock, Wirth [1]. It is however strictly below the standard lower semi-continuous relaxations of the Elastica which have been studied for many years (for boundaries, in the L^1 topology, see for instance [2] and the references therein).

References

1 Kristian Bredies, Thomas Pock und Benedikt Wirth. *A convex, lower semi-continuous approximation of Euler's Elastica energy.* SIAM Journal on Mathematical Analysis, 47(1):566–613, 2015.

2 F. Dayrens, S. Masnou et M. Novaga. *Existence, regularity and structure of confined elasticae.* ESAIM:COCV, to appear, 2017.

3.6 Regularized Optimal Transport

Marco Cuturi (CREST – Malakoff, FR)

The optimal transport problem, first studied by Monge and later by Kantorovich, has drawn the attention of both pure and applied mathematicians for several years now. An important tool that arises from optimal transport theory is the definition of a versatile geometry that can be used to compare probability measures, the Wasserstein geometry. Because probability measures are widely used to model social and natural phenomena, the toolbox of optimal transport has been increasingly adopted in a wide array of applied fields, such as economy, fluid dynamics, quantum chemistry, computer vision or graphics. The main motivation behind my work was to design new machine learning methodologies built upon the optimal transport geometry. The main obstacle to this goal was computational, since optimal transport is notoriously costly to compute. To avoid that issue, I have proposed 3 years ago a very efficient numerical scheme to solve optimal transport problems that can scale up to large scales and use recent progresses in hardware, namely GPGPUs. This breakthrough has inspired several works in the span of 3 years, in machine learning and beyond, which I tried to survey in this Dagstuhl seminar talk.

3.7 Ollivier Ricci curvature on network data and applications

Jie Gao (Stony Brook University, US)

In the talk I will introduce our recent work on Ollivier Ricci curvature for graphs and its applications in network analysis and graph mining. Ollivier Ricci curvature extends the notion of Ricci curvature from continuous setting to the case of graphs. For each edge xy, the curvature is defined by comparing the edge length and the optimal transport distance from x's neighbors to y's neighbors. We show that the Ollivier Ricci curvature exhibits interesting properties in real world graphs as those found for the Internet. Backbone edges, with negative curvature, connect communities of nodes which are connected by edges of positive curvature.

3.8 Model reduction for shape processing

Klaus Hildebrandt (TU Delft, NL)

Joint work of Klaus Hildebrandt, Christopher Brandt, Christoph von Tycowicz, Christian Schulz
Main reference C. von Tycowicz, C. Schulz, H.-P. Seidel, K. Hildebrandt, "Real-Time Nonlinear Shape Interpolation", ACM Transactions on Graphics, 34(3):34:1–34:10, ACM, 2015.
URL http://dx.doi.org/10.1145/2729972
Main reference C. Brandt, C. von Tycowicz, K. Hildebrandt, "Geometric Flows of Curves in Shape Space for Processing Motion of Deformable Objects", in Proc. of the 37th Annual Conf. of the European Association for Computer Graphics, 35(2):295–305, Eurographics Association, 2016.
URL http://dx.doi.org/10.1111/cgf.12832

The optimization of deformable, flexible or non-rigid shapes is essential for many tasks in geometric modeling and processing. In this talk, I will introduce model reduction techniques that can be used to construct fast approximation algorithms for shape optimization problems. The goal is to obtain run times that are independent of the resolution of the discrete shapes to be optimized. As examples, we will discuss methods for real-time elasticity-based shape interpolation and the efficient integration of a geometric flow of curves in shape space.

3.9 Similarity and correspondence problems for 3D shapes in motion

Franck Hétroy-Wheeler (INRIA – Grenoble, FR)

Joint work of Jinlong Yang, Aurela Shehu, Jean-Sébastien Franco, Franck Hétroy-Wheeler, Stefanie Wuhrer
Main reference J. Yang, J.-S. Franco, F. Hétroy-Wheeler, S. Wuhrer, "Estimation of Human Body Shape in Motion with Wide Clothing", in Computer Vision – ECCV 2016, LNCS, Vol. 9908, pp. 439–454, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-319-46493-0_27

In this talk I focus on moving 3D shapes, represented as sequences of meshes or point clouds without explicit temporal coherence. I review two recent works involving human shapes in wide clothing. The first work [2] estimates the body shape under the clothing with the help of a shape space. The second one [1] tracks the surface of the cloth over time, using the assumption that the deformation is locally near-isometric. Both methods involve different functoriality tools, namely statistical shape spaces and deformation models.

References

1 Aurela Shehu, Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, Stefanie Wuhrer. *Computing temporal alignments of human motion sequences in wide clothing using geodesic patches*. 4th International Conference on 3D Vision (3DV), Stanford, California, USA, 2016.

2 Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, Stefanie Wuhrer. *Estimation of Human Body Shape in Motion with Wide Clothing*. 14th European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 2016.

3.10 Part Structures in Large Collections of 3D Models

Vladimir G. Kim (Adobe Systems Inc. – Seattle, US)

Joint work of Vladimir G. Kim, Minhyuk Sung, Li Yi, Leonidas Guibas, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Roland Angst
Main reference L. Yi, V. G. Kim, D. Ceylan, I-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas , "A Scalable Active Framework for Region Annotation in 3D Shape Collections", ACM Transactions on Graphics – Proceedings of ACM SIGGRAPH Asia 2016, 35(6):210:1–210:12, ACM, 2016.
URL http://dx.doi.org/10.1145/2980179.2980238
Main reference M. Sung, V. G. Kim, R. Angst, L. Guibas , "Data-Driven Structural Priors for Shape Completion", ACM Transactions on Graphics (TOG) – Proceedings of ACM SIGGRAPH Asia 2015, 34(6):175:1–175:11, ACM, 2015.
URL http://dx.doi.org/10.1145/2816795.2818094

As large repositories of 3D shape collections grow, understanding the geometric data, especially encoding the inter-model similarity, their variations, semantics and functionality, is of central importance. My research addresses the challenge of deriving probabilistic models that capture common structure in large, unorganized, and diverse collections of 3D polygonal shapes. We present a part-based model for for encoding structural variations in collections of man-made shapes, and demonstrate its applications to shape completion, exploration, organization, analysis, and synthesis of geometric data. Our part based model is trained on collections of segmented 3D models with labeled parts. To facilitate creation of these data with rich annotations, we also propose an active tool for acquiring detailed region labels from crowd workers. Our tool models human behavior and explicitly minimizes human effort.

3.11 Invariants and learning for geometry analysis

Ron Kimmel (Technion – Haifa, IL)

Joint work of Ron Kimmel, Gautam Pai

In my presentation I discussed the relation between axiomatic understanding of geometric forms and structures and modern learning by convolutional networks. We focused on planar curves for which we demonstrated how Cartan's theorem indicates the sufficient invariants that could serve as a signature indicating the number of features required for matching curves. We then suggested a Siamese architecture for training such a network and demonstrated feature extortion with that network. Relation to curvature were demonstrated. Dealing with the Euclidean, affine, and similarity groups was presented. As for surfaces, we reviewed recent geometries related to equi-affine, similarity, and full affine groups of transformations with some recent matching results known for these transformations, for which Gromov distances, treating surfaces as metric spaces, were approximated.

3.12 Fully Spectral Partial Shape Matching

Or Litany (Tel Aviv University, IL), Alex M. Bronstein (Technion – Haifa, IL), Michael M. Bronstein (University of Lugano, CH), and Emanuele Rodolà (University of Lugano, CH)

Joint work of Or Litany, Emanuele Rodolà, Alex M. Bronstein, and Michael M. Bronstein
Main reference O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, "Fully Spectral Partial Shape Matching", Computer Graphics Forum (CGF) – Proc. EUROGRAPHICS 2017, Vol. 36(2), 2017; preprint available from author's webpage.
URL http://dx.doi.org/10.1111/cgf.13123
URL http://vision.in.tum.de/_media/spezial/bib/litany-eg17.pdf

We propose an efficient procedure for calculating partial dense intrinsic correspondence between deformable shapes performed entirely in the spectral domain. Our technique relies on the recently introduced partial functional maps formalism and on the joint approximate diagonalization (JAD) of the Laplace-Beltrami operators previously introduced for matching non-isometric shapes. We show that a variant of the JAD problem with an appropriately modified coupling term (surprisingly) allows to construct quasi-harmonic bases localized on the latent corresponding parts. This circumvents the need to explicitly compute the unknown parts by means of the cumbersome alternating minimization used in the previous approaches, and allows performing all the calculations in the spectral domain with constant complexity independent of the number of shape vertices. We provide an extensive evaluation of the proposed technique on standard non-rigid correspondence benchmarks and show state-of-the-art performance in various settings, including partiality and the presence of topological noise.

3.13 Efficient Deformable 2D-to-3D Shape Matching

Zorah Lähner (TU München, DE), Michael M. Bronstein (University of Lugano, CH), Daniel Cremers (TU München, DE), Emanuele Rodolà (University of Lugano, CH), and Frank R. Schmidt (TU München, DE)

Main reference Z. Lähner, E. Rodolà, F. R. Schmidt, M. M. Bronstein, D. Cremers, "Efficient Globally Optimal 2D-to-3D Deformable Shape Matching", in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2185–2193, IEEE, 2016; preprint available from author's webpage.
URL http://dx.doi.org/10.1109/CVPR.2016.240
URL http://vision.in.tum.de/~laehner/Elastic2D3D/

In this talk we present an algorithm for non-rigid 2D-to-3D shape matching, where the input is a 2D query shape as well as a 3D target shape and the output is a continuous matching curve represented as a closed contour on the 3D shape. We cast the problem as finding the shortest circular path on the product 3-manifold of the two shapes. Quantitative and qualitative evaluation confirms that the method provides excellent results for sketch-based deformable 3D shape retrieval.

3.14 Multiscale Methods for Dictionary Learning, Regression and Optimal Transport for data near low-dimensional sets

Mauro Maggioni (Johns Hopkins University – Baltimore, US)

Joint work of Mauro Maggioni, Wenjing Liao, Stefano Vigogna, Sam Gerber
Main reference W. Liao, M. Maggioni, S. Vigogna, "Learning adaptive multiscale approximations to data and functions near low-dimensional sets", in Proc. of the 2016 IEEE Information Theory Workshop (ITW 2016), pp. 226–230, IEEE, 2016.
URL http://dx.doi.org/10.1109/ITW.2016.7606829
Main reference S. Gerber, M. Maggioni, "Multiscale Strategies for Discrete Optimal Transport'', to appear in the J. of Machine Learning research (JMLR).

We discuss a family of ideas, algorithms, and results for analyzing various new and classical problems in the analysis of high-dimensional data sets. These methods we discuss perform well when data is (nearly) intrinsically low-dimensional. They rely on the idea of performing suitable multiscale geometric decompositions of the data, and exploiting such decompositions to perform a variety of tasks in signal processing and statistical learning. In particular, we discuss the problem of dictionary learning, where one is interested in constructing, given a training set of signals, a set of vectors (dictionary) such that the signals admit a sparse representation in terms of the dictionary vectors. We then discuss the problem of regressing a function on a low-dimensional unknown manifold. For both problems we introduce a multiscale estimator, fast algorithms for constructing it, and give finite sample guarantees for its performance, and discuss its optimality. Finally, we discuss an application of these multiscale decompositions to the fast calculation of optimal transportation plans, introduce a multiscale version of optimal transportation distances, and discuss preliminary applications.

3.15 On equilibrium shapes, Michell structures and "smoothness" of polyhedral surfaces

Martin Kilian (TU Wien, AT), Davide Pellis (TU Wien, AT), Johannes Wallner (TU Graz, AT), and Helmut Pottmann (TU Wien, AT)

Pure geometric shape modeling is not sufficient to achieve an efficient digital workflow from design to production, since it tends to result in costly feedback loops between design, engineering and fabrication. To overcome this problem, recent research incorporates key aspects of function and fabrication into an intelligent shape modeling process. In the present talk, we will illustrate this trend at hand of our ongoing research on shape modeling in the presence of structural and fabrication constraints. We discuss (i) 2D trusses with minimum weight (Michell structures), (ii) form-finding for architectural freeform shells and (iii) kink-minimizing polyhedral surfaces: All these themes are related to the minimization of total absolute curvature (integral of sum of absolute values of principal curvatures) of surfaces in various geometries and thus can be computationally treated with the same methodology.

3.16 Smooth Interpolation of Key Frames in a Riemannian Shell Space

Martin Rumpf (Universität Bonn, DE)

Joint work of Pascal Huber, Ricardo Perl, Martin Rumpf

Splines and subdivision curves are flexible tools in the design and manipulation of curves in Euclidean space. In this paper we study generalizations of interpolating splines and subdivision schemes to the Riemannian manifold of shell surfaces in which the associated metric measures both bending and membrane distortion. The shells under consideration are assumed to be represented by Loop subdivision surfaces. This enables the animation of shells via the smooth interpolation of a given set of key frame control meshes. Using a variational time discretization of geodesics efficient numerical implementations can be derived. These are based on a discrete geodesic interpolation, discrete geometric logarithm, discrete exponential map, and discrete parallel transport. With these building blocks at hand discrete Riemannian cardinal splines and three different types of discrete, interpolatory subdivision schemes are defined. Numerical results for two different subdivision shell models underline the potential of this approach in key frame animation.

3.17 Reduction and reconstruction of complex spatio-temporal data

Konstantin Mischaikow (Rutgers University – Piscataway, US)

It is almost cliche at this point to note that high dimensional data is being collected from experiments or generated through numerical simulation at an unprecedented rate and that this rate will continue rising extremely rapidly for the foreseeable future. Our interest is in data associated with high dimensional nonlinear complex spatiotemporal dynamics. The focus of this talk is on our efforts to use persistent homology both as a dimension reduction technique and a technique for reconstructing structures of the underlying dynamical system. I will present some results associated with dynamics of fluid convection and dense granular media and will try to highlight open questions.

3.18 Optimal transport between a simplex soup and a point cloud

Quentin Mérigot, Jocelyn Meyron, and Boris Thibert

In this talk, we are interested in solving an optimal transport problem between a measure supported on a simplex soup and a measure supported on a finite point set for the quadratic cost in $\mathbb{R}^d$. Similarly as in (Aurenhammer, Hoffman, Aronov, Algorithmica, 1998), this optimal transport problem can be recast as finding a Power diagram supported on the finite point set such that the Power cells intersected with the simplex soup have a prescribed measure.

We show the convergence with linear speed of a damped Newton algorithm to solve this non-linear problem. The convergence relies on a genericity condition on the point cloud with respect to the simplex soup and on a connectedness condition for the support of the measure defined on the simplex soup. Finally, we apply our algorithm in $\mathbb{R}^3$ to compute optimal transport plans between a measure supported on a triangulation and a discrete measure.

3.19 Partial Functional Correspondence

Emanuele Rodolà (University of Lugano, CH), Alex M. Bronstein (Technion – Haifa, IL), Michael M. Bronstein (University of Lugano, CH), Luca Cosmo (University of Venice, IT), Daniel Cremers (TU München, DE), Or Litany (Tel Aviv University, IL), Jonathan Masci (IDSIA, CH), and Andrea Torsello (University of Venice, IT)

Main reference E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, D. Cremers, "Partial Functional Correspondence", arXiv:1506.05274v2 [cs.CV], 2016.
URL https://arxiv.org/abs/1506.05274v2

In this talk we present our recent line of work on (deformable) partial shape correspondence in the spectral domain. We first introduce Partial Functional Maps (PFM), showing how to robustly formulate the shape correspondence problem under missing geometry with the language of functional maps. We use perturbation analysis to show how removal of shape parts changes the Laplace-Beltrami eigenfunctions, and exploit it as a prior on the spectral representation of the correspondence. We show further extensions to deal with the presence of clutter (deformable object-in-clutter) and multiple pieces (non-rigid puzzles). The resulting algorithms yield state-of-the-art results on challenging correspondence benchmarks in the presence of partiality and topological noise.

3.20 A Selection of Categorical Viewpoints on Shape Matching

Frank R. Schmidt (TU München, DE), Michael M. Bronstein (University of Lugano, CH), Daniel Cremers (TU München, DE), Zorah Lähner (TU München, DE), Emanuele Rodolà (University of Lugano, CH), Ulrich Schlickewei (TU München, DE), and Thomas Windheuser (TU München, DE)

Main reference Z. Lähner, E. Rodolà, F. R. Schmidt, M. M. Bronstein, D. Cremers, "Efficient Globally Optimal 2D-to-3D Deformable Shape Matching", in Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2185–2193, IEEE, 2016; preprint available from author's webpage.
URL http://dx.doi.org/10.1109/CVPR.2016.240
URL http://frank-r-schmidt.de/Publications/2016/LRSBC16a
Main reference F. R. Schmidt, T. Windheuser, U. Schlickewei, D. Cremers, "Dense Elastic 3D Shape Matching", in Efficient Algorithms for Global Optimization Methods in Computer Vision – International Dagstuhl Seminar, LNCS, Vol. 8293, pp. 1–18, Springer, 2014, preprint available from author's webpage.
URL http://dx.doi.org/10.1007/978-3-642-54774-4_1
URL http://frank-r-schmidt.de/Publications/2014/SWSC14/SWSC14.pdf
Main reference T. Windheuser, U. Schlickewei, F. R. Schmidt, D. Cremers, "Geometrically Consistent Elastic Matching of 3D Shapes: A Linear Programming Solution", in Proc. of the 2011 IEEE International Conference on Computer Vision (ICCV 2011), pp. 2134–2141, IEEE, 2011; preprint available from author's webpage.
URL http://dx.doi.org/10.1109/ICCV.2011.6126489
URL http://frank-r-schmidt.de/Publications/2011/WSSC11

While "shape" is an important concept in order to compare, detect and classify geometric data, the definition of a shape depends very much on the specific application.

Whether we understand a shape as a mere set, a topological or a metric space drives the development of different shape matching approaches.

In this presentation I talk about a selection of different categorical viewpoints on shape and shape matching in order to explore the strengths and weaknesses of these approaches from a theoretical and a practical point of view.

3.21 Towards a Geometric Functionality Descriptor

Ariel Shamir (The Interdisciplinary Center – Herzliya, IL)

The American architect Louis Sullivan coined the phrase "form follows function" in architecture. However, almost in all areas of design as well as in nature, one of the key aspects that define shapes is their functionality. In this talk I will describe several efforts to study the functionality of 3D objects by examining their shape and the connection between them. First, by trying to learn about functionality from interactions with other objects in scenes, then by trying to co-analyze similar shapes and their interactions and lastly, by studying movements and motion of shape parts.

3.22 Regularized Optimal Transport on Graphs: Rank-1 Hessian Updates for Quadratic Regularization

Justin Solomon (MIT – Cambridge, US) and Montacer Essid (New York University, US)

Optimal transportation provides a means of lifting distances between points on a geometric domain to distances between signals over the domain, expressed as probability distributions. On a graph, transportation problems such as computation of the Wasserstein distance between distributions can be used to express challenging tasks involving matching supply to demand with minimal shipment expense; in discrete language, these problems become "multi-commodity network flow" problems. Regularization typically is needed to ensure uniqueness for the linear ground distance case and to improve optimization convergence; state-of-the-art techniques employ entropic regularization on the transportation matrix.

In this work, we explore a quadratic alternative to entropic regularization for transport with linear ground distance over a graph. The dual of this problem exhibits elegant second-order structure that we leverage to derive an easily-implemented Newton-type algorithm with fast convergence. The end result is state-of-the-art performance compared with much more involved large-scale convex optimization machinery.

3.23 Common Manifold Learning with Alternating Diffusion

Ronen Talmon (Technion – Haifa, IL)

Joint work of Roy Lederman, Ronen Talmon, Hau-tieng Wu
Main reference R. R. Lederman, R. Talmon, "Learning the geometry of common latent variables using alternating-diffusion", in Applied and Computational Harmonic Analysis, Elsevier, 2015.
URL http://dx.doi.org/10.1016/j.acha.2015.09.002

We consider the problem of hidden common manifold extraction from multiple data sets, which have observation-specific distortions and artifacts. A new manifold learning method is presented based on alternating products of diffusion operators and local kernels. We provide theoretical analysis showing that our method is able to build a variant of the Laplacian of the hidden common manifold, while suppressing the observation-specific artifacts. The generality of this method is demonstrated in data analysis applications, where different types of devices are used to measure the same activity. In particular, we present applications to problems in biomedicine, neuroscience, and audio analysis.

References

1 R. Talmon, H.-T. Wu. *Latent common manifold learning with alternating diffusion: analysis and applications*. arXiv preprint arXiv:1602.00078, 2016.

3.24 Optimal transport between a point cloud and a simplex soup

Boris Thibert (Laboratoire Jean Kuntzmann – Grenoble, FR)

I will consider in this talk the computation of an optimal transport map between a measure supported on a simplex soup and a measure supported on a finite point set for the quadratic cost in Rd. Similarly as in work of Aurenhammer, Hoffman and Aronov, this optimal transport problem can be recast as finding a Power diagram supported on the finite point set such that the Power cells intersected with the simplex soup have a prescribed measure. I will show the convergence with linear speed of a damped Newton algorithm to solve this nonlinear problem and also apply our algorithm in 3D to compute optimal transport plans between a measure supported on a triangulation and a discrete measure. This work is in collaboration with Jocelyn Meyron and Quentin Mérigot.

3.25 Product Manifold Filter – Towards a Continuity Prior for 3D shape correspondence

Matthias Vestner (TU München, DE)

Many algorithms for the computation of correspondences between deformable shapes rely on some variant of nearest neighbor matching in a descriptor space. Such are, for example, various point-wise correspondence recovery algorithms used as a post-processing stage in the functional correspondence framework. Such frequently used techniques implicitly make restrictive assumptions (e.g., near-isometry) on the considered shapes and in practice suffer from lack of accuracy and result in poor surjectivity. We propose an alternative recovery technique capable of guaranteeing a bijective correspondence and producing significantly higher accuracy and smoothness. Unlike other methods our approach does not depend on the assumption that the analyzed shapes are isometric. We derive the proposed method from the statistical framework of kernel density estimation and demonstrate its performance on several challenging deformable 3D shape matching datasets.

3.26 Physical Graphic Design

Wilmot Li (Adobe Systems Inc. – Seattle, US)

Joint work of Wilmot Li, Danny Kaufman

Advances in digital fabrication technology enable an increasingly diverse spectrum of users to create physical artifacts. One emerging fabrication workflow involves the use of traditional graphic design (i.e., the design of layered 2.5D vector graphics) as a front end to produce physical objects. We refer to this process as physical graphic design. While creating physical artifacts via 2D design may seem like a solved problem (after all, there has been decades of research and commercial development on graphic design tools), the problem is more involved than it initially appears. In this talk, we discuss some of the fundamental challenges of physical graphic design, which stem from the complex interplay between design requirements and physical/fabrication constraints.

3.27 Data fitting tools in Riemannian spaces

Benedikt Wirth (Universität Münster, DE)

Joint work of Pierre-Antoine Absil, Pierre-Yves Gousenbourger, Behrend Heeren, Martin Rumpf, Peter Schröder, Paul Striewski, Max Wardetzky
Main reference P.-A. Absil, P.-Y. Gousenbourger, P. Striewski, B. Wirth, "Differentiable piecewise-Bezier surfaces on Riemannian manifolds", SIAM Journal on Imaging Sciences, 9(4):1788–1828, SIAM, 2016.
URL http://dx.doi.org/10.1137/16M1057978

Several computer vision, life science or imaging applications deal with data that can be viewed as points in a (possibly high-dimensional) Riemannian manifold. Many of these applications require data fitting of different types, for instance to perform statistical regression, interpolation, or data compression. Classical approaches to data fitting include computation of averages, of inter- and extrapolating curves, or fitting of low-dimensional manifolds. The talk presents some numerical techniques that can be employed to this end, in particular a nonlinear variational discretisation and different ways to define and compute curves or submanifolds in Riemannian spaces. The Riemannian space of (discrete) viscous shells, which can be used to model 3D geometries, serves as an example setting. Here, each point represents a (triangulated) surface embedded in 3D space. The Riemannian metric on this space takes in-plane stretching and bending of surfaces into account,which ultimately leads to elliptic PDEs in many of the data fitting problems.

3.28 Qualitative and Multi-Attribute Learning from Diverse Data Collections

Hao Zhang (Simon Fraser University – Burnaby, CA)

When dealing with a large collection of data possessing much diversity, it can be difficult, if not impossible, to properly quantify all pairwise similarities between the data entities using a single numerical distance to allow a meaningful global analysis. In this talk, I will first go over a qualitative analysis method we developed for organizing a heterogeneous collection of shapes. The method turns a set of *quartet* query results into a single categorization tree, where each quartet query gathers a similarity ranking between data pairs from a data quadruplet and these rankings are best reflected by the number of “edge hops” in the categorization tree. Next, I will introduce a new problem which tries to account for the multiple perspectives people may draw upon when performing similarity rankings. We are interested in how much one can learn of the different perspectives the human subjects may have used to provide the ranking results, where the only available information to us is the ranking results. Our first attempt is to classify the similarity queries, with reasonable confidence, based on latent perspectives. That is, we never explicitly identify the perspectives or data attributes in the output (they remain latent) nor do we pre-select a superset of candidate attributes to start the analysis.

Participants

- Dror Aiger
Google Israel – Tel-Aviv, IL
- Annamaria Amenta
Univ. of California – Davis, US
- Mirela Ben-Chen
Technion – Haifa, IL
- Benjamin Berkels
RWTH Aachen, DE
- Alex M. Bronstein
Technion – Haifa, IL
- Michael M. Bronstein
University of Lugano, CH
- Antonin Chambolle
Ecole Polytechnique –
Palaiseau, FR
- Frédéric Chazal
INRIA Saclay –
Île-de-France, FR
- Marco Cuturi
CREST – Malakoff, FR
- Jie Gao
Stony Brook University, US
- Franck Hétroy-Wheeler
INRIA – Grenoble, FR
- Klaus Hildebrandt
TU Delft, NL
- Vladimir G. Kim
Adobe Systems Inc. – Seattle, US
- Ron Kimmel
Technion – Haifa, IL
- Zorah Lähner
TU München, DE
- Wilmot Li
Adobe Systems Inc. – Seattle, US
- Or Litany
Tel Aviv University, IL
- Mauro Maggioni
Johns Hopkins University –
Baltimore, US
- Konstantin Mischaikow
Rutgers University –
Piscataway, US
- Maks Ovsjanikov
Ecole Polytechnique –
Palaiseau, FR
- Helmut Pottmann
TU Wien, AT
- Emanuele Rodolà
University of Lugano, CH
- Martin Rumpf
Universität Bonn, DE
- Frank R. Schmidt
TU München, DE
- Ariel Shamir
The Interdisciplinary Center –
Herzliya, IL
- Primoz Skraba
Jozef Stefan Institute –
Ljubljana, SI
- William Smith
University of York, GB
- Justin Solomon
MIT – Cambridge, US
- Ronen Talmon
Technion – Haifa, IL
- Boris Thibert
Laboratoire Jean Kuntzmann –
Grenoble, FR
- Matthias Vestner
TU München, DE
- Michael Wand
Universität Mainz, DE
- Benedikt Wirth
Universität Münster, DE
- Hao Zhang
Simon Fraser University –
Burnaby, CA

Report from Dagstuhl Seminar 17022

Automated Program Repair

Edited by

Sunghun Kim[1], Claire Le Goues[2], Michael Pradel[3], and Abhik Roychoudhury[4]

1 **HKUST – Kowloon, HK,** hunkim@cse.ust.hk
2 **Carnegie Mellon University – Pittsburgh, US,** clegoues@cs.cmu.edu
3 **TU Darmstadt, DE,** michael@binaervarianz.de
4 **National University of Singapore, SG,** abhik@comp.nus.edu.sg

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17022 "Automated Program Repair". The seminar participants presented and discussed their research through formal and informal presentations. In particular, the seminar covered work related to search-based program repair, semantic program repair, and repair of non-functional properties. As a result of the seminar, several participants plan to launch various follow-up activities, such as a program repair competition, which would help to further establish and guide this young field of research.

Seminar January 8–13, 2017 – http://www.dagstuhl.de/17022
1998 ACM Subject Classification D.2 Software Engineering, D.2.5 [Software Engineering] Testing and Debugging
Keywords and phrases Program repair, program analysis, software engineering
Digital Object Identifier 10.4230/DagRep.7.1.19

1 Executive Summary

Sunghun Kim
Claire Le Goues
Michael Pradel
Abhik Roychoudhury

Software engineering targets the creation of software for myriad platforms, deployed over the internet, the cloud, mobile devices and conventional desktops. Software now controls cyber-physical systems, industrial control systems, and "Internet of Things" devices, and is directly responsible for humanity's economic well-being and safety in numerous contexts. It is therefore especially important that engineers are able to easily write error-free software, and to quickly find and correct errors that do appear. Future generation programming environments must not only employ sophisticated strategies for localizing software errors, but also strategies for automatically patching them.

Recent years have seen an explosive growth in research on automated program repair, with proposed techniques ranging from pure stochastic search to pure semantic analysis. The Dagstuhl Seminar in January 2017 studies the problem of automated repair in a holistic fashion. This will involve a review of foundational techniques supporting program repair, perspectives on current challenges and future techniques, and emerging applications. The aim

Automated Program Repair, *Dagstuhl Reports*, Vol. 7, Issue 1, pp. 19–31
Editors: Sunghun Kim, Claire Le Goues, Michael Pradel, and Abhik Roychoudhury
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

is to broadly discuss and revisit underlying assumptions and methods towards the integration of automated patch synthesis into futuristic programming environments.

Conceptually, applications of program repair step far beyond the general goal of ensuring software quality, and the subject is relevant to a broad range of research areas. It is of obvious importance in software testing and analysis, because repair goes hand in hand with traditional testing and debugging activities. It is relevant to researchers in programming languages and systems, e.g., to study language and system-level techniques that integrate patch suggestions during development. The topic is relevant to researchers in systems security, as repair approaches may be customizable to patching vulnerabilities in both application and systems software. Researchers in formal methods may provide insight for provably correct repair, given appropriate correctness specifications. Finally, the topic is connected to human computer interaction in software engineering, since the repair process, if not fully automated, may involve eliciting appropriate developer feedback and acting on it accordingly.

At a technical level, one of the key discussion topics has been the correctness specifications driving the repair process. Most previous work in this domain has relied on test suites as partial correctness specifications. While often available, test suites are typically inadequate for fully assessing patch correctness. Alternative quality specifications, such as "minimality", could be explored. In addition, heavier-weight specifications, such as assertions, may provide stronger functional guarantees, leaving open the research challenge both in how to use them and how they may be derived to guide a repair process. Given the appropriate correctness specification, the task of repair usually involves three steps: localizing an error to a small set of potentially-faulty locations, deriving values/constraints for the computation desired at the faulty locations, and constructing "fix" expressions/statements that satisfy these values/constraints. Each of the three steps can be accomplished by a variety of methods, including heuristic search, symbolic analysis and/or constraint solving techniques. This allows for an interesting interplay for an entire design space of repair techniques involving ingenuous combinations of search-based techniques and semantic analysis being employed at the different steps of the repair process.

The Dagstuhl Seminar has attracted researchers and practitioners from all over the world, comprising participants active in the fields of software engineering, programming languages, machine learning, formal methods, and security. As a result of the seminar, several participants plan to launch various follow-up activities, such as a program repair competition, which would help to further establish and guide this young field of research, and a journal article that summarizes the state of the art in automated program repair.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Quality and applicability of automated repair

Yuriy Brun (University of Massachusetts – Amherst, US)

Program repair offers great promise for reducing manual effort involved in software engineering, but only if it can produce high-quality patches for defects that are important and hard for humans to fix manually. This talk presents an objective measure of repair quality, identifies shortcomings in existing automated repair techniques in terms of the quality of the patches they produce, and tackles the problem of identifying if program repair techniques can repair important and hard defects.

3.2 Automatic Tradeoffs: Accuracy and Energy

Jonathan Dorn (University of Virginia – Charlottesville, US)

Tradeoffs between competing objectives are an important part of software development and usability. However, the development effort to implement different tradeoffs is time consuming and potentially error-prone. In this work, we balance competing functional and non-functional properties via automatic program transformations. We apply multi-objective search to simultaneously optimize energy consumption and output accuracy of assembly programs. We find that relaxing the accuracy requirements enables greater energy reductions within the constraint of human-acceptability. We also find that our search-based approach identifies better tradeoff opportunities than less general techniques like loop perforation.

3.3 Deep Learning for Program Repair

Aditya Kanade (Indian Institute of Science – Bangalore, IN)

The problem of automatically fixing programming errors is a very active research topic in software engineering. This is a challenging problem as fixing even a single error may require analysis of the entire program. In practice, a number of errors arise due to programmer's inexperience with the programming language or lack of attention to detail. We call these common programming errors. These are analogous to grammatical errors in natural languages.

Compilers detect such errors, but their error messages are usually inaccurate. In this work, we present an end-to-end solution, called DeepFix, that can fix multiple such errors in a program without relying on any external tool to locate or fix them. At the heart of DeepFix is a multi-layered sequence-to-sequence neural network with attention which is trained to predict erroneous program locations along with the required correct statements. On a set of 6971 erroneous C programs written by students for 93 programming tasks, DeepFix could fix 1881 (27%) programs completely and 1338 (19%) programs partially.

3.4 Prex: Finding Guidance for Forward and Backward Porting of Linux Device Drivers

Julia Lawall (INRIA – Paris, FR)

A device driver forms the glue between a device and an operating system kernel. When the operating system kernel changes, the device driver has to change as well. Our goal is to automate this kind of forward porting, and analogously backwards porting, focusing on drivers for the Linux kernel. We propose a three step approach, based on 1) compilation of the driver with the target kernel to identify incompatibilities, 2) collection of examples of how to fix these incompatibilities from the commits stored in the revision control system, and 3) generalization of the identified examples to produce change rules appropriate for porting the driver to the target kernel. In this talk, we present the tools gcc-reduce that have been designed to carry out the first two steps. The third step remains future work. Our approach effectively exploits the specific nature of the driver porting problem: the device is fixed and so the required changes are in the interface with the kernel, limiting the kinds of changes required, and many drivers supporting devices with a similar functionality interact with the kernel in a similar way, implying that porting examples are available.

3.5 Automated Inference of Code Transforms and Search Spaces for Patch Generation

Fan Long (MIT – Cambridge, US)

We present a new system, Genesis, that processes sets of human patches to automatically infer code transforms and search spaces for automatic patch generation. We present results that characterize the effectiveness of the Genesis inference algorithms and the resulting complete Genesis patch generation system working with real-world patches and errors collected from top 1000 github Java software development projects. To the best of our knowledge, Genesis is the first system to automatically infer patch generation transforms or candidate patch search spaces from successful patches.

3.6 Learning-based Program Repair

Fan Long (MIT – Cambridge, US)

Code learning and transfer techniques have been recently adopted by many successful automatic patch generation systems to improve patch generation results. This talk presents an overview of these two kinds of techniques. A code learning technique learns useful human knowledge from a training set of past human patches. The technique then applies the learned knowledge either to prioritize correct patches ahead in a patch generation search space or to infer productive mutation transforms to form the search space. A code transfer technique extracts useful program logic from existing code of a donor program or examples of Q&A websites. It then converts the extracted logic to a patch for a recipient program.

3.7 ASTOR: A Program Repair Library for Java

Matías Sebastían Martínez (University of Valenciennes, FR)

During the last years, the software engineering research community has proposed approaches for automatically repairing software bugs. Unfortunately, many software artifacts born from this research are not available for repairing Java programs. To reimplement those approaches from scratch is costly. To facilitate experimental replications and comparative evaluations, we present Astor, a publicly available program repair library that includes the implementation of three notable repair approaches (jGenProg2, jKali and jMutRepair). We envision that the research community will use Astor for setting up comparative evaluations and explore the design space of automatic repair for Java. Astor offers researchers ways to implement new repair approaches or to modify existing ones. Astor repairs in total 33 real bugs from four large open source projects.

3.8 Combining syntactic and semantic repair

Sergey Mechtaev (National University of Singapore, SG)

Test-driven automated program repair approaches traverse huge search spaces to generate fixes. Several search space prioritization heuristics have been proposed to increase the probability of finding correct repairs. Although existing systems could generate correct fixes for large real-world projects, they suffer from limitations of the search space exploration algorithms. First, current techniques may omit high quality patches during exploration, which results in generation of suboptimal low quality repairs. Second, current techniques are able to explore only relatively small search spaces and, therefore, fix only a small number of defects. We propose a synergy of syntax-based and semantics-based patch generation methods that explicitly generates a search space and semantically partitions it during test execution. The proposed algorithm is able to efficiently traverse the search spaces in an arbitrary order. As a result, our technique is the first that guarantees to the most reliable patch (global maximum) in the search space according to a given static prioritization strategy and yet scales to large search spaces. Evaluation on large real-world subjects revealed that the proposed algorithm generates more repairs, more repairs equivalent to human patches, and find repairs faster compared to previous techniques. Apart from that, the algorithm and the design of the search space enable our system to traverse the search space faster than existing systems with explicit search space representation.

3.9 Antifragile Software and Correctness Attraction

Martin Monperrus (University of Lille & INRIA, FR)

Can the execution of a software be perturbed without breaking the correctness of the output? We present a novel protocol to answer this rarely investigated question. In an experimental study, we observe that many perturbations do not break the correctness in ten subject programs. We call this phenomenon "correctness attraction". The uniqueness of this protocol is that it considers a systematic exploration of the perturbation space as well as perfect oracles to determine the correctness of the output. To this extent, our findings on the stability of software under execution perturbations have a level of validity that has never been reported before in the scarce related work. A qualitative manual analysis enables us to set up the first taxonomy ever of the reasons behind correctness attraction.

3.10 Detecting and repairing performance bugs that have non-intrusive fixes

Adrian Nistor (Florida State University – Tallahassee, US)

Performance bugs are programming errors that slow down program execution. Unfortunately, many performance bugs cannot be automatically detected or repaired by existing techniques. In this talk I will present Caramel, a novel static analysis technique that detects and then automatically repairs performance bugs that have non-intrusive fixes likely to be adopted by developers. 116 of the bugs found and repaired by Caramel in 15 popular applications (e.g., Chrome, Mozilla, Tomcat, Lucene, Groovy, GCC, MySQL, etc) have already been fixed by developers based on our bug reports.

3.11 Automated Software Transplantation

Justyna Petke (University College London, GB)

Genetic Improvement is the application of evolutionary and search-based optimisation methods to the improvement of existing software. For example, it may be used to automate the process of bug-fixing or execution time optimisation. In this talk I present another application of genetic improvement, namely automated software transplantation. While we do not claim automated transplantation is now a solved problem, our results are encouraging: we report that in 12 of 15 experiments, involving 5 donors and 3 hosts (all popular real-world systems), we successfully autotransplanted new functionality from the donor program to the host and passed all regression tests. Autotransplantation is also already useful: in 26 hours computation time we successfully autotransplanted the H.264 video encoding functionality from the x264 system to the VLC media player; compare this to upgrading x264 within VLC, a task that we estimate, from VLC's version history, took human programmers an average of 20 days of elapsed, as opposed to dedicated, time.

3.12 Understanding and Automatically Preventing Injection Attacks on Node.js

Michael Pradel (TU Darmstadt, DE)

The Node.js ecosystem has lead to the creation of many modern applications, such as server-side web applications and desktop applications. Unlike client-side JavaScript code, Node.js applications can interact freely with the operating system without the benefits of a security sandbox. The complex interplay between Node.js modules leads to subtle injection vulnerabilities being introduced across module boundaries. This talk presents a large-scale study across 235,850 Node.js modules to explore such vulnerabilities. We show that injection vulnerabilities are prevalent in practice, both due to eval , which was previously studied for browser code, and due to the powerful exec API introduced in Node.js . Our study shows that thousands of modules may be vulnerable to command injection attacks and that even for popular projects it takes long time to fix the problem. Motivated by these findings, we present Synode , an automatic mitigation technique that combines static analysis and runtime enforcement of security policies for allowing vulnerable modules to be used in a safe way. The key idea is to statically compute a template of values passed to APIs that are prone to injections, and to synthesize a grammar-based runtime policy from these templates. Our mechanism does not require the modification of the Node.js platform, is fast (sub-millisecond runtime overhead), and protects against attacks of vulnerable modules while inducing very few false positives (less than 10%).

3.13 Anti-patterns in Search-Based Program Repair

Mukul Prasad (Fujitsu Labs of America Inc. – Sunnyvale, US)

Search-based program repair automatically searches for a program fix within a given repair space. This may be accomplished by retrofitting a generic search algorithm for program repair as evidenced by the GenProg tool, or by building a customized search algorithm for program repair as in SPR. Unfortunately, automated program repair approaches may produce patches that may be rejected by programmers, because of which past works have suggested using human-written patches to produce templates to guide program repair. In this work, we take the position that we will not provide templates to guide the repair search because that may unduly restrict the repair space and attempt to overfit the repairs into one of the provided templates. Instead, we suggest the use of a set of anti-patterns — a set of generic forbidden transformations that can be enforced on top of any search-based repair tool. We show that by enforcing our anti-patterns, we obtain repairs that localize the correct lines or functions, involve less deletion of program functionality, and are mostly obtained more efficiently. Since our set of anti-patterns are generic, we have integrated them into existing search based repair tools, including GenProg and SPR, thereby allowing us to obtain higher quality program patches with minimal effort.

3.14 Semantic Techniques for Program Repair

Abhik Roychoudhury (National University of Singapore, SG)

In this talk, I will first recapitulate briefly the challenges in automated program repair. I will then move to discuss semantic analysis techniques for program repair, and position them with respect generate and validate approaches to program repair. I will conclude the talk by discussing how semantic approaches can be combined with generate and validate approaches.

3.15 Automated techniques for fixing performance issues in JavaScript applications

Marija Selakovic (TU Darmstadt, DE)

Many programs suffer from performance problems, but unfortunately, finding and fixing such problems is a cumbersome and time-consuming process. My work focuses on JavaScript, for which little is known about performance issues and how developers address them. To address these questions, I present the empirical study of 98 reproduced performance-related issues from 16 popular JavaScript projects. The findings illustrate that developers optimize their code with relatively simple code changes and that the most common root cause of JavaScript performance issues is the inefficient usage of native and third-party APIs. To help developers find and fix performance problems related to API usages, I present an approach for finding conditionally equivalent APIs and detecting the usages of an API that can be replaced by an equivalent and more efficient alternative for a given input. The approach is based on two-phases dynamic analysis of API usages in web applications. In the first phase, the analysis detects potentially equivalent methods based on their type signatures and name equivalence. In the second phase, the analysis executes these methods with all observed inputs, approximates their execution times and derives an equivalence condition. Finally, the analysis points to all code locations that can be optimized by using more efficient API and suggests a refactoring to the developers.

3.16 I Get by With a Little Help From My Friends: Crowdsourcing Program Repair

Kathryn T. Stolee (North Carolina State University – Raleigh, US)

Regular expressions are commonly used in source code, yet developers find them hard to read, hard to write, and hard to compose. Motivated by the prevalence of regular expression usage in practice and the number of bug reports related to regular expressions, I propose several future directions for studying regular expressions, including error classification, test coverage, test input generation, reuse, and automated program repair. The repair strategies can work in the presence or absence of fault localization, and with or without test cases. I conclude by discussing the potential impact of integrating regex support into automated program repair approaches.

3.17 Towards Trustworthy Program Repair

Yingfei Xiong (Peking University, CN)

Many different approaches have been proposed for automatic program repair, but the precision and recall of current techniques are not satisfactory. This talk will introduce our work exploring the possibilities of improving precision and recall. First, we did a study of manual program repair, and the results show that human developer indeed can achieve high precision and recall under the same setting of automatic program repair. Second, we proposed two techniques, mining QA site and statistical condition synthesis, mainly for improving precision. Both techniques achieve a precision of around 80%.

3.18 How Developers Diagnose and Repair Software Bugs (and what we can do about it)

Andreas Zeller (Universität des Saarlandes, DE)

How do practitioners debug computer programs? In a retrospective study with 180 respondents and an observational study with 12 practitioners, we collect and discuss data on how developers spend their time on diagnosis and fixing bugs, with key findings on tools and strategies used, as well as highlighting the need for automated assistance. To facilitate and guide future research, we provide DBGBENCH, a highly usable debugging benchmark providing fault locations, patches and explanations for common bugs as provided by the practitioners.

3.19 Automated Test Reuse via Code Transplantation

Tianyi Zhang (UCLA, US)

Code clones are common in software. When applying similar edits to clones, developers often find it difficult to examine the runtime behavior of clones. The problem is exacerbated when some clones are tested, while their counterparts are not. To reuse tests for similar but not identical clones, Grafter transplants one clone to its counterpart by (1) identifying variations in identifier names, types, and method call targets, (2) resolving compilation errors caused by such variations through code transformation, and (3) inserting stub code to transfer input data and intermediate output values for examination. To help developers cross-check behavioral consistency between clones, Grafter supports fine-grained differential testing at both the test outcome level and the intermediate program state level.

In our evaluation on three open source projects, Grafter successfully reuses tests in 94% of clone pairs without inducing build errors, demonstrating its automated test transplantation capability. To examine the robustness of Grafter, we automatically insert faults using a mutation testing tool, Major, and check for behavioral consistency using Grafter. Compared with a static cloning bug finder, Grafter detects 31% more mutants using the test-level comparison and almost 2X more using the state-level comparison.This result indicates that GRAFTER should effectively complement static cloning bug finders.

Participants

- Yuriy Brun
University of Massachusetts – Amherst, US
- Celso G. Camilo-Junior
Federal University of Goiás, BR
- Jonathan Dorn
University of Virginia – Charlottesville, US
- Lars Grunske
HU Berlin, DE
- Ciera Jaspan
Google Inc. – Mountain View, US
- Aditya Kanade
Indian Institute of Science – Bangalore, IN
- Sarfraz Khurshid
University of Texas – Austin, US
- Dongsun Kim
University of Luxembourg, LU
- Sunghun Kim
HKUST – Kowloon, HK
- Julia Lawall
INRIA – Paris, FR
- Claire Le Goues
Carnegie Mellon University – Pittsburgh, US
- Fan Long
MIT – Cambridge, US
- Matías Sebastían Martínez
University of Valenciennes, FR
- Sergey Mechtaev
National University of Singapore, SG
- Martin Monperrus
University of Lille & INRIA, FR
- Adrian Nistor
Florida State University – Tallahassee, US
- Alessandro Orso
Georgia Institute of Technology – Atlanta, US
- Justyna Petke
University College London, GB
- Michael Pradel
TU Darmstadt, DE
- Mukul Prasad
Fujitsu Labs of America Inc. – Sunnyvale, US
- Abhik Roychoudhury
National University of Singapore, SG
- Marija Selakovic
TU Darmstadt, DE
- Kathryn T. Stolee
North Carolina State University – Raleigh, US
- David R. White
University College London, GB
- Yingfei Xiong
Peking University, CN
- Andreas Zeller
Universität des Saarlandes, DE
- Tianyi Zhang
UCLA, US

Report from Dagstuhl Seminar 17031

Planning and Robotics

Edited by

Malik Ghallab[1], Nick Hawes[2], Daniele Magazzeni[3], Brian C. Williams[4], and Andrea Orlandini[5]

1 **LAAS – Toulouse, FR,** malik@laas.fr
2 **University of Birmingham, GB,** n.a.hawes@cs.bham.ac.uk
3 **King's College London, GB,** daniele.magazzeni@kcl.ac.uk
4 **MIT – Cambridge, US,** williams@csail.mit.edu
5 **CNR – Rome, IT,** andrea.orlandini@istc.cnr.it

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17031 on "Planning and Robotics". The seminar was concerned with the synergy between the research areas of *Automated Planning & Scheduling* and *Robotics*. The motivation for this seminar was to bring together researchers from the two communities and people from the Industry in order to foster a broader interest in the integration of planning and deliberation approaches to sensory-motor functions in robotics. The first part of the seminar was dedicated to eight sessions composed on several topics in which attendees had the opportunity to present position statements. Then, the second part was composed by six panel sessions where attendees had the opportunity to further discuss the position statements and issues raised in previous sessions. The main outcomes were a greater common understanding of planning and robotics issues and challenges, and a greater appreciation of crossover between different perspectives, i.e., spanning from low level control to high-level cognitive approaches for autonomous robots. Different application domains were also discussed in which the deployment of planning and robotics methodologies and technologies constitute an added value.

Seminar January 15–20, 2017 – http://www.dagstuhl.de/17031
1998 ACM Subject Classification I.2 Artificial Intelligence, I.2.9 Robotics
Keywords and phrases adjustable autonomy, artificial intelligence, automated planning and scheduling, goal reasoning, human-robot interaction, plan execution, robotics, robust autonomy
Digital Object Identifier 10.4230/DagRep.7.1.32

1 Executive Summary

Andrea Orlandini
Malik Ghallab
Nick Hawes
Daniele Magazzeni
Brian C. Williams

Automated Planning and Scheduling (P&S) and Robotics were strongly connected in the early days of A.I., but became mostly disconnected later on. Indeed, Robotics is one of the most appealing and natural application area for the P&S research community, however such a natural interest seems to not be reflected by advances beyond the state-of-the-art in P&S

research in Robotics applications. In light of the accelerated progress and the growth of economic importance of advanced robotics technology, it is essential for the P&S community to respond to the challenges that these applications pose and contribute to the advance of intelligent robotics.

In this perspective, a Planning and Robotics (PlanRob) initiative within the P&S research community has been recently started with a twofold aim. On the one hand, this initiative would constitute a fresh impulse for the P&S community to develop its interests and efforts towards the Robotics research area. On the other hand, it aims at attracting representatives from the Robotics community to discuss their challenges related to planning for autonomous robots (deliberative, reactive, continuous planning and execution etc.) as well as their expectations from the P&S community. The PlanRob initiative was initiated as a workshop series (http://pst.istc.cnr.it/planrob/) started at the International Conference on Automated Planning and Scheduling (ICAPS) in 2013. The PlanRob workshop editions gathered very good feedback from both the P&S and Robotics communities. And this resulted also in the organisation of a specific Robotics Track at ICAPS since 2014.

The aim of this Dagstuhl Seminar was to reinforce such initiative and increase the synergy between these two research communities. Then, most of the attendees contributed with position statements (whose abstracts are available in this report) to present their major challenges and approaches for addressing them. In general, this involved sharing views, thoughts and contributions across the following main topics:

- **Long-term autonomy / Open world planning**, providing an overview on issues related to continuous planning for robots with partial information or even incomplete models;
- **Knowledge Representation and Reasoning in Planning**, with presentations on cognitive features and robot planning;
- **Challenges in Industrial, Logistics & Consumer Robotics**, providing relevant insights related to deployment of robots in real world scenarios;
- **Human-Robot Planning**, with a wide overview on planning solutions for dealing with interactions between humans and robots;
- **Planning and Execution**, discussing issues and challenges related to robust planning and execution for robot control;
- **Task & Motion Planning / Hybrid planners**, with presentations on integrated solutions for robot control at different levels;
- **Reliable and Safe Planning for Robotics**, providing an overview of ISO standards for robots and, more in general, investigating the exploitation of formal methods to guarantee reliability in robotic applications;
- **Technological Issues in Robot planning/Multi-robot Planning**, with statements on technological issues in (multi-) robot solutions.

Each session was animated by (i) an opponent, whose role was to be critical about the position statements and (ii) a moderator, to organise the discussion. Therefore, opponents and moderators have provided a short summary of the session ideas and discussion in dedicated Synthesis Sessions to further foster the discussion.

In addition, two panel sessions have been organised on (i) **Evaluation, Benchmarking and Competitions**, discussing the experience in RoboCup@Home and the organisation of the new Planning and Execution competition (that will be held in 2017), and (ii) **Outreach & Training**, discussing about the possible organisation of summer schools and the opening of new scientific networking initiatives (e.g., a COST action).

During the seminar, discussions focused on different issues, challenges, possible solutions and new promising trends over a very wide variety of relevant topics: knowledge representation, modelling issues, the need of incomplete models; cognitive features such as, for instance, learning and goal reasoning; human-aware solutions for flexible human-robot interaction; adaptive solutions for human-robot collaboration; robust execution capable of effectively dealing with failure; integration issues in robotic architecture that, e.g., exploit different kind of models and then perform hybrid reasoning; application of formal methods to provide verification and validation functionalities to guarantee reliable robotic systems; etc. Indeed, addressing the integration of P&S and Robotics for development of intelligent robots entails covering a heterogeneous spectrum of problems, often requiring complex solutions that require a vast set of knowledge and technologies.

During the seminar, there was a very high level of engagement and interaction between the participants, enabling a lively and productive week. The main outcome of the seminar was to share a common understanding of issues and solutions with thorough discussions. And the workshop ended with an open discussion on possible follow ups and possible actions to create further opportunities for fostering synergies and interactions between the two communities.

2 Table of Contents

Executive Summary

Overview of Talks

3 Overview of Talks

3.1 Joint Human-Robot Activity is a context and a challenge for pertinent investigation in Automated Planning

Rachid Alami (LAAS – Toulouse, FR)

Let us consider what should be the planning abilities for a robot that has to share Task and Space with a Human partner. Planning and more generally on-line deliberation is clearly a necessary ability since it allows the robot to reason on action and situation consequences, to anticipate or to act pro-actively.

This is a task oriented problem. The question can be expressed as "How to perform a task, in presence or in interaction with humans, in the best possible way i.e. taking into account safety and efficiency but also acceptability of robot behaviour by the humans and legibility of robot intentions".

The robot has to build and manage "Shared Plans" involving Humans and itself. Besides the criteri mentioned above, the models should integrate the key notion of predicting and reasoning about human mental state as well as human preferences. Based on this, the problem is not only to build a plan for a robot which collaborates with humans but to build a "sufficiently good" plan that answers satisfactorily, a various levels of abstraction, the questions: what, who, where, when, how? Our aim is to discuss the issues mentioned above and illustrate them based on preliminary results that not only give some concrete examples of human-aware task and motion planning but also how they can fit in a coherent architecture for a cognitive and interactive robot.

3.2 A Blueprint for the Evolution of Perspectives: Planning Technology as a Basis for the Mass Customization of Robots

Iman Awaad (Hochschule Bonn-Rhein-Sieg – St. Augustin, DE)

As the field of robotics expands beyond a critical mass with a view to advancing the public interest, a change in paradigm is needed that transfers the complexity of customizing some functions from those with deep technical competence to the users – a process of mass customization.

Allowing for customization of functionality will enable users to adjust the functions for which they acquired the robot to their own needs and biases, thus enabling it to serve as an extension to their own capacity, rather than of what the manufacturer might perceive to be a standard set of customers' capacities and needs. This ability to customize the functionality would inherently necessitate a transparent interaction that explains what the agent is doing and why, and enables the user to modify behavior by specifying preferences, contexts, and rules (what to do, what not to do and when). The user's own explanations may well play a role in specifying such knowledge. Given that planning technology is responsible for the decision making process that determines the behavior of the robot (plan-based robot control), it is perfectly placed to play a central role in this customization process by enabling end-users to directly customize and even create the planning domains that are used by the robots via other ubiquitous tools (such as tablets and mobile communication devices).

This is just one of many aspects that will need investigation in the process of scaling up robotic ubiquity, alongside issues such as legal/regulatory implications, embedded cultural bias and social acceptance, ethical ramifications, etc. (Looking at the two simple examples of diffusion of drones and autonomous vehicles already exemplifies many of these factors). One extra point is that now all "parameters of the robot's autonomy" are defined by the manufacturers, some of these parameters would need to be transferred to the customer/user. This would apply to aspects that relate to personal/cultural/practical/social preferences. (Even McDonalds "localizes", and offers veggie burgers in Hindu regions. In this case, we can't talk just of regional localization but individual customization, because each household has its own individuality and potential preferences for how the autonomy of a robot is manifested.) Perhaps, by keeping in mind this goal of enabling the customization (whether by learning, or by parametrization, and so on..), we may also find that we as developers have created toolsets and modalities that simplify the process of adding and changing the functionality of robots. More needs to be done within the community to speed up the development process and remove the extensive barriers to entry that currently exist. The sharing of best practices, lessons learned, solutions (which should be developed with re-use in mind), and even raw data sets would be a start in this direction. The creation of a central repository for the various application domains that makes available specifications of planning domains, tasks, actions, preferences, agendas, and context, in whatever representations they were formulated in is a worthwhile endeavor. Cooperation is important for interoperability but also for transboundary regional policy-making (e.g. Uber).

Finally, we need to be aware of (and exploit) technological innovation that is developing rapidly in parallel, and that may or may not accelerate or help shape the trajectory of autonomous Position paper for Dagstuhl Seminar 17031 Planning and Robotics (PlanRob) robots. For example, in the same way that the community has benefitted from technologies that were initially developed for the mobile communications market, we should capitalize on the technology (and standards) that have been developed for the Internet of Things and other innovations that are yet to appear. Similarly, in the same way that the development of selfdriving cars resulted from the cooperation between the community and the automotive industry, a similar cooperation with architects and home furnishing companies could go a long way in injecting the much sought-after structure in home environments and providing (both static and possibly dynamic) knowledge of these objects and environments to the planning and acting processes.

3.3 Towards Autonomous Robots via Technology Integration

Roman Bartak (Charles University – Prague, CZ)

We share the challenge of developing autonomous robots that can do anything that their hardware allows them to do. The ultimate testbed is a robot that can do any task that a human can do when remotely controlling the robot. The idea is learning how people are performing the task with a robot and then "programming" the robot to be able to solve the same (and similar) tasks hopefully in a more general setting. To fulfill this vision one needs expertise in many areas including control theory, computer vision, localization, path finding, activity planning, knowledge representation, machine learning, etc. Despite progress in all these areas separately, there are still big gaps between them that prevent efficient exploitation of research results to build advanced integrated systems such as autonomous robots.

Robotics today is very separated from Artificial Intelligence (and activity planning). We can identify several gaps there such as symbolic vs numeric reasoning and model-free vs modelbased methods. We believe that different approaches are better for different settings and hence it is more appropriate to find a way how to integrate them rather than preferring one approach over the other one to handle all the problems. Activity planning is a symbolic modelbased approach while robotics is based more on numeric model-free techniques. Many problems arise from the clash between these different worlds. How can we obtain the symbolic model necessary to apply planning techniques for a particular robotic hardware? Which modelling framework is appropriate to provide necessary expressivity and efficiency? How to formulate the planning goal based on the current state of the system? When (re-) planning should be initiated? How does the plan convert to executable instructions? How does the sense-act approach fit the plan-execute approach? Etc. In our research, we develop model-centric techniques with the focus on planning domain models that are efficient for problem solving. We use flying drones as a robotic platform because they are "kinetically" simple (opposite for example to robotic hands) – the drone can only fly and observe. Still, the drones can solve interesting practical tasks such as mapping, inspection, search, tracking, delivery etc. Our focus is on software for controlling the drone and for information processing rather than on hardware, which is a "standard platform". This is based on idea that current hardware is advanced enough to perform complex tasks but the weak part is software that controls it. Hence we believe that AI will play a more significant role in robotics in upcoming years.

The first question is what are the symbolic activities to be used in activity planning for robots. We are trying to identify activities as somehow homogenous behaviors using machine learning techniques (such as clustering) applied to sensor (and control) data obtained from a drone when being manually controlled. Currently, we do not use the camera as a sensor, but it would be very interesting to exploit computer vision techniques as a source of extra sensor inputs (very rich inputs in this case). The next step is, for such activities, finding some formal description that can be parameterized (for example flying forward for a specific distance) and finding a controller for executing such activities. This way we are trying to bridge the continuous (numeric) world of robots with the symbolic world of planning. Having the activities, the next step is finding a way of efficient planning with them. PDDL planning is based on the "flat" structure of activities with no extra control knowledge. Despite a huge progress in domain-independent planning, PDDL planners are still hardly applied to practical problems due to efficiency problems. There exist modelling frameworks such as hierarchical task networks and control rules to guide the planners, but it is cumbersome to obtain such models. We are going in the direction of recipe-based planning models where the causal structures of activities can be learnt by observing how the robot solves a specific task while being controlled by a human (activities need to be detected first from sensor data). Hierarchical structures seem desirable there to get better flexibility via having reusable tasks that decompose to simpler sub-tasks. Getting experience from linguistics, namely formal grammars that can describe hierarchical structures, might be beneficial there thanks to exiting support tools for formal grammars, for example, allowing one to do formal reasoning with the models such as verification. This is an active research topic where technology developed for one area (natural language processing) can be exploited in a very different area (activity planning). Though we are addressing automated techniques to obtain and use the activity models, we also see a big gap in authoring tools for developing control software for robotic platforms. Frameworks such as ROS simplified transfer of tools between robotic platforms, but using ROS still requires non-trivial knowledge and low-level programming skills that makes it hard to program "standard" robotic platforms for specific tasks. We believe that the above-described approach of using symbolic activities (directly executable by a robotic platform) that are connected via recipes for performing specific tasks can simplify development of robotic software. Visual programming languages and systems such as Ozobot are good motivation there. They can be used to manually describe recipes (plans) to solve specific tasks as well as to visualize automatically-learned recipes. The challenge is how to go beyond the models for classical sequential execution of activities to more flexible reactive models that are still easy to understand. The major reason for having some formalism for activity models is the need to verify such models before they can be used in industrial setting. In summary, we see symbolic models as a way to simplify development of robotic software. These models need to be tightly connected to control software that is based more on numeric techniques. Hence integration of various technologies is necessary there.

3.4 Robot Planning for the mastery of human-scale everyday manipulation tasks

Michael Beetz (Universität Bremen, DE)

Robot planning can be considered as the reasoning about the future execution of robot programs (plans) in order to optimize their performance in terms of achieving their goals and efficiency (McDermott). The holy grail of robot (action) planning ever since the Shakey project has been to equip robotic agents with human-level (manipulation) action capabilities. Unfortunately, the progress along this dimension has been modest at best. I believe that much of the lack of progress is caused by the way the research field of task planning abstracts (robot) actions (see PDDL). It makes the assumption that reasoning about abstract preconditions and effects is sufficient for planning complex manipulation tasks. If we interpret this assumption from a probabilistic point of view, we can restate it by asserting that the probability of achieving the desired effects of actions is conditionally independent of how the robot executes the actions given that the preconditions of the actions are satisfied. This means that our robot action planning systems would not change their belief about whether an action is executed successfully depending on whether the robot plans to grasp an object with one hand or two, which grasp type it applies, and so on. Or, if a fetch action is executed by two-year old or an experienced waiter. In contrast our experience with realizing human-scale manipulation activities for robotic agents shows that most of the intelligent problem-solving capabilities of robots are needed in order to decide how to execute the actions to make them succeed, that is to achieve the desired effects of an action and avoid the undesired ones.

I believe that in order to materialize the impact that robot planning technology can have for robotic agents that are to accomplish human-scale manipulation activities, we have to extend our representation and reasoning machanisms to include the concepts of motor cognition. Motor cognition is a discipline in cognitive psychology of action which is concerned with the learning, reasoning, and planning of how to parameterize and synchronize motions in order to accomplish actions. I foresee a new generation of powerful robot planning systems that do not only reason symbolically about their actions but also subsymbolically with their "eyes and hands". Today's disruptive technologies, in particular modern game technology, physics simulation, data analytics, and deep learning give us the opportunities to pursue this direction.

3.5 Plug&Play Autonomous Robots

Ronen I. Brafman (Ben Gurion University – Beer Sheva, IL)

Robotics today is reminiscent in many ways of personal computing in the early 80s. Some key industrial applications, various toy applications, and even more difficult to use than DOS. One of the keys to more powerful computing, and to more useful robotics, is the ability to easily integrate new software and new hardware without having to configure them manually. The difficulty in robotics is even greater as new capabilities can interact in complex ways not only internally, but also externally. Beyond this, we need simple ways of getting robots to do what we wish, and writing dedicated scripts each time is not a good solution. One of the key software-engineering challenges for robotics is to facilitate a world in which it is easy and safe to integrate new capabilities into a robotic platform. We believe AI, and AI planning in particular, provides some of the key ideas for addressing this challenge.

We are trying to address this need by developing formal, machine readable and actionable "robot-capability description language" – essentially, a rich action description language that replaces semi-formal software engineering formal techniques, and is closely related to efforts to standardize the specification of web-services. Essentially, we argue that action description languages should be elevated to the status of function specifications. We are aware of the existence of formal methods for programming robots that provide powerful tools for writing code with behavior guarantees. Yet, we fear that these will be confined to the small community of researchers working on them, as there is a large, and likely to be growing community of users that are continuously contributing useful robotic code, albeit one written using standard programming techniques and with standard tools. Providing a formal specification of the properties of this code seems much easier and more realistic than rewriting this code from scratch, and amenable, to some extent, to automation.

If every functional module has an associated formal description of its normal behavior, it is easy to provide added value services that

1. Monitor its performance and alerts of any abnormal situation,
2. Improve the model based on actual experience,
3. Verifies controllers that combine existing modules, and provides information about their probably effects,
4. Combines existing modules automatically,

– and probably additional added value services that would be developed in the future.

To this effect, we have developed a rich XML-based specification language that contains four classes of functions: achieve, maintain, observe, and detect, tools for generating monitoring code, and tools for generating automated interfaces between this code and the ROSPlan system.

In our work we continuously discover new desirable features, and we anticipate that this will continue to be the case, and hope to join efforts with other in providing the "right" specification language and tools that exploit it.

3.6 Planning with ROS

Michael Cashmore (King's College London, GB)

We are working towards integrating Hybrid Systems Planners with real-world systems. This involves a number of challenging and interesting questions. Given real-valued and non-linear functions within the planning process, what new kinds of models can be explored?

In each robotic system there is a decision as to what should be included in the planner's model, and what should be handled by a specialised component. This question, and the way in which external planning tools are connected together presents as many interesting problems as the modelling decisions themselves.

All too often the planner is seen by non-experts as a black box. It is expected to produce a behaviour that is already scripted. To use a general planner in a robotic system in a more interesting way, a large amount of integration takes place around it. Consider the following requirements sketch for a planning robot:

- The models used by the system are generated automatically:
  - This includes components for state estimation, state prediction, and abstraction.
  - These build a model of the current environment in the language of the planner.
  - The long-term goals of the robot, are either provided by hand or driven by the robot's motivations.
  - A model of the robot's capabilities are generated from a formal description of the hardware.
- The planning takes place at multiple levels of abstraction, starting with long-horizon strategic plans, which gradually are refined into short-horizon task plans.
- Plans generated by the planner are executed robustly with some prior preprocessing:
  - They are pre-processed into a structure that explicitly contains plan failure conditions and causal links (which are not often included in planner output).
  - They are also translated into a structure that has some formal guarantees on controllability; or are combined with execution rules. The resultant plan could better be described as a controller.
- Even using the most advanced planning techniques for uncertainty, the robotic system reliably deviates from the planner's model. Action or plan failure that occurs as a result is detected, and also repaired in a way that does not unwittingly affect other ongoing plans and processes.
- The robot interacts with humans and so:
  - A component is included for plan legibility, so that a user can understand that the robot lives between and outside of scripted and broken behaviour.
  - The user is able to modify the long term goals and behaviour of the robot.
  - The user is able to interact within the confines of planned behaviour; assisting the robot, being assisted, and communicating.

All of the above could arguably be included within the black box of the planning system. I am interested in exploring which components are essential in any "planning" robotic system, which are optional, and which can be replaced by equivalent components.

When engaging in the task of integrating components, the result is often the minimal functioning system. I am also interested in providing a generic architecture for linking those essential components that will facilitate the easy use of existing libraries in new systems, opening up the black box of planning to general (ROS) users.

3.7 Teach Once Logistics Perspective

Martin Davies (Guidance Automation Ltd – Leicester, GB)

Any repetitive task is ripe for automation. Since the invention of the mechanized weaving loom in the late 1700's, the mechanization and subsequent automation of manufacturing practices has revolved around three steps:

Firstly, the identification of a suitable process for completing a stage of the manufacturing process. Requiring skilled humans, throughput is low and cost is high.

Secondly, process optimization reduces variation and errors. The introduction of jigs, fixtures, an SOP (standard operating procedure) etc. de-skills the task and reduces product variance.

Finally, once the process is tested and rigid, the now dominant cost factor, the human element is removed. Automation is brought in, increasing profitability and securing market dominance.

The concrete example of this is an automotive production line. High numbers of identical products are manufactured for a number of years with minimal variance. The line consists of a number of cells, with each cell designed for a specific task. When the product line is refreshed, the factory is shutdown, the cells are reconfigured and robots within cells are re-taught via manual operation. They will be re-taught in less than a day and will follow that teaching for a number of years before being reconfigured. Teach once. Repeat infinite.

Robotics now is about replacing humans directly in dynamic environments. Guidance Automation automates and provides scheduling software for forklift automated guided vehicles (AGV's) in logistics environments. We are at the "third" stage in the manufacturing process, but the application of the technology is overly complex and unwieldy.

Consider now the installation of a fleet of forklift AGV's. We have to map the environment, and potentially install some form of navigational aids. We then have to align the map that the AGV's will navigate to CAD, enabling us to have AGV's that can navigate the environment in a frame of reference common to the existing warehouse management system (WHM).

Now we have to link all stock locations in the warehouse to physical locations in order to schedule the AGV's. This requires mapping and labelling all shelves, which currently may only be in a human readable format.

Then there is a large amount of scripting and teaching to be done, determination of locations to pick and drop pallets. Environmental variance is high, shelves may be at different heights, scripts cannot be cut and pasted. System operation must be guaranteed without the requirement to teach every pick location.

Finally, we also have to handle the inability to see all objects. Paper hanging down from pallets, obscuring shelves and markings. The process is rigid for deskilled human operators, but too flexible for robotic undertaking.

My question is how do we apply the teachings of the manufacturing industry to the logistics environment. Should our focus be on robot evolution or is the logistics problem too unconstrained to solve efficiently? How can we as robotics experts influence the logistics process enabling us to make it teach once.

3.8 Learning Spatial Models for Navigation

Susan L. Epstein (City University of New York, US)

Deliberative robot navigation architectures often model the world as a detailed metric map. Given a target destination, the robot constructs an optimal plan within the map and then executes it. Realistically, however, doors open and close, and people (or other robots) move rapidly about. In such dynamic worlds, a map identifies only static obstructions. Thus, plan-based navigation requires plan repair and often re-planning.

Our approach, SemaFORR, is intended for autonomous indoor navigation, where maps are unreliable or unavailable, and landmarks may be absent, obscured, or obliterated: complex office buildings, warehouses, and search-and-rescue settings. Rather than respond only to percepts and known obstructions, we have chosen to learn spatial affordances, spatial abstractions that facilitate movement and represent a robot's experience of the world. Our thesis is that spatial affordances learned from local sensing during travel can both support effective, autonomous robot navigation and provide a lingua franca for dialogue with a human traveling companion. SemaFORR's spatial affordances include unobstructed areas, useful transit points, route segments, doors, and passageways. Together they form a spatial model that represents the robot's world but is not a metric map. SemaFORR has rapidly learned spatial models that support efficient travel in a variety of simulated two-dimensional worlds. That approach was purely reactive, however, without recourse to a map or a planner.

SemaFORR can learn a spatial model either from its percepts as it navigates or in simulation on a map of its environment. Current work includes the construction of ROS-based SemaFORR modules parameterized for a variety of real-world robot platforms. Work is also underway to adapt SemaFORR for movement through crowd models based on well-documented human behaviors. We will extend work on movement toward and through crowds to real-world environments, where we can test a variety of human-robot interactions.

Thus we envision SemaFORR as a collaborator in navigation in two ways:

- As a companion to SLAM: Current development includes classical planning in a traditional metric map, novel planners in the spatial model, and techniques to integrate them. We expect that a plan derived in SemaFORR's spatial model will prove more flexible than those of traditional map-based planners.
- As a companion to a human traveller: Recent work in cognitive neuroscience (including the 2014 Nobel Prize in Physiology or Medicine) has detected place cells, grid cells, and direction mechanisms in mammalian brains that have strong analogies to our spatial affordances. Thus we believe that SemaFORR is a strong foundation for dialogue with a human traveling companion about decisions and the nature of the environment. The

user-friendly qualities of SemaFORR's spatial model and the simplicity of its reasoning structure provide a natural common ground within which to discuss which way to travel and why.

3.9 Flexible Execution of Human-Robot Collaborative Plans: a cognitive control

Alberto Finzi (University of Naples, IT)

In social and service robotics, complex collaborative plans should be executed while interacting with humans in a natural and fluent manner. Indeed, a robotic system is often provided with structured tasks to be accomplished; on the other hand, this execution should be continuously adapted to the human activities, commands, and interventions. In these scenarios, the human interaction is unpredictable and very complex (multimodal, verbal and non-verbal, either explicit or implicit, etc.), therefore, several mechanisms should be supported, such as human state/activity/intention recognition, joint attention, attention manipulation, referencing, turn-taking, action coordination, dialogue management.

Different frameworks have been proposed in the robotics literature to conciliate natural human-robot interaction and the execution of complex cooperative plans. The dominant approach relies on the planning and execution paradigm and deploys replanning to adapt task execution to the behaviors of the agents involved in the interaction. This paradigm is effective in mixed-initiative planning and execution, however, the associated continuous planning/replanning process usually impairs the naturalness and effectiveness of the interaction with the humans and the environment.

We propose to tackle these issues from a different perspective exploiting the concept of cognitive control introduced in cognitive psychology and neuroscience to describe the executive mechanisms/functions needed to support flexible, adaptive responses and complex goal-directed cognitive processes and behaviors. Inspired by this literature, we propose to deploy a supervisory attentional system paradigm [Norman Shallice 1986]. In this framework, executive attention plays a crucial rule. Indeed, the supervisory attentional system coordinates and monitors hierarchically organized behavioral schemata exploiting attentional regulations to facilitate the execution of desired processes, while inhibiting the inappropriate ones. This paradigm seems particularly relevant not only for flexible plan execution, but also for human-robot interaction, because it directly provides attentional mechanisms (attention manipulation, joint attention, action facilitation, habituation, etc.) considered as pivotal for implicit, non-verbal human-human communication [Tomasello 2008].

Following this approach, we propose and discuss an interactive framework that combines human-aware planning, flexible and interactive plan execution, human monitoring, multimodal interaction, and task teaching. In this setting, a cooperative plan is considered as an attentional guidance for an attentional executive system influenced by the human actions and the environmental changes. Finally, we discuss how the proposed framework can support not only flexible and interactive execution of structured tasks, but also incremental task adaptation through teaching by demonstration.

3.10 Combined Task and Motion Planning is Classical Planning

Hector Geffner (UPF – Barcelona, ES)

Robot planning is a broad area. I focus here on what's called "combined task and motion planning". Some approaches split the problem into two, task and motion planning, that are ad- dressed by two types of planning algorithms. Such a decomposition however tends to be ineffective as the two components are not independent. More recent approaches have aimed at exploiting the efficiency of modern classical planners, either by taking the spatial constraints into account as part of a symbolic, goal-directed replanning process [7], or by using geometrical information in the computation of the classical planner heuristic [3]. My position is that combined task and motion planning (CTMP) is classical planning, and that it may pay to address the problem in this way. What is classical planning? It's planning from a known initial state using deterministic actions with known effects for achieving a goal state. It is assumed that the state space is discrete and finite, and given in compact form as the values of a set of variables whose values are changed by the actions. The first obstacle that needs to be overcome in order to formulate and solve CTMP as classical planning is that the space of robot and object configurations is not finite or discrete. Yet, it's common for such configuration spaces to be discretized by means of probabilistic sampling schemes [5]. The second challenge is the limitation of existing classical planners for modeling CTMP problems even when discretized. It's not clear indeed how to express for example that "spatial collisions" are to be avoided in STRIPS-like languages without ending up with huge encodings. The third challenge is that, even if one develops a suitable planning language for modeling discretized CTMP as a classical planning problem, there may be no effective planners for dealing with such a language, nor efficient ways for translating it into one that can be handled by modern planners like LAMA. Yet these are all limitations of current classical planners, not of classical planning that is supposed to deal with sequential decision problems involving deterministic actions and a fully known initial state. Moreover, these limitations have little to do with robotics. For example, the Atari video-games, and many of the games of the General-Video AI game competition are classical planning problems that cannot be addressed by the standard classical planners. Indeed there is no PDDL encodings for such problems but just a simulator. In the last few years, we have developed expressive planning languages [2] and classical algorithms that can effectively plan with such languages and with simulators [6, 4]. More recently, we have shown how these ideas can be applied to CMTP where problems involving tens of objects and a PR2 robot can be fully compiled and solved as classical planning problems [1].

References

1 J. Ferrer, G. Frances, and H. Geffner. Submitted. Videos at http://bit.ly/2fnXeAd, 2016.
2 G. Frances and H. Geffner. *Modeling and computation in planning.* In Proc. ICAPS, 2015.
3 C. Garrett, T. Lozano-Perez, and L. Kaelbling. *FFRob: An efficient heuristic for task and motion planning.* In Algorithmic Foundations of Robotics XI, pages 179–195. Springer, 2015.
4 T. Geffner and H. Geffner. *Width-based planning for general video-game playing.* In Proc. AIIDE-2015, 2015.
5 S. M. LaValle. *Planning algorithms.* Cambridge, 2006.
6 N. Lipovetzky, M. Ramirez, and H. Geffner. *Classical planning with simulators: Results on atari.* In Proc. IJCAI, 2015.

7 S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. *Combined task and motion planning through an extensible planner-independent interface layer*. In Proc. ICRA, pages 639–646, 2014.

3.11 Planning for Long-Term Robot Autonomy

Nick Hawes (University of Birmingham, GB)

An open problem within the use of planning technologies on robots is the problem of planning for long--term autonomy. There are at least two challenges within this problem. The first is that within a long--term autonomous robot, planning may never formally start and end. Instead the robot should maintain a plan which achieves goals for some future time horizon, where that time horizon is only part of a longer--term schedule of goal--driven behaviour. For example the robot may have a known list of goals which it should achieve that day, or that week, and must also respond to goals provided to it in an on--demand fashion. It also must manage it's limited resources (notably battery and time) to ensure that it is able to achieve all its goals in its future, not just the ones in the time horizon. Finally it must also be able to deal with the inevitable failures and unexpected consequences of operating in the real world. This challenge brings together planning and scheduling along with prior work on oversubscription planning, continual planning and goal--driven autonomy. The second challenge within planning for long--term autonomy is being able to automatically generate planning domains, environment models etc. in such a way that they capture the experience of the robot (in plan execution, and of the environment more broadly) over the long time periods it operates for. This will allow for planning models which better match reality, resulting in better performing robots and fewer failures at execution time.

3.12 Plan-based robot control

Joachim Hertzberg (Universität Osnabrück, DE)

Robotics and AI planning are both in a healthy state, as we all know. A deep integration of the two is open in many respects. In my view, the depth of integration would increase with the number and caliber of processes of robot control, on the one side, and planning, on the other, that run in closed loop – and with closed loop, I mean that they both take input from the respective other one and generate output for the respective other one (which this one takes as input). To make such a closed-loop integration possible, requires a deep understanding of what happens on both ends of the loop, and requires deep integration in terms of representation formalisms, representation granularity, control granularity, and, of course, interfaces. I will name three issues, which are intertwined, where I see room as well as the need for improvement.

Execution monitoring

The classic. As long as a course of action runs perfectly as planned and as envisaged by the planning domain model, all is good. As soon as it deviates from this nominal course, we have little to say about it. This starts with the problem of recognizing in the first place when deviation starts. We may represent time lines about state variables to include timing behavior of actions – this allows to detect delays, but not every delay is a fatal deviation. Even if we determine that something has gone wrong with executing some action or plan, then what was the cause of the fault? AI planning is strong in modeling abstractly nominal courses of action; plan-based robot control needs deep models of the environment and its dynamics that allow non-nominal developments to be understood, too. To make it efficient, both should probably work in the same representation framework.

Semantic perception

Interpretation of the data flow from a configuration of robot sensors is, in the utmost of cases and methods, understood as a process of bottom-up aggregation and abstraction from sensor data "upward" to symbols, and eventually into pieces or sentences in a representation language. That is good, but it is just one part of the story. To have its knowledge influence the action of the robot efficiently, the inverse process is needed: the priming by context, as determined by reasoning about the knowledge about the current situation, of the act of perceiving. This starts from directing the sensors and the sensor data processing resources to salient spots or events in the environment, continues over discarding much if not most of the raw sensor data deemed uninteresting, and goes into interpreting the salient data in the current context according to the current needs. Sensor data – be they single still images or full ROS bags – don't hit us like rainfall. As part of robot control, they are, or should be, actively acquired, to a large extent.

Dealing with huge, flawed, and deficient bodies of knowledge

Knowledge in many robot domains is huge (think of all that needs to be known about an office building for a courier robot), facts are subject to change independent of robot actions (think of all that goes on in an office building outside of the robot's control), and the robot can impossibly know all that is the case in its world, even though it may once become relevant for its action (again, think of all that is the case in an office building). Yet, it has to get along with what it knows, as good as it knows it. AI has it since a long time that reasoning is defeasible. Robotics tells you: this is the norm rather than an exotic exception in a robot's knowledge base; whatever reasoning is used has to function with huge knowledge bases that contain large numbers of flaws and gaps. What is a formalism and a calculus to cope with that? What additional robot tasks are needed for making its precious knowledge base sustainable in spite of these flaws and gaps? As far as I know, no one knows.

3.13 Flexible Planning for HRI

Laura M. Hiatt (Naval Research Lab – Washington, DC, US) and Mark Roberts (Naval Research – Washington, US)

As robots become more pervasive in our daily lives, it becomes more important to be able to interact with them, and task them, naturally and spontaneously. One major hindrance to achieving this goal is a lack of flexibility in how robots can execute tasks and interact with the world. For example, if a robot that is carrying a tool to a teammate is asked by another human to help them hold open a door, the robot should conceptually be able to help. In most current robotic planning systems, however, unless such a scenario was specifically foreseen and engineered, this level of flexibility is not possible: the robot would either have to prematurely end the tool task, or deliver the tool and then return to help with the door. This both hinders overall robot performance and, we argue, decreases the quality of interactions between robots and human partners.

We are addressing this problem by beginning to investigate how different robotic tasks can be concurrently executed in an ad hoc fashion, even if they utilize overlapping resources on the robot (such as the same arm). One of the key questions of this concurrency is how to ensure the correctness of the combined execution of the tasks. In our approach, we address correctness by enabling tasks to specify constraints that other executing tasks must observe when executing concurrently. Returning to the earlier example, the robot carrying the tool to the teammate may specify that its arm can move around or be used in another task as long as the tool stays 3 inches away from any object. Both goal reasoning and planning algorithms must then be extended to support these constraints, allowing reasoning about domains not only where multiple tasks can execute at once, but also where the tasks can physically affect one another.

It is our belief that giving robots this additional level of flexibility will both increase their overall functionality, as well as increase their ability to naturally team with human partners.

3.14 Safety Reconsidered – planning for safe human-robot collaboration

Michael W. Hofbaur (Joanneum Research – Klagenfurt/Wörthersee, AT)

Real world applications of human-robot collaboration require conformity to relevant standards. For robotic manipulators, for example, it is obligatory to operate the machines according to the guidelines defined in ISO 10218 “Robots and robotic devices – Safety requirements for industrial robots” and ISO 15055 “Robots and robotic devices – Collaborative robots”. These standards introduce a framework for safety that is conceptually different from the safety concept that is typically used in computer science, AI and planning, in particular.

For example, ISO 15066 defines specific operational modes that restrict the functionality of a robot to physically safe manipulation operations. These limitations often lead to impractical robot applications (e.g. too slow, inadequate force / torque capabilities for real-world applications, etc.). Using environmental perception and task- / situation-aware high level control using advanced planning and scheduling could significantly improve the robot's capabilities and enable new applications of human-robot collaboration. However, this will also require the planning component to obey certain practical requirements, such as coding-standards and architectural-considerations and guaranteed dependability levels.

Our impulse talk will therefore consider safety concepts from both perspectives and sketches possible future directions for planning and scheduling in high-level control systems of collaborative robots that enable safe autonomous behaviors of these machines.

3.15 Conditional Planning for Human-Robot Interaction

Luca Iocchi (Sapienza University of Rome, IT)

Service robots interacting with people in home or public environments are required to execute many different tasks, including various forms of interaction with the users. These actions typically depends on the needs or requests of the users and generating all the possible combinations in advance and manually is a too demanding task. Moreover, when interacting with naive users, the robot has to be robust to many unexpected situations. Finally, the assumption that a robot is always able to have perfect knowledge about the situation is too unrealistic and plans must be robust to imperfect and noisy perception.

In this scenarios, automated planning procedures are very useful to generate many interactions with a compact representation of the domain, resulting in less effort for the designer of the system and better performance of the overall task.

However, the most common standard planning techniques present the following issues:

1. classical planning assumes perfect knowledge and perfect action execution;
2. replanning after failures assumes perfect sensing (to detect failures and to determine the new initial state for replanning);
3. Markov Decision Processes (MDP) require perfect observability of the state;
4. Partially Observable Markov Decision Processes (POMDP) allow for modelling perception uncertainty, but efficiency of the algorithms does not scale with the complexity of the problem and determining correct probability values is a difficult task.

Conditional planning instead has nice features in this scenario: (i) it is based on explicit sensing, thus perception is limited to the execution of such sensing actions (rather than passively used to determine any state); (ii) only partial knowledge about the initial state is required; (iii) conditional planners are efficient. In other words, conditional planning allows for execution of minimal sensing procedures that minimizes the risk of wrong execution of the plans due to wrong perceptions.

However, conditional planning still suffers from the following problems: (i) actions are assumed to be deterministic (except for different sensing outcomes) and perfect; (ii) plans do not contain loops, so it is not possible to model repetitions of parts of the plan (which is useful in HRI applications). We propose to solve these problems by adding an additional layer in the plan generation procedure that aims at improving the robustness of a plan generated by a conditional planner through execution rules. This additional layer takes the conditional plan generated by a planner and a set of declarative rules and generates a more complex and robust plan.

The execution rules allow to: (1) define execution variables (different from planning variables) associated to action executions; (2) check their values at run-time in order to execute local recovery procedures when the values of such execution variables affect the success of the execution of actions; (3) define conditions that allow repetitions of parts of the plan.

This idea has been implemented and experimented with different formalisms, including the transformation of an MDP policy to a conditional plan [Iocchi et al. ICAPS 2016] and the use of the conditional planner Contingent-FF integrated in ROSPlan (http://kclplanning.github.io/ROSPlan).

While the robust plan is represented using the Petri Net Plans (PNP) formalism (http://pnp.dis.uniroma1.it) and executed by the PNP engine.

The method has been tested with a real robot interacting with many users in public environments, within the COACHES project (https://coaches.greyc.fr/)

Although, the system is still sensible to wrong perceptions (i.e., it is still possible that wrong perceptions determine wrong executions of the plans), this risk is minimized, since perceptions are performed a minimum number of times, that is only when it is strictly required to proceed with the plan.

Discussion will include how to further improve the system, by adding more principled solutions of the above mentioned problems. Moreover, generation and execution of robust plans is an orthogonal feature that is useful for every robot planning application domain.

3.16 Rethinking Computational Investments in Planning and Execution

Gal A. Kaminka (Bar-Ilan University – Ramat Gan, IL)

Planning when to plan, and when not to. I work on multi-robot plan execution; in particular, on executing plans for teams of robots. I have been working on plan execution systems for teams of robots, for more than 15 years. A cornerstone of executing team plans is to recognize the points in which robots will make a decision, e.g., by voting or other social-choice mechanism. By recognizing these points, the systems I have designed are able to guarantee good teamwork of the robot, in the sense of carrying out their tasks in an agreed-upon manner.

There is really no feasible way to plan the voting process in advance, in the sense of planning out which robot will vote for what option. This is inherently an execution-time process. However, planning to hold a vote should be possible, just as it should ideally be possible to plan whether to hold a vote of a specific type (e.g., plurality vs. Borda vs. dictatorial). The lesson is that some executions you cannot plan, but you can plan to execute.

More generally …Back in 1997, I was taking a graduate course in artificial intelligence planning, taught by Craig Knoblock and Yolanda Gil. As a final project, we were asked to write a paper that would tackle an open question, provide the literature survey and recommend directions for further research. My paper addressed the computational effort in planning and execution. I contrasted the approaches of the planning community (graphplan and satplan were the newest planners), and the robotics community (subsumption was being pushed out in favor of behavior-based control). The two communities were seemingly at odds, scientifically. The mainstream planning community focused, as it does today, on building planners that extensively relied on simulating the effects of actions. To do this, they needed models of how the world behaves. The mainstream robotics community had just completed its embrace of reactivity, subsumption and behavior-based robotics, which emphasized throwing away models (and therefore planning), or at the very least limiting the role of planning significantly. Just a few years before, Matt Ginsburg wrote that "Universal Planning is an almost universally bad idea" in an issue of AI Magazine, and Agre and Chapman, having successfully built Pengo without a planner, were leaving both communities with the impression that there is a justified gap between robotic acting in dynamic environments, and planning for static environments. Sure, there were also researchers working on integrated planning and execution, but they were mostly working on softbots or learning policies via reinforcement learning.

I took the position that the planning community and the robotics community were in fact in complete agreement: they were both advocating the use of incredibly dumb executors. The planning community were expecting an executor which will blindly execute a series of actions, given as grounded operators. The most that would be expected from such an executor would be to check whether preconditions and effects hold as predicted. The robotics community, having given up hope on planning, was instead building very dumb mechanisms as well. Execution of policies, from this respect, is not very different: follow the policy and myopically respond as dictated. The various behavior selection and fusion mechanisms which are often discussed in this community are as myopic as the sequential selection of grounded operators for execution. In short, both communities were assuming essentially all computation is carried out in planning time. By comparison, decision-making during execution is computationally trivial, because it is myopic.

I would like to see us shifting the computational burden, to do more computation during execution (possibly, invested in projections of future state, but not only). Some HTN planners, and BDI agent architectures, come somewhat close to this, in the sense that they both allow on-line refinements for plan recipes. But their refinement is an all-or-nothing deal: recipes given in advance are either refined or are not; they are not usually modified during execution. The philosophical shift in focus of the autonomous agents community, from the agent as a planner, to the agent as a plan selector is not enough in this regard. It emphasizes the importance of integrating planning and execution, and it highlights the very real challenges involved in everything beyond generating plans. But plans are still thought of as rigid objects, generated by planners, to be handed off for execution after some filtering and selection.

I would like to have a planner that knows when to plan, and when to leave off planning to a later point in time; a planner that plans for later planning. I believe the way to achieve this goes through rethinking of the planning process, as a process that spans execution. No more interleaving of calls to a planner with calls to step execution. Rather, a single computational process that plans when it can, and automatically stops filling in details when it cannot.

3.17 Cognitive Robotics on the Factory Floor

Erez Karpas (Technion – Haifa, IL)

Industry 4.0 is one of the most common buzzwords heard today. The term is a reference to the industrial revolutions of the past: The (first) industrial revolution, in the 18th century, came about with the advent of the first steam- powered machines. The second industrial revolution, in the 19th and early 20th century, involved using electricity to power assembly lines in performing mass production. The third industrial revolution, from the 1970s, involved computer-integrated manufacturing (as well as computer-aided design), and is the origin of industrial robotics. While the term industry 4.0 has many different interpretations, it usually refers to integrating many sensors in the factory, and using analytics to derive some actionable insight from the data (this is often called Internet-of-Things and Big Data). In my talk, I will argue that cognitive robots could be of great benefit on the factory floor, and should be counted as an integral part of the fourth industrial revolution.

The first major benefit of cognitive robotics is in reducing the cost of setting up a factory. Traditional industrial robots typically require extensive, low-level, programming by highly specialized experts | a very expensive process in both cost and factory downtime. On the other hand, cognitive robots could be programmed by giving them a goal, such as "assemble 1000 electric razors of type X", and will plan all the low-level details by themselves. Second, customized manufacturing is a very important trend now. For ex- ample, Motorola now allows customers to customize the phones they order on the web, choosing between millions of possible configurations. Cognitive robots could manufacture each device according to order, further taking into account deadlines, shipping schedules, etc.

Finally, although ideally robots will be able to do everything humans can, this will not happen any time in the near future. Thus, human-robot teamwork is an important part of any cognitive manufacturing robot. The ability to communicate with, and work alongside, a human requires high-level reasoning, and is an interesting challenge to the planning and robotics community. Although putting cognitive robots on the factory floow is a significant challenge. However, I believe it is easier to succeed on the factory floor than, for example, in a home environment for service robots, because the environment is much more structured and controllable, and because there are far fewer ethical concerns. Furthermore, a manufacturing setting provides clear and measurable economic benefit, which can allow us to claim that it is worth investing in planning and robotics research.

3.18 Multi-Robot Planning with Spatial and Temporal Constraints

Sven Koenig (USC – Los Angeles, US)

There are several gaps between symbolic AI planning research (as typically presented at ICAPS) and robotics that need to be addressed to make AI planning research more useful for robotics:

- Robots operate in dynamic worlds, which makes on-line planning necessary. They also need to make many decisions quickly during execution, not only because stopping wastes time but also because the world continues to evolve. Thus, AI planning research needs to focus more on realtime planning.
- Robots operate in spatial environments. Thus, AI planning research needs to focus more on spatial planning and the integration of spatial and temporal planning.
- Robots operate around people. Thus, AI planning research needs to take into account that the behavior of robots needs to be predictable (for example, that similar tasks should result in similar behaviors). This criterion needs to be incorporated into the objective function of AI planners. (This issue is also important in the context of replanning for

teams of humans, such as for emergency teams, where it is often important to keep the modifications of the previous plan small in case of contingencies in order to avoid a large coordination overhead. Again, similar situations should result in similar behaviors.)

- Robots cannot execute plans perfectly since planning uses models of the world and models never represent reality perfectly. Thus, plan execution will frequently deviate from the plan, which makes plan-execution monitoring and replanning necessary. AI planning research often assumes that this issue can be handled well with online replanning, which always re-solves the planning problem from the current state in case plan execution deviates from the plan. However, planning is often too slow for online replanning to be a viable strategy for real-time planning. Thus, AI planning research needs to develop integrated planning and plan-execution architectures that use slow replanning only very selectively. They could, for example, use hierarchical replanning strategies that use fast plan-adaptation whenever possible and slow replanning only as a last resort.
- Finally, multi-robot systems are more fault-tolerant and allow for more parallelism than single-robot systems. Thus, AI planning research needs to focus more on cooperative multi-agent planning, both in centralized and – very importantly – decentralized settings. Some AI planning research studies multi-agent planning but focuses on privacy, which is less important for robotics than other applications.

The research of my research group addresses these issues in the context of multi-robot path finding, where multi-robot teams have to assign target locations among themselves and then plan collision-free paths to them. Examples include automated warehouse systems, autonomous aircraft towing vehicles, office robots and game characters in video games. For example, hundreds of robots already navigate autonomously in Amazon fulfillment centers to move inventory pods all the way from their storage locations to the packing stations. Path planning for these robots is NP-hard, yet planning must find high-quality collision-free paths for them in real-time.

There is a long way to go to bridge the gap between AI planning research and robotics. We advocate robotics-friendly planning domains for IPC competitions as one possible way to engage AI planning researchers. For example, one suggestion is to use a planning domain that models the Harvard TERMES robots as part of the multi-agent planning competition. The Harvard TERMES project investigated how multiple robots can cooperate to build userspecified three-dimensional structures much larger than themselves. Planning is required, even for single robots, to build structures effectively since they need to build ramps to reach high places but ramps consist of many blocks and are time-consuming to build. Thus, robots need to plan carefully when and where to build ramps and, once built, how to utilize them best. Planning for single robots is already difficult due to the large number of blocks and long plans. Planning for multiple robots is even more difficult since, as for multi-robot path finding, it needs to reason about how to achieve a high degree of parallelism without robots obstructing each other even though many robots operate together in tight spaces.

3.19 Explainable Robotics

Lars Kunze (University of Birmingham, GB)

Planning and decision making in the real world requires autonomous robots to draw on different sources of knowledge. This includes prior knowledge about the domain, knowledge acquired through longterm experience, and knowledge derived from more recent observations through robotic sensors. Hence, planning and decision making as well as the resulting behavior of robot systems, can be quite complex. For end-users, and sometimes even for robot developers, it might not be clear why a system behaves in a particular way. Therefore, I argue that autonomous robots should be equipped with principled ways that allow them to explain their own behavior and their own decisions. For example, to answer the question "why did you stop in front of a green traffic light?" an autonomous car could generate an explanation such as "I stopped because I saw a person approaching at very high speed". By providing explanations and/or justifications for decisions users can follow and comprehend the internal processes of robot systems. Such transparency can lead to an increased user acceptance and eventually to trust in autonomous robots in general. Moreover, while analyzing the behavior of a robot system developers could benefit from mechanisms that can explain 'why' an action was performed.

Realizing systems that can provide explanations about themselves and their own behavior poses several challenges. First of all, these systems require explicit (and interpretable) representations about the world, about themselves, and their planning and decision making processes. Secondly, robots need to be equipped with inference mechanisms to reason about different possible explanations. Finally, to provide explanations to users and developers novel interfaces for Human-Robot Interaction (HRI) are required. In previous work, we have developed the Semantic Robot Description Language (SRDL). SRDL is based on the Web Ontology Language (OWL) and provides a principled way to describe robots, their components, and their capabilities semantically. It allows robots to explain what tasks (and actions) they are able to perform and it allows them to infer why they are not able to perform certain actions. Hence, I believe that SRDL is a good starting point for enabling robots to reason about their own planning and decision making processes. For reasoning about different possible explanations we are currently investigating Answer Set Programming (ASP). ASP has been successfully integrated with ontologies and uses explicit representations to reason about different possible worlds (or Answer Sets). Hence, I think that ASP is a reasonable candidate for generating sets of different explanations for planning decisions.

Finally, I believe that the design and the development of novel HRI interfaces that make interpretable models accessible to users and developers is an open problem which should be addressed by researchers from various disciplines including (Cognitive) Psychology, Computer Science (HCI), and Robotics.

To summarize, I believe that autonomous robots (and other AI systems) should be equipped with the capability to 'explain' their own behavior and their own decisions. I suspect that such explanations will lead to a better understanding of robot systems in general. Thereby, user acceptance will be increased and trust in robot systems will be build up. Additionally, developers will benefit from transparent planning and decision processes when analyzing the behavior and the performance of complex robot systems.

3.20 Probabilistic Planning for Mobile Robots with Formal Guarantees

Bruno Lacerda (University of Birmingham, GB)

In recent years, the field of mobile service robots has witnessed important developments in terms of the ability for robust deployments in real life environments. Thus, we are quickly approaching an era where robot systems are regularly deployed among humans as an extra tool to improve productivity, for example, in office environments. Robots in such scenarios can perform a range of tasks such as fetch-and-carry, or security checks. In order to perform such deployments, one needs the ability to quickly generate robust and efficient plans for large scale systems. On top of that, another aspect I see as a crucial element in any safe, robust and efficient design/deployment loop is the ability to provide formal guarantees of performance for such plans. For example, guarantee that a robot will never navigate into dangerous regions of the environment; provide a value for the probability of a task being successfully completed; or give an expectation of the time that the execution of a task will take.

I have been researching the use of probabilistic model checking techniques for the generation of high-level plans for mobile robots with probabilistic formal guarantees, and using such plans on real life mobile service robot deployments. In parallel with the research on probabilistic model checking, and using many similar techniques, there have been efforts from the artificial intelligence and planning community on sequential decision making under uncertainty. This research generally does not provide formal guarantees. On the other hand, it focuses on fast generation of plans for large problems, using approximation techniques, something which probabilistic model checking approaches struggle with. Broadly speaking, even though probabilistic model checking and sequential decision making use many of the same underlying models, historically, the point of view of each field has been slightly different. A key point of my research agenda is to bring ideas from these two fields together, applying them to the deployment of safe and robust robot deployments in the real world.

Finally, I believe that extending probabilistic model checking techniques to multi-robot systems can yield very significant contributions to the field. In particular, there are many works applying sequential decision making techniques to multi-agent coordination. Building on those, ad also extending single-robot verification techniques in order to provide team level guarantees at different levels of abstraction is currently my main research goal. These techniques will combine approaches from sequential decision making and probabilistic verification in order to generate policies for task allocation and multi-robot coordination, with attached probabilistic guarantees, such as "regardless of the state of the team, there will always be one robot able to get to reception within 5 min, with probability 0.95". The main challenges in order to achieve this goal are to correctly apply different techniques in a well founded way, such that the overall framework can scale while still being able to provide meaningful guarantees over both individual robots, and overall team performance.

3.21 Planning for Persistent Autonomy: Where are we struggling?

Daniele Magazzeni (King's College London, GB)

AI Planning is about determining actions before doing them, anticipating the things that will need to be done and preparing for them. Planners use domain-independent heuristics to guide the search in huge state spaces.

Recently, AI Planning has been successfully applied to handle complex systems. PDDL+ is the formalism used to describe hybrid systems, and allows the modelling of the differential equations governing the continuous behaviour of the system. This talk provides an overview of how PDDL+ can be used to model robotics and autonomous systems; presents a new PDDL+ planner based on SMT and the ROSPlan framework for planning with ROS; highlights some open challenges on the integration between planning and robotics.

3.22 Temporal Planning for Execution

Lenka Mudrova (University of Birmingham, GB)

Many researchers dream about a mobile service autonomous robot who will roam our work environments and homes and assist us with our every day tasks. If we pretend that robotics has solved all "low-level" problems to make such a robot possible (hence the robot has certain capabilities, such as moving around, grasping objects, opening doors, observing and recognising, understanding speech, etc.) then we face at least the following questions.

1. *How* the robot can *plan* its behaviour in order to be able to perform required tasks given by a human?
2. *When* the robot should act in order to satisfy human's time constraints?
3. How to *execute* obtained plan in more robust sense, i.e., the robot is not stopping and re-thinking everything all the time...
4. How to *react* to situations that can go wrong in the execution? How to react when experiencing *unknown unknowns*?

In my currect research, Ive focused on giving some answers to Questions 1 and 2, developing an approach based on merging of partial order plans with durative actions, that can quickly and effectively generate a plan for a set of independent tasks. This plan exploits some of the synergies and demands of the plans for each single task, such as common locations where certain actions should be executed. This approach also handles situations when a task is required to be satisfied within a time window, and the partial order of the plan is a strong benefit for execution, when the final plan can be joined online in reaction to the current observations.

In order to make a progress with Questions 3 and 4, I think the community needs to move away from the current evaluation type "it runs" to benchmarking. As benchmarking in the real world is hard due to influence of many uncontrollable events, I propose to focus on developing benchmarking domains in simulations (using standard robot simulators) where different aspect affecting execution can be plugged in and repeat it many times under same conditions. Hence, more discusstion is needed about what are the aspects to be monitor during execution and modelled in such a benchmark domain.

3.23 Symbiotic Human Robot Planning

Daniele Nardi (Sapienza University of Rome, IT)

While operating in domestic environments, robots will necessarily face difficulties not envisioned by their developers. Moreover, the tasks to be performed by a robot will often have to be specialized and/or adapted to the needs of specific users and specific environments. Hence, the conventional approach to planning, based on a fixed action specification does not seem a suitable modeling tool.

Learning how to operate by interacting with the user seems a key enabling feature to support the introduction of robots in everyday environments. Symbiotic autonomy is a recently introduced viewpoint where the user should help the robot to improve its performance and to perform tasks otherwise not achievable. This novel perspective to the design of intelligent robots leads to a number of interesting research questions that are related to planning. First, the robot should plan including speech acts and, more specifically, requests for help from the user. Second, the robot can learn from the user plans to accomplish complex task. Third, the robot can learn from action/plan failures by requesting explanations to the user.

Our aim is to explore the above research question and illustrate some initial contributions following this approach. In particular, we present a novel approach for learning, through the interaction with the user, complex task descriptions that are defined as a combination of primitive actions. The proposed approach makes a significant step forward by allowing task descriptions parametric with respect to domain specific semantic categories. Moreover, by mapping the task representation into a task representation language, we are able to express complex execution paradigms and to revise the learned tasks in a high-level fashion. The approach is implemented in multiple practical test cases with a service robot.

3.24 Towards an Integrated Approach to Planning and Execution

Tim Niemüller (RWTH Aachen, DE) and Gerhard Lakemeyer (RWTH Aachen, DE)

Challenges

Building a robotics system is inherently an integration challenge. A diverse set of software components must be combined and the interaction with the physical world places high demands on robustness and fault tolerance. Still, task-level planning and reasoning for autonomous mobile robots – that claims to help in solving in some of these challenges by automatic and flexible behavior design – is still the exception rather than the norm. A part of the problem is that the scope of the planning community often ends once the plan has been generated. In the robotics community, on the other hand, most researchers are concerned with other components such as perception, navigation, or manipulation. The modeling and integration overhead for planning systems often appears considerable even for small problems, and unable to scale to larger ones.

Therefore, from our perspective one of the fundamental challenges for the closer cooperation and mutual benefit of both such communities is an integrated approach to planning and execution. That starts with good craftsmanship to design and implement the appropriate software interfaces. But it also contains research questions, such as what would a unified language for planning and execution look like, and what would be its model and semantics? What would be an accurate and expressive representation of plans that allows to combine, for example, classical and temporal plans for a common executive and to choose the appropriate planner depending on the sub-problem to solve? What does execution monitoring for generated plans mean in the presence of uncertainty and contingencies?

Some Pieces of the Puzzle

To tackle these questions, efforts are required in both communities. While the planning community's first and foremost goal is to efficiently generate plans, for the robotics community task-level behavior is often only means to test and demonstrate other components. Bringing these worlds together requires dedicated work on the interface part, the integration of efficient planning systems and execution and execution monitoring of generated plans.

An observation is that execution of plans (handled by an executive of some sort) is often only an afterthought, if at all. It often requires interfacing with planners that produce output in a mostly non-unified format (contrary to the somewhat unified input language based on PDDL), making it harder to replace a specific planning system and also leading to many ad-hoc solutions if planning is used on a robot. We think that a unified language that combines the definition of a planning domain and problem, and also of the execution and monitoring of the resulting plans is desirable. We have made first steps with GologCP [5] which builds on ideas for automatic Golog/PDDL translation [3] and continual planning [2] to create an integrated framework to planning and execution. The results show a significant performance improvement and sound modeling. However, a language more similar to PDDL or other planning languages might be desirable, for example, building on PRS ops [1].

A generalized representation of plans might be desirable. There have been systems already using simple temporal networks (STN) on the planning [4] and the execution [6] side. It can represent simple or more complex plans and thus scale with the capabilities of the planning system. We are currently investigating the possibility to use a classical planner to solve sub-problems in a multi-robot context, and then use additional information to transform the resulting sequence into an STN to allow for parallel execution of parts of the plan.

It seems useful if the execution of plans itself was more carefully modeled including the interactions between planning and execution. This might also include extensions for plan repair if re-planning is too costly, starting execution already once a (likely) prefix of the plan has been determined, or to include assertions as in continual planning, which are conditionally expanded sub-plans that allow to cope, for example, with incomplete knowledge (postponing it to be a run-time decision, requiring a closer integration).

While we do not suggest giving up the separation between planning and execution generally, we think it is still worth investigating the possible interactions of the two processes, and benefits or drawbacks on making these more explicit, fine-grained, and more expressive.

Evaluation Scenario

The integration of planning and execution demands different evaluation scenarios as are used, for example, in the International Planning Competition. Autonomous mobile robotics scenarios naturally require such a closer integration as the environments are typically dynamic, short reaction times are necessary, and there are many uncertainties. In cooperation with Karpas, Vaquero, and Timmons we therefore proposed a Planning Competition for Logistics

Robots In Simulation [7]. It provides a suitable scenario at a comprehensible size. It is to be understood as a first step and we explicitly deem other scenarios relevant and useful. The chose scenario is based on the RoboCup Logistics League [8], an established real-world robotics league that focuses on production in modern manufacturing environments downscaled and build with readily available hardware. It has a natural focus on planning and reasoning systems for multirobot coordination and cooperation. Performing the simulation in simulation (at least initially) allows to easily adjust complexity, size, and duration.

References

1 Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An architecture for autonomy. The International Journal of Robotics Research 17(4).

2 Brenner, M., and Nebel, B. 2009. Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems 19(3).

3 Claßen, J.; Röger, G.; Lakemeyer, G.; and Nebel, B. 2012. PLATAS – integrating planning and the action language Golog. KI – Künstliche Intelligenz 26(1).

4 Halsey, K.; Long, D.; and Fox, M. 2004. CRIKEY – A Temporal Planner Looking at the Integration of Scheduling and Planning. In Workshop on Integrating Planning into Scheduling at 13th International Conference on Automated Planning and Scheduling (ICAPS).

5 Hofmann, T.; Niemueller, T.; Claßen, J.; and Lakemeyer, G. 2016. Continual Planning in Golog. In Proceedings of the 30th Conference on Artificial Intelligence (AAAI).

6 Levine, S. J., and Williams, B. C. 2014. Concurrent plan recognition and execution for human-robot teams. In Int. Conf. on Automated Planning and Scheduling (ICAPS).

7 Niemueller, T.; Karpas, E.; Vaquero, T.; and Timmons, E. 2016. Planning Competition for Logistics Robots in Simulation. In WS on Planning and Robotics (PlanRob) at Int. Conf. on Automated Planning and Scheduling (ICAPS).

8 Niemueller, T.; Lakemeyer, G.; and Ferrein, A. 2015. The RoboCup Logistics League as a Benchmark for Planning in Robotics. In WS on Planning and Robotics (PlanRob) at Int. Conf. on Aut. Planning and Scheduling (ICAPS).

3.25 How much reliable are plan-based controllers for autonomous robots?

Andrea Orlandini (CNR – Rome, IT)

Decisional autonomy is considered among one of the key system abilities for robotics applications. This entails robotics platforms to be endowed with a wide set of automated reasoning capabilities to be implemented by means of suitable technologies. Among these, automated planning and scheduling (P&S) technology plays a crucial role.

In general, automated P&S systems are finding increased application in real-world mission critical systems that operate under high levels of unpredictability. Given a description of a desired goal, and a model of possible actions and their causal/temporal constraints, the planning problem consists of finding a plan, which is a sequence of actions, the execution of which is calculated to lead to the goal state under normal circumstances. Such technology can be used to generate plans to control a plant (for example a robot), driven by goals often issued by humans. Such technology is occasionally referred to as model--based autonomy. Then, a P&S system takes as input a domain model and a goal, and produces a plan of actions to be executed, which will achieve the goal. A P&S system typically also offers plan execution and monitoring engines.

To foster effective use of Automated P&S systems in (near future) robotics applications such as, for instance, service robots, it is of great importance to significantly increase the trust of end users in such technology.

On one hand, automated P&S systems often bring solutions which are neither "obvious" nor immediately acceptable for them. This is mainly because these tools directly reason on causal, temporal and resource constraints; moreover, they employ resolution processes designed to optimize the solution with respect to non trivial evaluation functions. On the other hand, due to the non--deterministic nature of planning problems, it is a challenge to construct correct and reliable P&S systems, including, for example, declarative domain models. That is, it is not straightforward to guarantee the correctness/reliability of P&S systems.

In this regard, Verification and validation (V&V) techniques may represent a complementary technology with respect to P&S, that contribute to develop richer software environments to synthesize a new generation of robust problem--solving applications. The aim of this talk is to discuss open issues related to V&V techniques in P&S considering multiple needs, i.e., considering V&V of domain models, V&V of plans, V&V of plan executions, V&V of planners, V&V of plan execution engines and V&V of plan execution monitors.

References

1 Bensalem S, Havelund K, Orlandini A (2014) Verification and validation meet planning and scheduling. International Journal on Software Tools for Technology Transfer 16(1):1–12.

2 A. Cesta, A. Finzi, S. Fratini, A. Orlandini, and E. Tronci, (2010) Validation and Verification Issues in a Timeline-Based Planning System, Knowledge Engineering Review, vol. 25, no. 3, pp. 299–318.

3 Orlandini A, Finzi A, Cesta A, Fratini S (2011) TGA-based controllers for flexible plan execution. In: KI 2011: Advances in Artificial Intelligence, Proceedings of the 34th Annual German Conference on AI, Springer, LNCS, vol 7006, pp. 233–245.

4 Cesta A, Finzi A, Fratini S, Orlandini A, Tronci E (2010) Analyzing flexible timeline-based plans. In: Proceedings of the 19th European Conference on Artificial Intelligence (ECAI 2010), pp. 471–476.

3.26 Some practical issues for social consumer robots: an industrial perspective

Amit Kumar Pandey (Aldebaran Robotics – Paris, FR)

Personal social robot for everyone is the next big thing in the history of robotics. It is the time when robots are entering into our day to day life. And we, the human, together with such social robots, are converging towards creating an intelligent and embodied eco-system of living, where robots will coexist with us in harmony, for a smarter, healthier, safer and happier life. There are some key ingredients for such robots to be a successful personal and social robots. Such robots are expected to behave in socially accepted and socially expected manners. For this to achieve, Social Intelligence of robots will be the paramount.

For achieving Social Intelligence in robots, there is a great need towards developing coherent theoretical and functional framework, by identifying the basic ingredients of development of social skills. Some of those are social interaction, situation assessment, social learning, socially-aware manipulation & navigation. For each of these aspects, the robot has to plan and act accordingly to fulfil the needs, while taking into account that it is operating in a human-centered environment with potentially human around it. This creates a new era of planning problem, which goes beyond the mere safety aspect of planning towards socially-aware aspects. For example, the robot has to plan the interaction action, for it to appear social, the robot has to chose where to focus and what to "perceive" to be useful for the situation and interaction, the robot has to react to stimuli, the robot has to understand the meaning of the day to day tasks, so that it can plan to perform them differently in different situations, without the need of pre-programming for each and every situation it can encounter in daily life, the robot has to be able to incorporate high-level human-oriented and Social constraints in its manipulation and navigation plannings, and should be able to come up with shared plans if necessary, (and by involving human) to achieve a task in socially intelligent manner. Further, all these have to be achieved with the additional constraint of being realtime, intuitive and for real environment.

The talk will illustrate such issues, through some of the use cases for social robot grounded with some European Projects. It will try to provide a generalized definition of action from Human-Robot Interaction and Socially Intelligent Robot perspectives. This will be followed by feedback from real users and discuss some of the immediate multi-disciplinary R&D challenges and needs from industrial perspective, and some initial results towards solving them, including planning for interaction, perception, manipulation & navigation, and highlighted human-in-the-loop based learning aspects of robots for understanding task semantics, complex affordances, and being proactive. The talk will conclude with some open and grand challenges ahead, for us, the interdisciplinary community, to brainstorm and solve.

3.27 Multirobot coordination

Simon Parsons (King's College London, GB)

This short paper describes progress on the challenging problem of planning and executing missions with teams of robots. The overall goal of this work is as follows:

> Given a mission specification and access to a group of robots with a range of abilities, first select a team of robots, some subset of the full group, which can achieve the mission. Then construct a plan for achieving the mission, distributing tasks to individual team members. Finally, execute the mission, monitoring the progress of the plan, and adjusting it if and when that is necessary[1].

While we have made progress on several of the elements, we are still a considerable way from the goal.

[1] I think of this as the Mission Impossible problem since it was a major feature of the plot of pretty much every episode of the early seasons of the 1960s TV show. Later series dropped the "Dossier Scene" in which the team leader picked the members of his team according to their skills.

The area on which we have made the most progress in recent years is that of task allocation. Provided that the mission is specified as a set of sub-tasks, task allocation is enough to generate a plan. (A necessary extension to this work is being able to handle the decomposition of missions into tasks in the case in which a mission is not specified so conveniently.) In [2, 4, 3], we have looked at task allocation mechanisms in a range of scenarios. These range from the simple case in which all tasks are known at the start, all tasks require one robot, and all tasks can be carried out by every robot [2], through the case in which tasks are given to the team over time [4] to the case in which tasks can require more than one robot, and there are constraints between tasks [3]. The approaches that we used were all market-based, in which team members bid for tasks based on the cost to them of executing the tasks (in the scenarios we have looked at, cost is related to the distance of the robot from the task, fuel cost if you will). More work is necessary here, for example to allow robots to switch tasks when that is appropriate, and to handle truely heterogeneous capabilities.

The other area in which we have made progress, overlaps somewhat with what was just described. In this work, [6, 7] we looked at constructing a plan for a team in a distributed fashion, here using standard planning representations. The advantage of constructing a plan like this, is that team members can make use of local information (see [5] for an example of how exploiting local information can be advantageous). Once a plan is constructed, team members can then monitor their progress against it, and can decide whether it is necessary to contact their teammates to ensure that joint activities are correctly coordinated[2].

The major area in which there is still work to do (in addition to the areas already mentioned) is the one that corresponds to the first element of the description above – from a mission specification, and a set of potential members, identify the team members with the necessary skills to complete the task. This could be viewed as a planning task: pick a subset of the group, see if there is a feasible plan to complete the mission, if not then try another subset. However, a more attractive approach is one which views this as both a knowledge representation problem (how to represent a mission, and the capabilities of a team member) and a problem of reasoning at different levels of granularity, since it should not be necessary to fully plan out the completion of a mission to select a team that is capable of achieving it.

References

1 S. Parsons, S. Poltrock, H. Bowyer, and Y. Tang. Analysis of a recorded team coordination dialogue. In *Proceedings of the Second Annual Conference of the International Technology Alliance*, London, 2008.

2 Eric Schneider, Ofear Balas, A Tuna Özgelen, Elizabeth I Sklar, and Simon Parsons. Evaluating auction-based task allocation in multi-robot teams. In *Workshop on Autonomous Robots and Multirobot Systems (ARMS) at Autonomous Agents and MultiAgent Systems (AAMAS)*, Paris, France, May 2014.

3 Eric Schneider, Elizabeth I. Sklar, and Simon Parsons. Evaluating multi-robot teamwork in parameterised environments. In Lyuba Alboul, Dana Damian, and Jonathan M. Aitken, editors, *Towards Autonomous Robotic Systems: 17th Annual Conference, TAROS 2016*, pages 301–313. Springer International Publishing, 2016.

4 Eric Schneider, Elizabeth I Sklar, Simon Parsons, and A Tuna Özgelen. Auction-based task allocation for multi-robot teams in dynamic environments. In *Towards Autonomous Robotic*

[2] In human teams, a large amount of the communication seems to be status updates [1]. By updating only when necessary, we can save communication bandwidth.

Systems: 16th Annual Conference, TAROS 2015, pages 246–257. Springer International Publishing, 2015.

5 E. I. Sklar and M. Q. Azhar. Argumentation-based dialogue games for shared control in human-robot systems. *Journal of Human-Robot Interaction*, 4(3):120–148, 2015.

6 Y. Tang. *A Symbolic Exploration of the Joint State Space and the Underlying Argumentation-based Reasoning Processes for Multiagent Planning.* PhD thesis, Graduate Center, City University of New York, 2012.

7 Y. Tang, T. Norman, and S. Parsons. A model for integrating dialogue and the execution of joint plans. In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems*, Budapest, Hungary, 2009.

3.28 Knowledge-level planning for Human-Robot Interaction

Ron Petrick (Heriot-Watt University – Edinburgh, GB)

At a basic level, the automated planning problem is one of context-dependent action selection: given an initial state, a domain description, and a set of goals, generate a sequence of actions whose execution will bring about the goal conditions. However, the problem of action selection is not unique to automated planning. One important field where this issue is also of primary concern is that of spoken dialogue systems, whose tools play a central role in addressing the problem of human-robot interaction. At the heart of the dialogue system is the interaction manager whose primary task is to carry out a form of action selection: based on the current state of an interaction, the interaction manager makes a high-level decision as to which spoken, non-verbal, and task-based actions the system should apply. An important aspect of research in this area has been the development of toolkits to support the construction of end-toend systems. Given the parallels between the planning and dialogue tasks, our recent work has explored the application of automated planning techniques to human-robot interaction (HRI) as an alternative to standard dialogue system toolkits (such as Trindikit, COLLAGEN, IrisTK, OpenDial, among others).

While the link between natural language processing and automated planning has a long tradition, going back to at least the 1980s, in recent years the two communities have focused on different problems and solutions, with planning for natural language problems largely overlooked in favour of more specialpurpose solutions. For instance, the interactive systems toolkits attempt to offer a one-stop solution for system building combining action selection, representation, and technical architectures. In contrast, the planning community has focused on defining domains in common representation languages like PDDL and comparing different domain-independent strategies within this context through events like the International Planning Competitions; the study of the representation languages themselves has also led to a better understanding of the trade-offs between different representations.

Our own work in this area has focused on applying domain-independent knowledge-level planning techniques to the problem of action selection in human-robot interaction. In particular, the beliefs of the planning agent (robot) about the world and other agents are represented, and sensing actions are used to model certain types of information-gathering speech acts. Task-based actions are also planned using the same general-purpose planning mechanisms.

However, the problem of human-robot interaction also offers some wider opportunities and lessons for the planning community. First, the presence of action selection at the core of interaction management offers the obvious possibility of applying other types of planning techniques. Second, the nature of the applications addressed by many HRI systems also highlights the importance of building real-world systems – an area that has gained wider traction in the planning community but one that is still somewhat outside the mainstream of most planning research. Finally, the process for evaluating robot-based dialogue systems, and in particular the role of human users, also presents new directions and challenges for planning.

3.29 Goal Reasoning for Robotics

Mark Roberts (Naval Research – Washington, US)

Goals are the hinge-pins of deliberative behavior. Whether explicit (e.g., as a symbolic structure), implicit (e.g., as a reward or error signal), provided by a designer, or learned over time, goals unify motivations and action. They can be used to prune the search for solutions, to label or query persistent storage, or to structure learning from experience. Researchers have recently proposed a synthesis of the research disciplines that examine deliberation about goals, calling it goal reasoning: the ability of an agent to determine, pursue, and modify its own goals in response to notable events. In this talk, we identify three challenges we have studied with respect to goal reasoning systems in robotics. We will present the goal lifecycle, a formal model of goal reasoning built on goal-task networks, and showcase its implementation in a system called ActorSim, which links several robotic and virtual autonomous systems.

To perform complex tasks, a team of robots requires both reactive and deliberative planning. For reactive control, a restricted variant of Linear Temporal Logic called General Reactivity(1) can be used to synthesize correct-by-construction controllers in polynomial time, but they often ignore time and resource constraints to maintain tractable synthesis. For deliberation, hierarchical goal reasoning can be used to reason about time and resources. However, the coordination of reactive control and deliberation remains a challenge, which we accomplish through a set of Coordination Variables. We integrate these two approaches in the Situated Decision Process (SDP), a predecessor of part of ActorSim. The SDP will allow an Operator to control a team of semi-autonomous vehicles performing information gathering tasks for Foreign Disaster Relief operations. We demonstrate that the SDP responds to a dynamic, open world while ensuring that vehicles eventually perform their commanded actions.

In complex and dynamic scenarios, autonomous vehicles often need to intelligently adapt their behavior to unexpected events. We extend the ActorSim to include information measures and expectations used by the vehicles to assess their performance. This system, called Goal Reasoning with Information Measures, is demonstrated using a disaster relief scenario in which a small team of vehicles is tasked with surveying a predefined set of geographical regions. Additionally, a preliminary study shows that the inclusion of resolution strategies increases the likelihood that it successfully finishes its goals.

Finally, robots are increasingly performing well on focused tasks in constrained worlds over increasing time horizons. We argue that goal reasoning is essential as autonomous systems, robotic or virtual, transition to operating in open worlds over time horizons of months or years while maintaining hundreds of goals that vary in duration and priority. But to achieve long-duration autonomy presents two new challenges. First, existing robots can have relatively short life spans, limiting progress. As a contingency, we plan to study long-duration autonomy in 3D game engines similar to the way in which the Robocup simulator served as an early testbed until robotics systems became more capable and reliable. Second, we must enable a robot to store, access, and learn over very long time horizons of weeks, months, or even years. This presents challenges not only in how to capture this information but also in how to maintain an ever growing knowledge base, retrieve relevant memories and experience, and update it with new knowledge. We argue that cognitive structures are needed to manage long-term memory structures, focus effort, and derive curricula. Further, we argue that hybrid model-based and reactive control architectures must be leveraged because each excels at complementary tasks in a robot.

3.30 Effective Hybrid Planners for robotics

Enrico Scala (Australian National University – Canberra, AU)

A fundamental building block towards an effective integration and exploitation of AI planning in robotics systems requires reasoning mechanisms over mixed representations combining logical and numeric constraints. This becomes apparent even in very simple problems of integration between task and motion planning, where geometric and causal reasoning have to be considered in an intertwined way [15, 5]. Decoupling them may result in very poor performances.

Planning for such hybrid systems is a very hard computational problem (even very restricted models are NP-hard to solve) since it requires an intertwined reasoning over discrete and continuous domain variables along with changes of the states that can also be both discrete and continuous. Work to better handle such problems has been done [4, 3, 14, 8, 7], but more work is still needed to scale up to realistic size problem.

In my research I am investigating different methods to deal with a discretised version of this class of problems: forward state space planning via heuristic search, compilation to satisfiability modulo theory, robust plan execution and plan repair ([9, 10, 11, 13, 12]). These works adapt and extend well known classical planning techniques to the hybrid case in different ways, all of them starting from the key observation that a powerful computational representation of the hybrid case (including processes) is that of sequential numeric planning with global constraints. Improving on (and exploiting) the reasoning about the exposed numeric structures of the problem becomes of crucial importance. By adapting and extending previous work done in classical planning, these works do actually attempt to solve classical (task) and numeric (motion) planning in an integrated way.

These approaches though suffer in some situations, and in particular when many and complex constraints need to be enforced. I see this seminar as a great opportunity to engage discussions with people working on constrained-based/timeline-based planning [1, 2] and/or motion planning, to study synergies and exchange of methods among these approaches.

References

1 Javier Barreiro, Matthew Boyce, Minh Do, Jeremy Frank, Michael Iatauro, Tatiana Kichkaylo, Paul Morris, James Ong, Emilio Remolina, Tristan Smith, et al. Europa: a platform for ai planning, scheduling, constraint programming, and optimization. *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2012.

2 John L. Bresina, Ari K. JÃ³nsson, Paul H. Morris, and Kanna Rajan. Mixed-initiative planning in mapgen: Capabilities and shortcomings. In *In Proceedings of the ICAPS-05 Workshop on Mixed-initiative Planning and Scheduling*, 2005.

3 Michael Cashmore, Maria Fox, Derek Long, and Daniele Magazzeni. A compilation of the full PDDL+ language into SMT. In *Planning for Hybrid Systems, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 13, 2016.*, 2016.

4 Giuseppe Della Penna, Daniele Magazzeni, Fabio Mercorio, and Benedetto Intrigila. Upmurphi: A tool for universal planning on PDDL+ problems. In *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19–23, 2009*, 2009.

5 Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trúg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *Proc. of International Conference on Automated Planning and Scheduling (ICAPS-09)*, 2009.

6 Maria Fox and Derek Long. Modelling mixed discrete-continuous domains for planning. *J. Artif. Intell. Res. (JAIR)*, 27:235–297, 2006.

7 Drew V. McDermott. Reasoning about autonomous processes in an estimated-regression planner. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003), June 9-13, 2003, Trento, Italy*, pages 143–152, 2003.

8 Wiktor Piotrowski, Maria Fox, Derek Long, Daniele Magazzeni, and Fabio Mercorio. Heuristic planning for pddl+ domains. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York (NY), USA*, 2016.

9 Enrico Scala, Patrik Haslum, and Sylvie Thiébaux. Heuristics for numeric planning via subgoaling. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3228–3234, 2016.

10 Enrico Scala, Patrik Haslum, Sylvie Thiébaux, and Miquel Ramírez. Interval-based relaxation for general numeric planning. In *ECAI 2016 – 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 655–663, 2016.

11 Enrico Scala, Miquel Ramírez, Patrik Haslum, and Sylvie Thiébaux. Numeric planning with disjunctive global constraints via SMT. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, pages 276–284, 2016.

12 Enrico Scala and Pietro Torasso. Proactive and reactive reconfiguration for the robust execution of multi modality plans. In *ECAI 2014 – 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, pages 783–788, 2014.

13 Enrico Scala and Pietro Torasso. Deordering and numeric macro actions for plan repair. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 1673–1681, 2015.

14 Ji-Ae Shin and Ernest Davis. Processes and continuous change in a sat-based planner. *Artificial Intelligence*, 166(1-2):194–253, 2005.

15 Siddharth Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stuart Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 639–646. IEEE, 2014.

3.31 Planning for Open-ended Missions

Matthias Scheutz (Tufts University – Medford, US)

Many envisioned applications for future robots (e.g., robots for search and rescue domains, space environments, etc.) will take place in contexts that are "open", i.e., where aspects of the task and the mission are not known ahead of time and where unforeseen events can alter mission planning and execution. Most current robotic architectures, however, are only able to deal in a very limited way with open-world aspects. In particular, most planning algorithms assume that the domain model for the planner is given in its entirety, so that planning really amounts to search over the states and actions defined by the model. We argue that such an assumption is not warranted in open worlds and that task and action planners integrated into robotic architectures need to be able to intrinsically cope with unknown aspects of the domain model (e.g., goals that involved entities where the planner does not know all relevant aspects of the entity).

3.32 Planning with Incomplete models

Reid Simmons (NSF – Arlington, US)

Planners need models in order to predict the effects of actions. We all know, however, that any model of the real world is merely an incomplete representation. While some models may be higher fidelity than others, all are deficient in some respects, or another. Thus, planning can never accurately predict all the effects of actions in every context.

In robotics, feedback controllers are often used to address this deficiency – plans provide higherlevel of abstractions and the low-level controller behaviors are designed to achieve the intended effects of the plan. Clearly, however, there are drawbacks with this approach; for instance, suboptimal plans may be produced that cause the controllers to expend much more time/energy than would otherwise be necessary, and, in the worst case, the plans may fail altogether because the controllers are not applicable for the contexts in which the plans have put them.

The question, then, is how to deal with the inevitable fact of planning with incomplete models? I suggest three approaches that can help – planning with probabilistic models, learning to improve models, and planning with multiple models.

Planning with probabilistic models, such as MDPs or POMDPs, is now considered fairly standard in robotics. The idea is to "mask" the incompleteness of the models using a distribution of possible effects of actions, and to plan to achieve a metric such as maximizing (or exceeding some threshold of) probability of plan achievement, maximizing expected reward, or maximizing expected utility, taking risk into account. Such approaches produce plans that work well, on average, provided that the reward function, and transition and observation probabilities, are close to their true values.

This leads to learning to improve models. I start with the assumption that the models are reasonably accurate, but do not completely reflect reality. Here, experience-based learning can be used to improve transition and observation probabilities, and techniques such as learning from demonstration and inverse reinforcement learning can be used to improve the reward function. In addition, by modeling the uncertainty in the model parameters, active learning can be used to guide the agent to situations where it can efficiently learn the parameters it is most uncertain about.

Things get more complicated, though, in situations where models have hidden (latent) state. Effective methods for learning such models exist, such as EM or spectral methods, but they typically need large amounts of training data. We are exploring an alternate, data-efficient, approach that uses statistical tests to identify regions of the state space that appear to be drawn from different distributions than other parts of the state space. The approach incrementally searches ellipsoidal regions of the state space to find the contexts in which the observed distribution differs significantly from the model. For instance, it might discover that the robot typically slips more than expected when turning left at a high velocity in a given area of the building. While we, as people, may understand that is because the area is tiled, to the robot it is sufficient to have improved its navigational model in the given context, which enables it to create better plans (e.g., by slowing down in that area, or avoiding it altogether).

Finally, we acknowledge that different models often have different strengths and weaknesses. By choosing the most appropriate model for a particular planning task, one may produce better plans for that particular situation. To that end, we are exploring an approach that uses of a hierarchy of models. The idea is to plan first in a lower fidelity model, then check if the plan is valid in higher fidelity models. If not, it is assumed the higher fidelity model contains information relevant to a potential plan failure, and so the plan is patched using that model. In this way, the planner can use lower fidelity (and typically computationally less expensive) models, when appropriate, but still make use of higher fidelity models, when necessary.

This approach differs from standard abstraction and HTN planning in two major ways. First, unlike abstraction and HTN planning, any of the models in the hierarchy can produce a directly executable plan, thus, it is not necessary to plan in multiple models if the situation is “simple” enough. Second, rather than a linear, predefined ordering of models, as with abstraction planning, our approach supports a complete lattice of models. Different models in the lattice make different information explicit – for instance, one model may represent vehicle dynamics while another model may represent shape and material properties of the vehicle. Thus, choosing with which model to plan, in a given context, is a key issue with this approach, and one that we are actively researching.

While it is inevitable that all models of the real world are incomplete, to some extent, we can use that knowledge to our advantage in designing planners that explicitly handle sources of incompleteness, either probabilistically or through explicit model choice, and learners that can efficiently improve models through experience. The ultimate goal is to develop robotic systems that can reason about the incompleteness of their models and actively characterize and improve them, through interactions with the environment.

3.33 On the Shoulders of Giants: The Case for Modular Integration of Discrete Planners and Continuous Planners for Robotics

Siddharth Srivastava (UTRC – Berkeley, US)

The field of automated planning has its roots in planning for SHAKEY, a problem-solving robot created at SRI in the 70s. For the most part, the modern planning paradigm for autonomous robots assumes a similar architecture, making a clean separation between the abstract problem of task planning with discrete states and actions, and the problems of motion planning and control synthesis involving the computation of continuous trajectories in a high-dimensional configuration space. However, for the most part these fields have developed in mathematical isolation from each other. Naïve approaches for reconciling them (e.g., first computing a task plan and then "implementing" each action through a motion planner) result in solutions that are unexecutable because of the lossy abstractions required for constructing task-level models.

The situation in planning for robotics is thus not very different from the state of model checking just before the advent of SAT-modulo-theories (SMT) techniques. Many of the motivators for SMT research hold in our setting: we have efficient solvers (planners) at each level of abstraction; modeling entire tasks at the finest level of granularity is cumbersome when it is not impossible and typically results in computationally intractable problems. Indeed, from the point of view of making collective progress, developing new "task and motion" planners from scratch for various formulations (deterministic, non-deterministic or stochastic models of actuators with deterministic, non-deterministic or stochastic models of sensors for single or multiple agents), most of which are addressed independently in these fields, would be inefficient at best and reinvention in the worst case.

I believe that these observations provide evidence for new opportunities as well as challenges for innovative research in planning for robotics, geared towards the modular utilization of existing paradigms for discrete sequential decision making, motion planning, and control synthesis in a hierarchical planning paradigm. Opportunities stem from a perspective that allows us to learn from the development of SMT solvers. The challenges, and the corresponding domains of innovation, arise from the numerous aspects of planning for robotics that are not addressed in the existing theory for SMT solvers; a direct application of those techniques is unlikely to succeed. We need new research on rigorous methods for the synthesis and analysis of abstraction functions that translate planning problems between different levels of the hierarchy. Existing domain description languages require new constructs and semantics to succinctly and correctly express abstractions of sequences of low-level actions (existing representations lead to incorrect models and unexecutable solutions). Solutions to motion planning problems are continuous trajectories that need new symbolic representations suitable for high-level reasoning. Finally, uncertainty in sensing and actuation in each level of abstraction makes it harder to construct "lemmas" for use at the next higher level of abstraction.

3.34 Persistent, Instructable, Interruptible, Transparent Autonomy

Manuela Veloso (Carnegie Mellon University – Pittsburgh, US)

Besides the Roomba robots, and instances of service mobile robots in some hospital, public environments, there are not many autonomous robots that persist in real human environments. We have experienced the CoBot mobile robots that have persisted at Carnegie Mellon capable of autonomously performing navigation and service tasks for the last 5-6 years. (There are the Google-Uber-Baidu-etc self-driving cars, which are becoming a reality.) We are interested of further understanding how to better plan for such persistent "co-existing" robots.

The real world offers challenges to robot autonomy, at all levels, namely perceptual, cognitive, and actuation. Focusing on the real world where humans live, robots further concretely face navigation, timing, quality, and interaction challenges. Planning involves having models of the world, and models of actions. It is challenging, or most probably impossible, to get appropriate and correct models up front: the real world is dynamic, uncertain, personalized, and it includes other external, at best poorly modeled, actuators, such as other robots and people.

We are interested in the issues underlying robot autonomy, in particular planning, that needs to persist, exist, improve in the real world. Along this goal, we pursue research on instructing and correcting a robot, interrupting the execution of its plans, and making it be transparent about its actions and plans.

We have developed instruction graphs as a way for humans to provide verbal command-based instructions to become procedural plans. The robot is equipped with action and sensing primitives that the instruction graphs then organize in sequencing, conditionals, and looping constructs. Upon execution of the instructed graphs, humans can check that the plan is suitable or not, and correct it as needed. As the human knows the plan that is being executed, we conjecture that it may be easier for the human to correct the robot behavior than if the planning model were non-procedural actions. Such instruction graphs are independent of the robot platform. We have used instruction graphs for our CoBot and Baxter robots. We have also addressed the problem of acquiring a library of plans, as instruction graphs, for different tasks. The challenge is to recall a similar past learned plan and reuse it. We have also researched on generalizing different plans and proposing autocompletions of possible plans when a human is instructing or correcting a robot. We believe there is a lot left to do in terms of providing, generalizing, revising, and reusing plans.

We also research on the challenge of enabling a robot with the ability to replan when interrupted. When the robots generate plans to achieve their tasks, they then execute them determined with the sole goal of executing their plans. If CoBot is executing its plan to deliver a package to someone's office, and Manuela finds CoBot in the middle of the corridor, and she wants to tell "CoBot, thanks for the package, I got it!", she can't: CoBot will go all the way to the destination office to finish its delivery task. Tasks can of course be managed and interrupted remotely through an administrative interface, but not by naturally interrupting the robot. We have created an approach to interrupt a robot, but there is a lot left to do in terms of investigating the need to replan when and how, so that the robots can take input from humans about their plans and new task requests. In the general scenario of our research, in which robots encounter a wide variety of people, and not just the robot developers, such interruptible autonomy is a challenging research question, as the robot needs to be able to evaluate the requests and attend to them according to models of priorities, authority, level of accomplishment and type of tasks under execution and scheduled. Replanning becomes a question, which is not just a function of the failure of preconditions, but includes dynamic task optimization and model learning.

Autonomous robots that plan cannot be opaque to their human users. We research on methods to increase the transparency of the robot planners, such that the robot explains its choices, and reveals the actions selected, as well as possibly future actions. Interestingly, robot act according to plans that include future actions, so they can make their future actions known to humans, who could potentially better understand and possibly correct the intentions and plans of the robots. We have developed a verbalization algorithm which enables a robot to describe its experience in natural language. The robot can transform its planned and executed route into natural language. We also research on multiple techniques to enable a robot to be transparent to humans in their decisions, planning, and learning. We believe there is a lot left to do in terms of augmenting planning algorithms with the ability to increase their transparency towards humans. We research on expressive lights to improve the understanding of the robot's actions, on verbalization in different dimensions to generate varied descriptions of experience in natural language, and on augmenting video capturing of robot's plan execution with markings that aim at visually explaining the robot's performance.

In summary, we are interested in planning, as an integral part of a persistent autonomous mobile service robot, that needs to persistently interact with humans. We will discuss our research and open planning research directions in robot instructability, interruptibility, and transparency. Our underlying approach assumes that robots need to learn improved planning models over time with experience.

Participants

- Rachid Alami
LAAS – Toulouse, FR
- Iman Awaad
Hochschule Bonn-Rhein-Sieg – St. Augustin, DE
- Roman Bartak
Charles University – Prague, CZ
- Michael Beetz
Universität Bremen, DE
- Ronen I. Brafman
Ben Gurion University – Beer Sheva, IL
- Michael Cashmore
King's College London, GB
- Martin Davies
Guidance Automation Ltd – Leicester, GB
- Minh Do
NASA – Moffett Field, US
- Susan L. Epstein
City University of New York, US
- Alberto Finzi
University of Naples, IT
- Hector Geffner
UPF – Barcelona, ES
- Malik Ghallab
LAAS – Toulouse, FR
- Nick Hawes
University of Birmingham, GB
- Malte Helmert
Universität Basel, CH
- Andreas Hertle
Universität Freiburg, DE
- Joachim Hertzberg
Universität Osnabrück, DE
- Laura M. Hiatt
Naval Research Lab – Washington, DC, US
- Michael W. Hofbaur
Joanneum Research – Klagenfurt/Wörthersee, AT
- Jörg Hoffmann
Universität des Saarlandes, DE
- Felix Ingrand
LAAS – Toulouse, FR
- Luca Iocchi
Sapienza University of Rome, IT
- Gal A. Kaminka
Bar-Ilan University – Ramat Gan, IL
- Erez Karpas
Technion – Haifa, IL
- Oliver E. Kim
University of Birmingham, GB
- Sven Koenig
USC – Los Angeles, US
- Lars Kunze
University of Birmingham, GB
- Bruno Lacerda
University of Birmingham, GB
- Gerhard Lakemeyer
RWTH Aachen, DE
- Daniele Magazzeni
King's College London, GB
- Lenka Mudrova
University of Birmingham, GB
- Daniele Nardi
Sapienza University of Rome, IT
- Tim Niemüller
RWTH Aachen, DE
- Andrea Orlandini
CNR – Rome, IT
- Amit Kumar Pandey
Aldebaran Robotics – Paris, FR
- Simon Parsons
King's College London, GB
- Ron Petrick
Heriot-Watt University – Edinburgh, GB
- Mark Roberts
Naval Research – Washington, US
- Enrico Scala
Australian National University – Canberra, AU
- Matthias Scheutz
Tufts University – Medford, US
- Reid Simmons
NSF – Arlington, US
- Elizabeth Sklar
King's College London, GB
- Stephen Smith
Carnegie Mellon University – Pittsburgh, US
- Siddharth Srivastava
UTRC – Berkeley, US
- Manuela Veloso
Carnegie Mellon University – Pittsburgh, US
- Brian C. Williams
MIT – Cambridge, US

Report from Dagstuhl Seminar 17032

Network Function Virtualization in Software Defined Infrastructures

Edited by

David Hausheer[1], Oliver Hohlfeld[2], Diego R. López[3], Bruce MacDowell Maggs[4], and Costin Raiciu[5]

1 **TU Darmstadt, DE,** hausheer@ps.tu-darmstadt.de
2 **RWTH Aachen, DE,** hohlfeld@comsys.rwth-aachen.de
3 **Telefonica I+D – Seville, ES,** diego.r.lopez@telefonica.com
4 **Duke University – Durham, US,** bmm@cs.cmu.edu
5 **University Politehnica of Bucharest, RO,** costin.raiciu@cs.pub.ro

Abstract

The softwarization of networks by introducing concepts such as Software-Defined Networking (SDN) or Network Functions Virtualization (NFV) currently massively changes network management by enabling more flexible communication networks. The main goal of this seminar was to gather researchers from academia, industry, and standardization bodies to discuss a joint perspective on research questions in the field of NFV. This report contains talk summaries, reports on the discussion groups, as well as the personal statements and main challenges contributed by the seminar participants.

Seminar January 15–18, 2017 – http://www.dagstuhl.de/17032
1998 ACM Subject Classification Network Architecture and Design, Network Operations
Keywords and phrases flexible network management, network function virtualization, software-defined networking
Digital Object Identifier 10.4230/DagRep.7.1.74
Edited in cooperation with Leonhard Nobach

1 Executive Summary

David Hausheer
Oliver Hohlfeld
Diego R. López
Bruce MacDowell Maggs
Costin Raiciu

Network management currently undergoes massive changes towards realizing more flexible management of complex networks. Recent efforts include slicing data plane resources by using network (link) virtualization and applying operating system design principles in Software Defined Networking (SDN). Driven by network operators, network management principles are currently envisioned to be even further improved by virtualizing network functions which are currently realized in dedicated hardware appliances. The resulting Network Function Virtualization (NFV) paradigm abstracts network functions from dedicated hardware to

Network Function Virtualization in Software Defined Infrastructures, *Dagstuhl Reports*, Vol. 7, Issue 1, pp. 74–102
Editors: David Hausheer, Oliver Hohlfeld, Diego R. López, Bruce MacDowell Maggs, and Costin Raiciu
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

virtual machines running on commodity hardware and enables a Cloud-like network management. All of these efforts contribute to a softwarization of communication networks. This softwarization represents a significant change to network design and management by allowing the application of operating system design and software engineering principles to make network management more efficient, e.g., by enabling flexible and dynamic service provisioning.

Since the NFV efforts are currently mainly driven by carriers and standardization bodies, academic research is decoupled from the industry driven NFV attempts in redesigning network management. Due to this missing link to academic research, opportunities for groundbreaking research and broad impact in academia are currently missing out. This Dagstuhl Seminar thus gathered researchers from academia, industry, and standardization bodies to establish this missing link by fostering collaborations and joint research initiatives. Thus, a particular focus of the seminar was on identifying the diverse connections between industry driven NFV efforts and current academic networking research.

The seminar brought together 24 participants in January 2017 to discuss a potential NFV research agenda within 2.5 days. The program included different invited talks that provided an overview on selected aspects of NFV and lightning talks by each participants to provide first research questions and to sketch research directions. We summarize each talk in the remainder of this report. The main focus of the seminar was then the in-depth discussion the research areas identified in the lightning talks in several breakout sessions, which we also summarize. We closed the seminar by collecting and discussing several opinions from each participant: (i) lessons learned and surprises on NFV during the seminar and (ii) open research questions. We further collected controversial statements on NFV research and asked the seminar participants on whether they agree or disagree to each presented statement. We summarize the outcomes at the end of the report.

2 Table of Contents

Executive Summary

Madness Talks

Lightning Talks

Working Groups

Wrap-up

3 Madness Talks

3.1 Who Am I? What Am I doing here?

Diego R. López (Telefonica I+D – Seville, ES)

Challenges:

- Performance
- Security
- Design Patterns
- Deployment Patterns
- A software network discipline
- Evolve management
- Evolve architectures
- Evolve business models
- Evolve industry culture

3.2 Personal Introduction

Bruce MacDowell Maggs (Duke University – Durham, US)

- Largest DDOS Attacks by Year
- Leverage Compromised Home Cable Modems/Routers
- Account Takeover Campaign Attack Architecture
- Attacking IP Persistence: Finance Customer

3.3 Personal Introduction

Dirk Kutscher (Huawei Technologies – München, DE)

- Performance in the presence of heterogeneity and dynamic network conditions
- Rethinking collaboration of apps, transport and forwarding
- New forwarding abstractions and SDN control for that
- Enabling dynamic computation in the network securely
- Use cases: IoT, blending VoD and live streaming

3.4 Personal Introduction

Felipe Huici (NEC Laboratories Europe – Heidelberg, DE)

Despite years of research and a whole lot of noise made about the great benefits of NFV, especially in terms of lower OPEX/CAPEX, how to provide high utilization and performance isolation on x86 servers for production-level deployments have proven so tricky that most operators resort to throwing money at the problem: each NFV instance runs in its own core, ensuring reliable performance but woeful utilization. In this brief talk I make the case for applying machine learning techniques, and in particular reinforcement learning, to develop novel, more efficient NFV resource allocators.

3.5 Personal Introduction

Gabriela Gheorghe (PwC – Luxembourg, LU)

- Cloud & secure storage
- SDN security & network troubleshooting
- Policy-based security monitoring at network level

3.6 Performance Assessment of NFV

Thomas Zinner (Universität Würzburg, DE)

The variety of different virtualized network functions as well as the availability of diverse deployment options requires new means for their performance assessment. This includes appropriate benchmarking methods and tools, but also well-suited performance models based on methods like discrete-time analysis, queuing networks or the network calculus. The resulting models and corresponding key performance indicators will then enable an optimized placement of virtualized network functions.

3.7 NFV: Where are we?

James Kempf (Ericsson – Santa Clara, US)

The Gartner hype curve is a well-known way of classifying where a technology is in terms of deployment. Right now, NFV seems to be in the “Valley of Disillusionment”. This is a period in which the initially promised benefits of the technology fail to pan out and early adopters are left trying to figure out what benefit, if any, the technology has. With respect to NFV, it has promised that it would reduce operator costs and provide much faster development of new services. In 2016, the cost per bit for operators to deliver data increased above the revenue per bit (according to analyst Tom Nolan). Can NFV help solve this problem? NFV deployments have so far not shown promise in this regard. As for services, there have been a few (SD-WAN for example), and this area seems more promising. Over the next couple of years, we will see if NFV reaches the "Slope of Enlightenment" and the "Plateau of Everyday Use", and becomes a widely deployed and useful technology for operators and enterprises.

3.8 Checking Policy Compliance

Aaron Gember-Jacobson (Colgate University – Hamilton, US)

Joint work of Aaron Gember-Jacobson, Raajay Viswanathan, Aditya Akella, Ratul Mahajan, Chaithan Prakash, Robert Grandl, Junaid Khalid, Sourav Das
Main reference A. Gember-Jacobson, R. Viswanathan, A. Akella, R. Mahajan, "Fast Control Plane Analysis Using an Abstract Representation", in Proc. of the 2016 ACM SIGCOMM Conf., pp. 300–313, ACM, 2016.
URL http://dx.doi.org/10.1145/2934872.2934876

This talk highlights two major pieces of work I have done to improve the correctness of networks: OpenNF and ARC. The former is a framework for quickly and safely transferring NF state amidst traffic redistribution. The latter is an efficient control plane verifier that uses graphs to model the network's behavior and computes properties of these graphs to check that a policy is satisfied under arbitrary failures. An interesting area of future research is verifying that networks conform to more complex policies involving stateful NFs.

3.9 Challenges in Network Functions Virtualization

David Hausheer (TU Darmstadt, DE)

Our existing work on SDN/NFV covers Software-defined Multicast (SDM), NFVI benchmarking, and application scenarios of bare metal switches. Moreover, resource models for SDN/NFV data planes and seamless elasticity in hardware-accelerated NFV, as well as VNF state migration are of interest. SDN support of CDN networks and SDN/NFV machine learning have been partially tackled as well. Novel challenges include the discussion of relevant killer applications, functions, and use cases for NFV that show the greatest benefits. Also, what's going to be the long-term impact on hardware vendors, network providers, and end-users? Moreover, what are the biggest barriers for research? Finally, is NFV just another hype? And whats going to be next after NFV/SDN?

References

1 Julius Rückert, Jeremias Blendin, Rhaban Hark, David Hausheer: Flexible, Efficient, and Scalable Software-Defined Over-the-Top Multicast for ISP Environments with DynSDM. TNSM, 2016

2 Julius Rückert, Jeremias Blendin, David Hausheer: Software-Defined Multicast for Over-the-Top and Overlay-based Live Streaming in ISP Networks. JNSM Special Issue on Management of Software Defined Networks, July 2014.

3 Jeremias Blendin, Julius Rückert, Sascha Bleidner, David Hausheer: Taking the Sting out of Flow Update Peaks in Software-Defined Service Chaining. 2nd International Workshop on Management of SDN and NFV Systems (ManSDN/NFV 2015), November 2015.

4 Julius Rückert, Jeremias Blendin, Rhaban Hark, David Hausheer: DYNSDM: Dynamic and Flexible Software-Defined Multicast for ISP Environments. 11th International Conference on Network and Service Management (CNSM 2015), November 2015.

5 Jeremias Blendin, Julius Rückert, Tobias Volk, David Hausheer: Adaptive Software Defined Multicast. 1st IEEE Conference on Network Softwarization (NetSoft 2015), April 2015.

6 Leonhard Nobach, David Hausheer: Open Elastic Provisioning of Hardware Acceleration in NFV Environments. Workshop on Software-Defined Networking and Network Function Virtualization for Flexible Network Management (SDNFlex 2015), March 2015.

7 Jeremias Blendin, Julius Rückert, Nicolai Leymann, Georg Schyguda, David Hausheer: Position Paper: Software-Defined Network Service Chaining. EWSDN 2014, September 2014.

8 Julius Rückert, Roberto Bifulco, Muhammad Rizwan-Ul-Haq, Hans-Joerg Kolbe, David Hausheer: Flexible Traffic Management in Broadband Access Networks using Software Defined Networking. IEEE/IFIP Network Operations and Management Symposium (NOMS 2014), May 2014.

9 Leonhard Nobach, Oliver Hohlfeld, David Hausheer: New Kid on the Block: Network Functions Virtualization: From Big Boxes to Carrier Clouds. Computer Communication Review. July 2016.

10 Leonhard Nobach, Benedikt Rudolph, David Hausheer: Benefits of Conditional FPGA Provisioning for Virtualized Network Functions, SDNFlex Workshop, March 2017.

11 Jeremias Blendin, Yuriy Babenko, Dennis Kusidlo, Georg Schyguda, David Hausheer: Position Paper: Towards a Structured Approach to Developing Benchmarks for Virtual Network Functions. EWSDN, October 2016.

12 Leonhard Nobach, Ivica Rimac, Volker Hilt, David Hausheer: SliM: Enabling Efficient, Seamless NFV State Migration. IEEE ICNP, November 2016.

3.10 Road towards lightweight virtual network functions

Oliver Hohlfeld (RWTH Aachen, DE)

Main reference F. Schmidt, O. Hohlfeld, R. Glebke, K. Wehrle, "Santa: Faster Packet Delivery for Commonly Wished Replies", Computer Communication Review – SIGCOMM'15, 45(5):597–598, ACM, 2015.
URL http://dx.doi.org/10.1145/2785956.2790014

Current research efforts concern the creation, execution, or the placement of network function – mainly encapsulated in VMs or containers. While these virtualization efforts enable elastic scaling and cost efficient provisioning of virtual network functions, some functions require smaller and more lightweight execution environments. In this talk, I argue that there exists a case for realizing lightweight on-path network functions that do not need fully-fledged VM-based execution environments but rather run on network elements (e.g., switches). We have created a first prototype of an application agnostic execution environment to implement intelligent on-path functions in the network. By using this prototype, we show that server application performance can be optimized by utilizing these accelerated functions executed in the network core.

References

1 Florian Schmidt, Oliver Hohlfeld, René Glebke, Klaus Wehrle: Santa: Faster Packet Delivery for Commonly Wished Replies. Computer Communication Review 45(5):597–598, 2015.

2 Amir Mehmood, Oliver Hohlfeld, Dan Levin, Andreas Wundsam, Florin Ciucu, Fabian Schneider, Anja Feldmann, Ralf-Peter Braun: The RouterLab – Emulating Internet Characteristics in a Room. 11th ITG Conference on Photonic Networks, 2010.

3 Yvonne Coady, Oliver Hohlfeld, James Kempf, Rick McGeer, Stefan Schmid: Distributed cloud computing: Applications, status quo, and challenges. Computer Communication Review 45(2):38–43, 2015.

4 Marc Werner, Johannes Schwandke, Matthias Hollick, Oliver Hohlfeld Torsten Zimmermann, Klaus Wehrle STEAN: A storage and transformation engine for advanced networking context. IFIP Networking Conference, 2016

3.11 NFV for Industrial Control Networks

Jan Rüth (RWTH Aachen, DE)

The rising demand for automation and flexible manufacturing challenges the way today's Network Control Systems are structured. Current control architectures are composed of expensive and proprietary controllers that have been tailored for a specific use case. It is at the moment impossible to flexibly change the control algorithms and configurations. Therefore, the idea of a "controller in the cloud" is promising to give the flexibility and scaling properties that cloud computing has brought to networking. However, a controller in a cloud environment hosted over the Internet cannot meet the demands of current control algorithms, namely low latency and low jitter. We therefore propose to enable flexible lightweight-VNFs that can be computed on path within the switches of an industry control network that are under orchestration of the controller in a cloud environment. This idea challenges how packet processing within the switches can be flexibilized and how control algorithms may be expressed in this domain while maintaining high performance.

3.12 Seamless Elasticity in NFV Infrastructures with Heterogeneous Hardware Requirements

Leonhard Nobach (TU Darmstadt, DE)

Network Functions Virtualization (NFV) applies the principles known from *cloud computing* to network functions traditionally running on dedicated, inflexible and expensive hardware. In our work, we delve into two major topics:

First, we argue that the increased elasticity and fast provisioning properties of NFV come to the cost of performance issues on a single VNF instance, as these commonly operate on sequentially-working processors, unlike ASIC-driven high-performance appliances. However, the inclusion of hardware acceleration (HWA) into NFV infrastructures carries the danger of bringing back rigidness and inflexibility. We therefore propose a framework for elastic provisioning of *reconfigurable* hardware. VNF instances can dynamically claim and release FPGA (or NPU) resources over the network from a pool in the datacenter, if their performance requirements increase.

Secondly, the arising advantages for elasticity and dynamicity require state transfer mechanisms (for relocation or scale-in/out, x86 or HWA) that are *seamless* – the end user does not perceive any service disruption. Existing work does not consider the high costs of seamless state transfer traffic as a critical resource, however, links of network functions tend to be higher utilized than traditional server-cloud traffic, and state transfer attempts from or to *edge* or even *fog* instances aggravate this issue. We introduce the concept of *statelets* and our *SliM* state transfer mechanism, which can increase the possible link utilization up to a factor of three compared to previous approaches.

References

1 Leonhard Nobach, David Hausheer: Open Elastic Provisioning of Hardware Acceleration in NFV Environments. In: Workshop on Software-Defined Networking and Network Function Virtualization for Flexible Network Management (SDNFlex 2015), March 2015.

2 Leonhard Nobach, Ivica Rimac, Volker Hilt, David Hausheer: SliM: Enabling Efficient, Seamless NFV State Migration. In: IEEE International Conference on Network Protocols (ICNP), November 2016.

3.13 NFV Performance Profiling and Optimization

Andreas Kassler (Karlstad University, SE)

As network functions will be virtualized and may run on any given hardware the NFVI operator owns, we need proper methods that allow us to predict the NFV performance for a given infrastructure configuration. This requires a flexible profiling framework which allows to benchmark a given virtualized NF and configure the infrastructure in a flexible way. Having available KPIs from both inside virtual NF and the hypervisors, we would be able to model and predict the VNF performance. For optimization of VNF placement, we typically do not know precisely the input to the optimization problem. For example, we cannot accurately predict the amount of resources a given VNF needs to perform its functions. From optimization theory it is a well-known fact that once parameter are allowed to deviate from the nominal values, the optimal solution may become highly infeasible one. Consequently, we need to develop fast online solution algorithm that are able to cope with the uncertainty of information in a robust way.

3.14 Bring those network functions back to the plumbing!

Fernando M. V. Ramos (University of Lisbon, PT)

With Network Function Virtualization (NFV) the network functionality of dedicated hardware middleboxes is replaced by software running in VMs in commodity servers. In this talk we propose to bring (part of) the network functions back to the network (i.e., run the functions in network switches). This is made possible today by the recent advances in network data plane programmability (enabled by switch chip architectures such as RMT-PISA, for instance). Recently, high-performance switches (e.g., Barefoot Tofino) that can be programmed using high-level languages (e.g., P4) have emerged. The strengths of the proposed solution is mainly a gain in performance, in both lower latency (the NFs can be exactly where they're needed) and higher aggregate throughput (as NFs run in switches). The main challenges are the limited memory and computation resources of network switches. Interesting research questions in this space include how much of the network function can be offloaded to programmable hardware, the implications of the additional network programmability in terms of security, and how networks composed of stateful boxes that lead to "mutable data paths" (in which the handling of a packet depends not just on immutable forwarding state, but also on state that changes at packet time scales) can be verified, tested and debugged.

3.15 Network Function Virtualization in Software Defined Infrastructures

Michael Scharf (NOKIA – Stuttgart, DE)

Software Defined Networking (SDN) is an essential part of dynamic enterprise services, which include, amongst others, Network Functions Virtualization (NFV). Existing Carrier-SDN solutions provide network function management, analytics and assurance, as well as dynamic management and control. These Carrier-SDN solutions are already deployed in operational networks. Still, open research issues remain. Examples include the future design of data modeling languages (e.g., YANG, TOSCA) or verification of the interoperability in heterogeneous environments.

3.16 On Research Challenges of NFV

Christian Esteve Rothenberg (State University of Campinas, BR)

Critical issues to realize NFV in practice include proper performance evaluation methodologies towards predictable behavior and in support of optimized VNF placement. An open-source, collaborative approach is desirable to provide rich, reproducible platforms for VNF testing and benchmarking. Those issues are on our road-map and I hope the community will be able to effectively solve them.

3.17 NFV at the Edge

Tim Wood (George Washington University – Washington, DC, US)

NFV offers the opportunity to run high-performance network services in a flexible way. Edge clouds may become a new place to provide such services, potentially with very low latency access for users. The types of "users" connecting to such services also may change over time – not just users on phones or laptops, but autonomous vehicles, robots, smart city infrastructure, etc. These new types of users may require new types of NFV services, and may cause us to rethink the line between a network function and an application.

3.18 Utilizing Hardware Capabilities for Efficient NF Deployment

Fabian Schneider (NEC Laboratories Europe – Heidelberg, DE)

Current trends in SDN research, such as BFBA or P4, lead to extended possibilities to program network elements and NICs. Furthermore, hardware components such as GPUs or encryption cards allow for extended options to instantiate (virtual) network functions. The challenges stemming from this include, but are not limited to, (a) decision algorithms on how to split/delegate/offload sub-functions to hardware or in-network (b) programming abstraction that allows for such a functional split and (c) dynamic and online re-composition of NFs.

3.19 Dynamic Placement of Network Functions for Mitigating Internet-Scale Attacks

Oliver Michel (University of Colorado – Boulder, US)

Distributed Denial of Service (DDoS) attacks and other large-scale network attacks are common in today's Internet. They cost significant amounts of money each day by making information and services unavailable or forcing businesses to scale up their resources in the network or in the cloud while under attack. Over the past years however, mitigation techniques also become more advanced. Advanced Intrusion Detection and Prevention Systems (IDS/IPS) often are ASIC-based and can inspect large volumes of traffic at high rates. They use advanced application-layer analysis techniques, and leverage learning-based approaches to detect anomalies. We argue that wide-area SDN and NFV technologies can complement such advanced detection and filtering techniques. In particular, network functions can be placed on-demand during an attack to inspect and filter traffic at multiple locations throughout the network. These locations can be dynamically chosen to be closer to the attacker sources as opposed to placing all filtering at the victim's network gateway where the attack traffic volume can already be intractable to control. As we imagine such an architecture also in an inter-domain (AS) setting, several questions regarding a centralized trusted party and business models for cooperative, on-demand DDoS defense arise.

3.20 User-Centric NFV

Theophilus Benson (Duke University – Durham, US)

NFV has emerged as a panacea for a multitude of problems. Surprisingly, many solutions aim to provide homogeneous substrates of infrastructures. In this talk, I will argue for specializing the configuration and infrastructure to optimize the performance delivered to an end user. Although this talk focuses on performance, the broad concept of specialization can be used to improve other properties: security and reliability.

In this talk, I will present several configuration examples that impact the end user's performance. Motivated by these benefits, I will present a framework for optimizing and specializing NFV infrastructures: I will discuss inputs and constraints to the optimization problem. I will conclude by asking others to think of the potential of heterogeneity on emerging use-cases.

3.21 Flexibility as a Design Guideline for NFV and SDN

Wolfgang Kellerer (TU München, DE)

The following questions are still challenging in NFV research: How to cope with the emerging network dynamics? How to design a network for flexibility? How to migrate (network) functions? What is the role of virtualization and SDN? How can we use "flexibility" as a measure to compare different system designs?

4 Lightning Talks

4.1 An overview of network verification

Costin Raiciu (University Politehnica of Bucharest, RO) and Aaron Gember-Jacobson (Colgate University)

Networks are a critical part of our society, but managing them is tedious and error-prone. Configuration errors are common and bring major disruptions. NFV and SDN will further amplify the difficulty of ensuring networks are functioning correctly, i.e. they obey their operator policy.

The talk will cover efforts to verify formally that a network obeys the policy of its operator and has two parts focusing on the control plane verification and data plane verification. Data plane verification means taking a snapshot of the data plane state, converting it into a verifiable model; next the policy is checked against the model.

The talk will briefly cover HSA and NOD the best-known data plane verification tools, and details the design and implementation of Synet, our symbolic execution tool optimized for networks, and the associated SEFL language.

Control plane verification models the control plane, rather than the data plane, allowing policies to be verified across arbitrary failure scenarios. The talk will briefly cover Batfish, a recent control plane verifier, as well as discuss ARC, a control plane verifier that speeds up network verification by casting verification as a graph analysis problem. One limitation of control plane verifiers is their inability to capture subtle ties in the implementation of routing protocols on different vendors' devices.

4.2 Network Programmability: A Primer of Routing Synthesis

Laurent Vanbever (ETH Zürich, CH)

Between 50% and 80% of the network downtime are due to human, not equipment failing. Most of these are due to configuration mistakes, i.e. due to humans. In this talk, I present new directions in network management, one area akin to declarative programming. Specifically, I describe two complementary ways to declare and provision forwarding state, network-wide, of legacy equipment. One of the body of work is fibbing, a way to generate inputs to a distributed algorithm (Dijkstra) is that it computes what the operator wishes. In the second body of work, Synet, I explain how one can actually generate the distributed algorithm itself.

4.3 The NFV Standardization Concoction

Diego R. López (Telefonica I+D – Seville, ES)

The goal of the talk was to provide an introduction to the most relevant standardization efforts in the NFV arena, and to explore how they were related to current research challenges, and how the technology itself was shaping a new way to produce such standards.

We started with a brief introduction to the research challenges in NFV, from the perspective of a network service provider that has participated in the development of the NFV concepts since their inception. A list of technology and business challenges were introduced and briefly discussed.

The core of the talk was focused on the two main standards-focused groups currently involved in NFV matters: the ETSI NFV ISG and IRTF's NFVRG. The evolution, structure, and current plans of ETSI NFV where introduced, discussing the three types of deliverables the community aims at: normative and informative (as in many other standards bodies), and demonstrative (in the form of PoCs and early interoperability assessments)

The origin and goals of the NFVRG were introduced afterwards, detailing its research agenda, as an initial input to the discussion planned for the seminar. Later, the activities connected with NFV in other standards bodies were presented, and finally a discussion of relevant open-source projects and their influence in the standardization process was presented.

The talk concluded with a summary of how NFV is shaping standardization while it is standardized itself, and a call to the participants to bring their results to these ongoing efforts.

4.4 The Functions Placement Problem

Wolfgang Kellerer (TU München, DE)

Joint work of Wolfgang Kellerer, Arsany Basta, Andreas Blenk, Marco Hoffmann, Hajo Morper, Klaus Hoffmann

Network Function Virtualization (NFV) together with Software Defined Networking (SDN) opens up new challenges for the composition, placement and migration of network function in an operator's network. We refer to this class of problems as the Function Placement Problem (FPP) inspired from the Controller Placement Problem that has been introduced for SDN controllers by Heller in HOTSDN 2012. However, we see new types of challenges arising with the FPP. First, as part of an optimal placement an optimal function (de-)composition and chaining has to be considered. SDN and NFV offer complementing concepts here where network functions can be moved completely (based on NFV) or partially (based on the SDN control/data plane split) into a data center [1]. Second, we address dynamic placement and migration as a further important network design aspect [2]. Finally, we show how to analyze the flexibility a chosen network design to understand its overall benefits [3].

References

1 A. Basta, W. Kellerer, M. Hoffmann, H. Morper, K. Hoffmann. *Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem*. AllThingsCellular14, Workshop ACM SICGOMM, Chicago, IL, USA, August 2014.

2 A. Basta, A. Blenk, M. Hoffmann, H. Morper, K. Hoffmann, W. Kellerer. *SDN and NFV Dynamic Operation of LTE EPC Gateways for Time-varying Traffic Patterns*. 6th International Conference on Mobile Networks and Management (MONAMI), Würzburg, Germany, September 2014.

3 W. Kellerer, A. Basta, A. Blenk. *Using a Flexibility Measure for Network Design Space Analysis of SDN and NFV*. Software-Driven Flexible and Agile Networking (SWFAN), IEEE INFOCOM Workshop, San Francisco, USA, April 2016.

4 W. Kellerer, A. Basta, A. Blenk. *Flexibility of Networks: a new measure for network design space analysis?* arXiv report, December 2015[1].

4.5 Getting out of the NFV Deployment Quagmire

Felipe Huici (NEC Laboratories Europe – Heidelberg, DE)

Containers are in great demand because they are very lightweight when compared to virtual machines: both boot times and memory usage are significantly smaller than traditional VMs, and this allows massive consolidation of workloads on the same hardware. On the downside, containers have fundamentally weaker isolation properties than VMs.

In this talk, we examine whether there is indeed a strict tradeoff between isolation (VMs) and efficiency (containers). By redesigning the control plane of Xen and using small, optimized unikernel based virtual machines we show that it is possible to achieve VM boot times on the order of milliseconds while packing thousands of VMs on modest hardware.

[1] http://www.lkn.ei.tum.de/forschung/publikationen/dateien/Kellerer2015FlexibilityofNetworks:a.pdf

5 Working Groups

5.1 NFV Security, Validation, and Verification

Aaron Gember-Jacobson (Colgate University – Hamilton, US)

Joint work of Aaron Gember-Jacobson, Gabriela Gheorghe, James Kempf, Diego R.López, Bruce MacDowell Maggs, Costin Raiciu

- Service level agreements drive operator's efforts to verify/attest correctness, performance, etc.
- How does verification change as you go from hardware to software?
  - NFV is less trustworthy:
    - With hardware boxes, the development and deployment cycle is long: The vendor takes time to develop an NF and the operator takes time to certify the NF.
    - NFV tends to follow a DevOps model: less testing → higher likelihood of bugs → verification becomes more important. To benefit from NFV, you need to change the corporate culture within the ISP regarding the need for extensive certification and having individual boxes fulfill 99,999% (5-nines) uptime.
  - Verification of chaining:
    - With hardware, you simply trace the cables.
    - With software, it is much more difficult – you now have the chance to deploy very complex chains of small VNFs that were not deployable before, which makes the big picture more complex.
- How do we go about doing verification?
  - Lots of languages/tools for expressing/checking policies – not clear these are used.
  - P4 is valuable as a modeling language.
  - Verification could be made simpler by leveraging flexibility offered by NFV, for example...
    - . . . always put NFs in a particular order that is easier to verify
    - . . . break NFs into micro-services – introduces interoperability challenges; could also make verification harder, as noted above.
  - If you optimize for performance, then verification is hard, and vice versa.
- Operators want to be able to provide micro-containerized services that can be flexibly moved around.
  - Infrastructure needs to know if one can trust the hardware and software needs to know if one can trust the infrastructure.
  - Who vouches for the authenticity of an NF? How do you do this in real time? – we know how to do attestation, but doing it at scale is hard.
  - How do we apply attestation mechanisms for hypervisors to SDN controllers and switches?
  - How do you protect private keys stored on NFs that split TCP connections? – Can't put keys on every edge NF.

5.2 Placement

Wolfgang Kellerer (TU München, DE)

Joint work of Andreas Kassler, Wolfgang Kellerer, Diego Lopez, Michael Scharf, Fabian Schneider, Andreas Wundsam, Thomas Zinner

- Scope: Offline vs. online; the focus is online!
  1. Ways to decide to program and deploy depending on hardware capabilities, for example DPDK or P4. → Hardware offloading/transition? Heterogeneous hardware (adaptive hardware)?
     - Decomposition of VNFs:
       - Manual/Automatic,
       - Optimization,
       - Structural – designing the system,
       - Dynamic – reacting to environment/traffic.
  2. Machine Learning (control loop)
     - Actions:
       - Resource allocation, e.g. VM migration,
       - scheduling; single server (early papers exist).
     - Metrics/objective functions (different ones).
       - What metrics? – traditional (hardware), software, service level (KPI).
     - Monitoring / telemetry
  3. Heuristics for VNF placements (e.g., co-location)
     - Online and offline placement is mostly a planning process.
     - There could be placement (optimization) or assignment problems, or also other interconnected problems (e.g. from switch to controller).
  4. Dynamics – Changing the system
     - What is the cost of migration (resource, time)?
     - Verification of continuity of operation
     - Debugging / root cause analysis
     - Robustness, e.g. during state migration, uncertainty of prediction, impact?
- Open Questions / Discussion
  - Machine Learning – how good is it?
    - Where does the data/training come from? The assumption is that data is available.
    - Machine Learning <-> Heuristics / Optimizers
    - Both may may be combined, e.g. machine for pre-filtering and then heuristics or optimization of a simpler problem.
  - Reaction to unknown situations (e.g., attacks!)
  - Models and abstractions for performance
    - How realistic is a "world model"?
      - Where to place, when to place?
      - What are the parameters that we need to optimize for?
      - How is the real hardware working?
    - SLA metrics for NFV (under standardization)? KPI? Abstractions?
      - *Remark from Diego Lopez:* There is some ongoing work in ETSI on performance characterization and maybe some early work on KPI. There is nothing on SLA so far.
  - Conflicting goals → how to sort this out?

    - User: Best performance (SLA-defined)
    - Operator: “Run infrastructure hot”
- Discussion:
  - *James Kempf:* What algorithms would you use? Traditional optimization algorithms are not designed for online use.
  - *Diego Lopez:* Call to practitioners for data-driven management [seminar conclusion].
  - *Thomas Zinner:* Machine learning is not good for systems under change since the algorithms are not trained for the new conditions.
  - *Andreas Kassler:* How would one compare different heuristics? There is no standard set of agreed-upon topologies, workload, . . . for the evaluation. We need something like a standardized VNF evaluation database.
  - Suggestion: This may be a topic for another Dagstuhl seminar or a project proposal.
  - *Diego Lopez:* To have predictable performance, you need plenty of knowledge about the underlying hardware/platform (ongoing work with Intel). Open-source MANO: osm.etsi.org.
  - Security properties of the network need to be reflected, not only topology.
  - *Tim Wood:* What is different about NFV placement as compared to other placement problems?
  - Network topology has more impact on how things are run.
  - NFV, by definition, has high I/O.
  - In the end it is an optimization problem, just more complex.

5.3 NFV Economics

Leonhard Nobach (TU Darmstadt, DE)

Joint work of Tim Wood, David Hausheer, Jan Rüth, Leonhard Nobach, Oliver Hohlfeld, Theophilus Benson

- ISPs provide services to over-the-top (OTT) providers to deploy services.
  - Why would they do that? Money, Network offloading.
  - Follow-me-Cloud: Which applications exist? High computing power required with low latency.
  - However, much latency added at the edge (e.g. forward error correction).
  - Privacy issues
    - Locality → local laws
    - Power over infrastructure
  - Do we need more than sandboxing via SDN?
  - Are there mechanisms to guarantee that the provider cannot e.g. snoop the memory of my VNF?
  - Asymmetric bandwidth: The reason is DDoS. Edge clouds would make it worse.
  - DDoS Detection would be a good use case: An NF may search for signs for DDoS.
  - Suggestion: An NF which returns aggregates of customer data only (data protection).
  - New economics:
    - The ISP provides feature for OTTs to install VNFs, It then charges the customer.
    - Customer requests OTT service, OTT installs VNF to deliver the better service to customer.
    - Aggregation of multiple customers possible.

    - ∗ Small and medium enterprises.
  - What is the cost/revenue share between the ISP and OTT provider?
  - Regulation: Is this net neutrality?
    - ∗ If every OTT can deploy VNFs: No need for regulation.
  - Use case for low latency: Tactile gaming.
- Discussion:
  - Objections from *Diego Lopez*
  - Idea: Tactile e-commerce (James)
    - ∗ *Bruce Maggs:* E-Commerce is the one place today where people pay a lot of money for low latency. How can you use NFV to realize that?
  - *Dirk Kutscher:* There are only two use cases for edge computing.
    - ∗ Lots of data is generated locally.
    - ∗ Interactive VR
    - ∗ For all other use cases, one is much better off reducing the overall latency.
  - *Tim Wood:* The pricing of these resources is going to be complicated
  - *Bruce Maggs:* This sort of net neutrality doesn't exist yet. Operators currently charge whatever they want
  - *Michael Scharf:* For a VNF to be useful you need 1) a VNF and 2) connectivity. This can provide some interesting economics related to location: 1) infrastructure is idle (and thus cheap) but connectivity is overloaded. 2) infrastructure is hot, but connectivity has idle resources.
  - *Michael Scharf:* The other interesting implication is the last-mile connectivity is typically the expensive part.

5.4 NFV Performance

Theophilus Benson (Duke University)

Joint work of Theophilus Benson, Christian Esteve Rothenberg, David Hausheer, Oliver Hohlfeld, Leonhard Nobach

- Fundamental questions:
  - What is a VNF?
  - How can we systematically model and measure the performance characteristics of a VNF?
  - How can we make (or develop) techniques and frameworks that enable a software (VNF) to provide similar performance guarantees as hardware-based functions?
- Understanding and framing the performance problem:
  - Types of metrics: App-specific (SIP rps, HTTP rps), user-specific (page load time), or general VNF metrics.
  - General VNF Metrics: first class (latency/throughput), second class (migration/boot-up time).
- Use cases for VNF:
  - What is a VNF: a middlebox running in a VM or a middlebox rewritten for "the cloud"?
  - Control functions – DNS, DHCP, BGP aggregation.
  - Data functions – focus on network functions that cannot be supported in ASICs/P4.

    - A stateful load balancer which requires HTTP header information,
    - A dynamic deep packet inspector (DPI): flexible regular expression or signature matching not supported,
    - Software-defined WAN: WAN optimization not supported in ASIC,
    - Billing/Accounting: cannot be scalable done in ASIC,
    - Fine-grained and specialized code: ASIC doesn't allow sufficient programmability for arbitrary code,
    - Interface and API for deployment: Automated attestation and provisioning.
- Measure / Benchmark VNFs
  - Dimensions to analyze and compare
    - Describe and quantify input of: SFC, operating system and hardware stack, Versioning, collateral workload, etc. etc.
    - Versioning: code release, configuration change.
    - Hardware specifications: x86 vs. FPGA vs GPU with performance guarantees (consistent / constant performance behavior).
    - Guarantees under normal and adversarial workloads: e.g. physically co-located VNFs competing for I/O or trashing memory caches.
    - Cost vs. performance trade-off: Cost performance characteristics or guarantees of a VNF.
  - Who uses the benchmarks? How much control does the benchmark have?
    - An entity using a third-party cloud/cloudlet?
    - An entity that controls code and infrastructure?
  - Open / General framework for testing (functional, regression, performance) during development. See Fd.io approach.
  - Linked-in versus Telco data centers.
  - Containers vs. VM: how do you reason about the guarantees.
  - Existing infrastructure support will impact how we design and decompose functions.
  - Definition of a VNF: Composed of several (one or more) virtualization units connected by an infrastructure network.
    - Think more broadly about not just a unit but an orchestration of VNFs.
    - SDN in the VNF and outside of the VNF.
    - VM in the data path – there are complex (legacy) constraints between the VM.
    - VM in the control space – multiple VMs for redundancy.
    - ETSI has the notion of VNFC which are components of the VNF – the potential unit.
- Ongoing / Related Work:
  - Gym [*Christian Rothenberg* & Ericsson Research][2]
  - vnfbench[3]
  - VBaaS[4]
  - Towards a Structured Approach to Developing Benchmarks for Virtual Network Functions.[5]
  - *Andreas Kassler*'s recent project (see Lightning talks).

[2] http://materials.dagstuhl.de/files/17/17032/17032.ChristianEsteve%20Rothenberg1.Slides.pdf

[3] https://tools.ietf.org/html/draft-rosa-bmwg-vnfbench-00

[4] https://datatracker.ietf.org/doc/draft-rorosz-nfvrg-vbaas/

[5] Jeremias Blendin, Yuriy Babenko, Dennis Kusidlo, Georg Schyguda and David Hausheer. In EWSDN 2016.

- Discussion:
  - *Dirk Kutscher:* The evolution of non-telco datacenters might be very relevant to the function placement / composition problem, e.g., the way that LinkedIn is running $change_me?
  - *Diego Lopez:* The VNF does not necessarily equal to a virtualization unit, e.g. a VNF can be composed out of more than one VM.
- To do:
  - Definition of a VNF highly unclear → a definition should be added to the report → discuss with *Diego Lopez*.
  - The VNF can contain forwarding, as well.
  - In ETSI, the term VNFC is used (c for component).
  - *Fabian Schneider:* It would be good to outline deployment options for VNF.
  - Container-based
  - In switches

5.5 NFV Use Cases

- Operator view
  - NFs in operator networks
  - Transparent middleboxes (e.g., proxy, application acceleration) are going to die – because all traffic is encrypted.
  - At the wired edge as a DSLAM and CGNAT.
  - In mobile networks there are eNodeBs, S-GWs (end of encrypted tunnels), and P-GWs.
  - Control boxes: accounting, radio service, monitoring.
  - We want to move all of these to software.
  - Vision of moving virtualized functions to the eNodeB and moving them to another eNodeB when the user moves.
  - NFV challenges: security, placement (includes hardware/software split), testing/visibility/verification, performance, economics?
  - We want to combine wired and mobile networks into one network.
- End users (and apps?) view
  - We want to insert VNF along the path.
  - This is being done a little by allowing insertion of, for example, virtual firewalls in the mobile edge cloud – only tenants with a business relationship with the provider are allowed to put VMs in the mobile edge cloud; mobile edge computing is like a CDN edge.
  - Not really cloud computing – call it "middleboxes-as-a-service".
  - Service providers have same perspective.

6 Wrap-up

6.1 Research Areas and Questions

At the end of the seminar, we asked each seminar participant to identify research areas and open research questions on NFV and anonymously provide them on paper cards. We state the returned questions below.

- Impact of a microservice architecture on NFV performance, flexibility, resilience, etc.
- Metrics-Analytics-Policing-Control Loops: How much of network management can be automated and how can possible fatal clashes between control loop actions (e.g. one loop changes routing in one direction of another in the opposite direction) be identified and mitigated?
- Distributed Cloud, Edge Cloud: Does this have any impact on NFV deployment?
- How can HW capabilities (e.g. programmable hardware) be exposed by the hypervisor and used for NFV (placement)?
- How can a provider run untrusted NFs safely? How can a customer run NFs on an untrusted provider safely?
- How can an NFV platform provide real-time performance guarantees?
- How do you estimate the performance of a VNF in an early stage of the development?
- How to characterize/predict the performance of complex/dynamic systems with many VNFs.
- Placement optimization that considers the peculiar properties (e.g. the number of lookups, cache behavior). Optimized schedulers for this environment.
- How to include FPGA resources into NFV infrastructures while keeping elasticity and flexibility properties known from COTS processors?
- Networking vs. Commodity: What area should we match to?
- What sorts of applications would external parties (e.g. CDNs) like to deploy as network functions in carrier networks? One example might be a distributed denial of service mitigation function.
- How could carriers safely allow the deployment of such functions?
- What functionality could NFV provide that can't be easily provided any other way?
- Does it make sense to unify SDN with some sort of NFV control?
- Applicability of Microservice Architectures to Telco NFV.
- Better concepts for edge computing.
- Usable and Actionable SLAs for NFV.
- NFV introducing exponentially-growing heterogeneity may require to rethink current approaches to Data and Information Modelling.
- Methods to measure and guarantee consistent performance, including multi-dimensional KPIs.
- Computer-assisted de-composition & re-composition of network functions: Programming Language, Instantiation Optimization.
- How to decompose big, monolithic NFs into smaller microservices? Can we do this automatically? Creation of use cases beyond current ASIC/HW-centric NFs to drive NFV (open problem rather than specific question).
- How to enable inter-AS VNFs that span over administrative boundaries?
- Definition of metrics and performance evaluation principles for VNFs.
- How can you quantify the user-level impact of employing NFV? Performance impact? Privacy? Availability?
- How can you reason about the implication of configuration changes in a principled manner?
- How do we convert home network gateway devices, laptops, cell phones into infrastructures (Edge cloud) for supporting user / application-specific NFV?
- How do we perform proactive data plane verification in a network that uses traditional routing protocols?
- How do we build higher-fidelity – yet tractably analyzable – models of NFs for verification?

- How do we speed up the NF certification process performed by operators?
- How to unify a network containing thousands of stateful data plane nodes?
- Which new security threats arise with NFV?
- What can be offloaded to (programmable) hardware?
- Automatic leveraging of hardware and in-network capabilities for VNFs.
- YANG, TOSCA, ______ ? What comes next?[6]
- How to bring performance benchmarking results and placement/optimization together without getting killed by complexity; where is the right abstraction?
- How to design appropriate mechanisms to "check" decisions/output of machine learning models; how to do a quality assessment of the machine learning results on short time scales?
- What fundamental design differences in NFV architectures we would envisage if we assume the Datacenter network has 500 terabytes per second of bi-sectional bandwidth within a single datacenter (Google Prediction)?
- Making SDN and NFV convergence happen.
- A general theory (and practice!) of software-based networking.
- Data-driven network management.
- Security in all aspects.
- New business model
- Application of Machine Learning to function placement
- Techno-economic analysis
- Placement w.r.t. dynamics, functional decomposition
- Flexibility as a metric to analyze designs
- Verification

6.2 Lessons Learned, Surprises

We anonymously asked the seminar participants about their lessons learned during the seminar and state them below.

- The extent of the TUM work on EPC decomposition virtualization. They seem to have covered most of the important points.
- Significant disconnect between academia and industry.
- Control may or may not be different from management.
- How messy and complex the services are inside telco providers.
- There is no clear distinction in NFV: Middlebox vs. edge-cloud vs. 5G core virtualization.
- There is no clear consensus on the definition of a VNF.
- The distinction between conceptual and implementation aspects is fuzzy (e.g. VM/Container).
- That there is little excitement and novelty. Unfortunately.
- Pessimism about the impact that OpenFlow will have in the future (You can't buy OpenFlow switches)
- NFV isn't really about virtualization, it is about implementing network functionality in software – in a portable way.

[6] https://www.ietf.org/mail-archive/web/teas/current/msg01900.html

- SDN isn't about software, it is about separating the control plane from the dataplane and perhaps centralizing the control plane.
- Lack of common views and amount of disagreement.
- The amount of faith in Machine Learning / Artificial Intelligence to solve "untrackable" challenges in NFV/SDN.
- The definition of a VNF and NFV use cases is still a highly controversial topic that has not settled yet.
- NFV still is telco-centric: needs of service providers other than telcos do not seem to play a major role currently.
- The role of NFV in 5G networks.
- The plethora of open questions in NFV placement.
- There are research questions in NFV.
- Most researchers do not read architectural documents developed by SDOs in their domain. This leads to unnecessary discussions on terminology and common understanding of principles and concepts. *Point from the discussion: A literature research is not complete if you only consider academic papers.*
- Recent progresses in (control plane) verification.
- Foglets maybe make it to real deployments (just making fun...).
- American people do not say "allow to".
- Many people say there are no research challenges for NFV: No!!! When it comes to operational questions etc.
- "Intent" is not the powerful buzzword, I thought it was.
- Good old control plane protocols have still a lot to squeeze out when it comes to verification, control, management...
- People mistake OpenFlow to be equal to SDN!
- The control vs. management question is still not solved in industry.
- Confirmation of: SDN is not OpenFlow, which is dead.
- The NFV ecosystem (users, operators, vendors) needs further consideration.
- ETSI NFV standardization status (still fuzzy)

6.3 Controversial statements about NFV

In this session, we asked the 20 participants to write down controversial statements about NFV anonymously on blank cards. We collected them afterwards, read them aloud, and asked all participants to raise hands if they agree on the statement (👍) or if they disagree on it (👎).

"NFV is an attempt to re-invent Intelligent Networks."
👍 8 👎 8

"Will the network management overhead kill network virtualization and NFV?"
👍 0 👎 21

"SDN will never make it to the CORE of the Internet backbone"
👍 3 👎 18

"NFV will never make it to the CORE of the Internet backbone"
👍 13 👎 8

"There is no technology which is generic enough to solve different use-cases. We will end up in use-case specific solutions! Even P4 will (probably) be used to the interconnection DL/WAN. SDN is "only" used for TE in WAN (or what might be something like SDN)."
👍 1 👎 20

"SDN and NFV is both all about network management and not about control! (And network management is not well researched)"
👍 3 👎 12

"There is no research challenge in NFV"
👍 0 👎 21

"Data-driven (ML)-based network management will render networks infeasible to debug"
👍 7 👎 10

"Because SW is slower than optimized hardware, we need to throw much more hardware at the problem to meet current performance criteria."
👍 11 👎 6

"Like SDN, in practice, NFV apps will run only on end hosts, and not on network hardware/appliances deployed in networks"
👍 0 👎 21

"NFV will enable more smaller-size companies to enter NF markets (e.g. providing CDNs)"
👍 15 👎 5

“We (terribly) lack good software engineering principles in the application of SDN/NFV”

👍 15 👎 3

“Regulation will anyway restrict SDN & NFV to the enterprise networks”

👍 0 👎 21

“There has been no fundamental and deployed innovation in network management for the last”

👍 0 👎 0

“SDN/NFV as a research area is slowly but surely dying”

👍 5 👎 13

“NFV will increase costs due to Software maintenance and make misconfiguration even easier than today”

👍 8 👎 11

“Heterogeneity of deployment options will result in unmanageable systems”

👍 5 👎 15

“OpenFlow is no longer required!”

👍 13 👎 7

“NFV won’t significantly lower costs for operators – they will need to structurally reform or will die”

👍 13 👎 5

“NFV research is mostly dead”

👍 0 👎 0

“NFV will never see widespread real-world deployment”

👍 0 👎 21

“There will only be at most 5 different useful network functions (i.e. there is no need for an NFV app store”

👍 13 👎 6

“Edge clouds will never provide a useful service normal people care about”

👍 1 👎 20

“Only NFs without cross-flow state can be dynamically scaled, because state sharing is prohibitively expensive”

👍 5 👎 12

“SDN will make it, but not with OpenFlow (how we know it)”

👍 18 👎 2

"NFV is going to converge with SDN"
👍 9 👎 7

"High-throughput, ASIC-based network appliances are not going to be replaced by NFV."
👍 17 👎 2

"NFV will have its first widespread deployment in the 5G RAN and mobile core"
👍 12 👎 1

"OpenFlow will remain a niche protocol and will never achieve its initial promise to replace existing protocols due to the lack of suitable hardware available in a timely fashion."
👍 21 👎 0

"Constructing real networks based on NFV will require a large amount of system integration (i.e. glue code), and until that is sorted through, networks based on NFV will be more expensive than traditional networks, but afterwards will be much cheaper."
👍 14 👎 1

"The impact of open source on standardization will be disruptive and there will be less SDOs in the future."
👍 9 👎 3

"NFV will lower the innovation, because vendors cannot sell boxes anymore, so there is no incentive to develop novel functions."
👍 0 👎 21

"NFV allows to speed up NF performance."
👍 10 👎 6

"There is only one way to meet the required NF performance: Put the NF in the switch!"
👍 1 👎 20

6.4 Wrap-up Notes

Author: Leonhard Nobach

- *Orchestration* has been identified as the current "hype buzzword".
- When NFV is solved, what will be the next hype? For example how to automate network management? How to avoid feedback loops (critical)? How to maximize automation? How to bring data-driven network management to the masses? Still, no buzzword exists for these questions.

- Other problems to solve are *intent management* (on-going work in the ONF[7]), and occurring conflicts.
- There might be a lack of solving “new” problems with NFV, i.e. a lack of innovation.

A very important outcome was that there is still no consensus on a very simple question: the definition of a network function (NF). Existing definitions have been discussed in a recent survey [1]: An NF can be considered as a logical network building block doing tasks which go beyond (SDN) forwarding. Taking such a definition, Layer 3 forwarding (involving more complex decisions, exchanging data link addresses and decrementing time-to-live counters) can be considered as a grey area, while any *stateful* processing (stateful firewalls, dynamic NAT) could be clearly considered as an NF. This does not exclude the possibility that parts of these NFs are implemented on an SDN data plane (for example the NF implementation dynamically adding or removing flows for established NAT or firewall sessions via OpenFlow).

According to this definition, a (V)NF can be implemented in one or multiple VMs (VNF instances), in the data plane itself, on reconfigurable, hardware–accelerated devices (e.g. FPGAs), or in a distributed system comprising a combination of these components.

References

1 Leonhard Nobach, Oliver Hohlfeld, David Hausheer: New Kid on the Block: Network Functions Virtualization: From Big Boxes to Carrier Clouds (Editorial). In: ACM Computer Communications Review (CCR), July 2016.

6.5 What comes after NFV?

Author: Oliver Hohlfeld

- Lots of pessimism in the research questions.
- Currently listed topics are more on the management side.
- Promise of SDN/NFV was to enable in the network. Where is this happening? *Comment James Kempf:* SDWAN.
- Don’t apply verification to OpenFlow control plane aspects. Stop with OpenFlow. The important questions are in the management plane.
  - Verification for intent: ONF TIPI
  - Northbound interface of a controller.
  - Published four months ago, not yet accessible to research.
- *Remark Fabian Schneider:* Move the abstractions up the stack. We need abstractions going above OpenFlow and come with verification properties.
- *Comment Michael Scharf:* Scripting etc. needed way up in the management plane.
- To-do point: A literature research is not complete if you *only* consider academic papers.
- Microservice ecosystem (a.k.a. Active Networks)
- HW/SW Codesign (for flexibility)

[7] https://www.opennetworking.org/?p=1633&option=com_wordpress&Itemid=155

Participants

- Theo Benson
Duke University – Durham, US
- Christian Esteve Rothenberg
State University of Campinas, BR
- Aaron Gember-Jacobson
Colgate University – Hamilton, US
- Gabriela Gheorghe
PwC – Luxembourg, LU
- David Hausheer
TU Darmstadt, DE
- Oliver Hohlfeld
RWTH Aachen, DE
- Felipe Huici
NEC Laboratories Europe – Heidelberg, DE
- Andreas Kassler
Karlstad University, SE
- Wolfgang Kellerer
TU München, DE
- James Kempf
Ericsson – Santa Clara, US
- Dirk Kutscher
Huawei Technologies – München, DE
- Diego R. Lopez
Telefonica I+D – Seville, ES
- Bruce MacDowell Maggs
Duke University – Durham, US
- Oliver Michel
University of Colorado – Boulder, US
- Leonhard Nobach
TU Darmstadt, DE
- Costin Raiciu
University Politehnica of Bucharest, RO
- Fernando M. V. Ramos
University of Lisbon, PT
- Jan Rüth
RWTH Aachen, DE
- Michael Scharf
NOKIA – Stuttgart, DE
- Fabian Schneider
NEC Laboratories Europe – Heidelberg, DE
- Laurent Vanbever
ETH Zürich, CH
- Timothy Wood
George Washington University – Washington, DC, US
- Andreas Wundsam
Big Switch Networks – Santa Clara, US
- Thomas Zinner
Universität Würzburg, DE

Report from Dagstuhl Seminar 17041

Randomization in Parameterized Complexity

Edited by

Marek Cygan[1], Fedor V. Fomin[2], Danny Hermelin[3], and Magnus Wahlström[4]

1 **University of Warsaw, PL,** cygan@mimuw.edu.pl
2 **University of Bergen, NO,** fomin@ii.uib.no
3 **Ben Gurion University – Beer Sheva, IL,** hermelin@bgu.ac.il
4 **Royal Holloway University of London, GB,** magnus.wahlstrom@rhul.ac.uk

Abstract

Dagstuhl Seminar 17041 "Randomization in Parameterized Complexity" took place from January 22nd to January 27th 2017 with the objective to bridge the gap between randomization and parameterized complexity theory. This report documents the talks held during the seminar as well as the open questions arised in the discussion sessions.

Seminar January 22–27, 2017 – http://www.dagstuhl.de/17041
1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory
Keywords and phrases fixed-parameter tractability, intractability, parameterized complexity, randomness
Digital Object Identifier 10.4230/DagRep.7.1.103
Edited in cooperation with Marc Roth

1 Executive Summary

Marek Cygan
Fedor V. Fomin
Danny Hermelin
Magnus Wahlström

Randomization plays a prominent role in many subfields of theoretical computer science. Typically, this role is twofold: On the one hand, randomized methods can be used to solve essentially classical problems easier or more efficiently. In many cases, this allows for simpler, faster, and more appealing solutions for problems that have rather elaborate deterministic algorithms; in other cases, randomization provides the only known way to cope with the problem (e.g. polynomial identity testing, or deciding whether there exists a perfect matching with exactly b red edges in an edge-colored bipartite graph). On the other hand, there are also cases where randomness is intrinsic to the question being asked, such as the study of properties of random objects, and the search for algorithms which are efficient on average for various input distributions.

Parameterized complexity is an approach of handling computational intractability, where the main idea is to analyze the complexity of problems in finer detail by considering additional problem parameters beyond the input size. This area has enjoyed much success in recent years, and has yielded several new algorithmic approaches that help us tackle computationally

Randomization in Parameterized Complexity, *Dagstuhl Reports*, Vol. 7, Issue 1, pp. 103–128
Editors: Marek Cygan, Fedor V. Fomin, Danny Hermelin, and Magnus Wahlström
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

challenging problems. While randomization already has an important role in parameterized complexity, for instance in techniques such as color-coding or randomized contractions, there is a common opinion within researchers of the field that the full potential of randomization has yet to be fully tapped.

The goal of this seminar was to help bridge this gap, by bringing together experts in the areas of randomized algorithms and parameterized complexity. In doing so, we hope to:

- Establish domains for simpler and/or more efficient FPT algorithms via randomization.
- Identify problems which intrinsically need randomization.
- Study parameterized problems whose instances are generated by some underlying distribution.
- Stimulate the development of a broadened role of randomness within parameterized complexity.

2 Table of Contents

Open problems

3 Overview of Talks

3.1 Hardness in P

Amir Abboud (Stanford University, US)

The class P attempts to capture the efficiently solvable computational tasks. It is full of practically relevant problems, with varied and fascinating combinatorial structure.

In this talk, I will give an overview of a rapidly growing body of work that seeks a better understanding of the structure within P. Inspired by NP-hardness, the main tool in this approach are combinatorial reductions. Combining these reductions with a small set of plausible conjectures, we obtain tight lower bounds on the time complexity of many of the most important problems in P.

3.2 Towards Hardness of Approximation for Polynomial Time Problems

Arturs Backurs (MIT – Cambridge, US)

Proving hardness of approximation is a major challenge in the field of fine-grained complexity and conditional lower bounds in P. How well can the Longest Common Subsequence (LCS) or the Edit Distance be approximated by an algorithm that runs in near-linear time? In this paper, we make progress towards answering these questions. We introduce a framework that exhibits barriers for truly subquadratic and deterministic algorithms with good approximation guarantees. Our framework highlights a novel connection between deterministic approximation algorithms for natural problems in P and circuit lower bounds.

In particular, we discover a curious connection of the following form: if there exists a $\delta > 0$ such that for all $\epsilon > 0$ there is a deterministic $(1+\epsilon)$-approximation algorithm for LCS on two sequences of length n over an alphabet of size $n^{o(1)}$ that runs in $O(n^{2-\delta})$ time, then a certain plausible hypothesis is refuted, and the class E^{NP} does not have non-uniform linear size Valiant Series-Parallel circuits. Thus, designing a “truly subquadratic PTAS” for LCS is as hard as resolving an old open question in complexity theory.

3.3 Directed Hamiltonicity parameterized by the largest independent set

Andreas Björklund (Lund University, SE)

Joint work of Andreas Björklund, Petteri Kaski, Ioannis Koutis

We present a Monte Carlo algorithm deciding Hamiltonicity in n-vertex directed graphs in $O^*(3^{n-\mathsf{mis}(G)})$ time and polynomial space, where $\mathsf{mis}(G)$ is the size of the largest independent set in the graph. In particular, in bipartite graphs we get a $O^*(1.733^n)$ time and polynomial space algorithm improving over the $O^*(1.888^n)$ time and exponential space algorithm by Cygan et al. from STOC 2013.

3.4 Fine-grained dichotomies for the Tutte plane and Boolean #CSP

Cornelius Brand (Universität des Saarlandes, DE)

Joint work of Cornelius Brand, Holger Dell, Marc Roth

Jaeger, Vertigan, and Welsh proved a dichotomy for the complexity of evaluating the Tutte polynomial at fixed points: The evaluation is #P-hard almost everywhere, and the remaining points admit polynomial-time algorithms. Dell, Husfeldt, and Wahlén and Husfeldt and Taslaman, in combination with Curticapean, extended the #P-hardness results to tight lower bounds under the counting exponential time hypothesis #ETH, with the exception of the line $y = 1$, which was left open. We complete the dichotomy theorem for the Tutte polynomial under #ETH by proving that the number of all acyclic subgraphs of a given n-vertex graph cannot be determined in time $\mathsf{exp}(o(n))$ unless #ETH fails. Another dichotomy theorem we strengthen is the one of Creignou and Hermann for counting the number of satisfying assignments to a constraint satisfaction problem instance over the Boolean domain. We prove that all #P-hard cases are also hard under #ETH. The main ingredient is to prove that the number of independent sets in bipartite graphs with n vertices cannot be computed in time $\mathsf{exp}(o(n))$ unless #ETH fails. In order to prove our results, we use the block interpolation idea by Curticapean and transfer it to systems of linear equations that might not directly correspond to interpolation.

3.5 A Near-Linear Pseudopolynomial Time Algorithm for Subset Sum

Karl Bringmann (MPI für Informatik – Saarbrücken, DE)

Main reference K. Bringmann, "A Near-Linear Pseudopolynomial Time Algorithm for Subset Sum", in Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017), pp. 1073–1084, SIAM, 2017.
URL http://dx.doi.org/10.1137/1.9781611974782.69

Given a set Z of n positive integers and a target value t, the SubsetSum problem asks whether any subset of Z sums to t. A textbook pseudopolynomial time algorithm by Bellman from 1957 solves SubsetSum in time $O(nt)$. Here we present a simple randomized algorithm running in time $\tilde{O}(n+t)$. This improves upon a classic result and is likely to be near-optimal, since it matches conditional lower bounds from SetCover and k-Clique. One of our main tools originated in the field of parameterized algorithms. We also present a new algorithm with pseudopolynomial time and polynomial space.

3.6 Relatively recent insights into counting small patterns

Radu Curticapean (Hungarian Academy of Sciences – Budapest, HU)

Joint work of Radu Curticapean, Holger Dell, Dániel Marx

We consider the problem of counting subgraphs. More specifically, we look at the following problems #$\mathsf{Sub}(C)$ for fixed graph classes C: Given as input a graph H from C (the pattern) and another graph G (the host), the task is to count the occurrences of H as a subgraph in G. Our goal is to understand which properties of the pattern class C make the problem #$\mathsf{Sub}(C)$ easy/hard. For instance, for the class of stars, we can solve this problem in linear time. For the class of paths however, it subsumes counting Hamiltonian paths and is hence #P-hard.

As it turns out, the notion of #P-hardness fails to give a sweeping dichotomy for the problems #$\mathsf{Sub}(C)$, since there exist classes C of intermediate complexity. However, adopting the framework of fixed-parameter tractability, and parameterizing by the size of the pattern, it was shown in 2014 how to classify the problems #$\mathsf{Sub}(C)$ as either polynomial-time solvable or #$\mathsf{W}[1]$-hard: A class C lies on the polynomial-time side of this dichotomy iff the graphs appearing in C have vertex-covers of constant size.

In this talk, we introduce a new technique that allows us to view the subgraph counting problem from a new perspective. In particular, it allows for the following applications:

1. A greatly simplified proof of the 2014 dichotomy result, together with almost-tight lower bounds under ETH, which were not achievable before.
2. Faster algorithms for counting k-edge subgraphs, such as k-matchings, with running time n^{ck} for constants $c < 1$.

3.7 Finding Detours is Fixed-parameter Tractable

Holger Dell (Universität des Saarlandes, DE)

URL https://arxiv.org/abs/1607.07737
Joint work of Ivona Bezáková, Radu Curticapean, Holger Dell, Fedor V. Fomin

We consider the following natural “above guarantee” parameterization of the classical Longest Path problem: For given vertices s and t of a graph G, and an integer k, the problem Longest Detour asks for an (s,t)-path in G that is at least k longer than a shortest (s,t)-path. Using insights into structural graph theory, we prove that Longest Detour is fixed-parameter tractable (FPT) on undirected graphs and actually even admits a single-exponential algorithm, that is, one of running time $\mathsf{exp}(O(k)) \cdot \mathsf{poly}(n)$. This matches (up to the base of the exponential) the best algorithms for finding a path of length at least k.

Furthermore, we study the related problem Exact Detour that asks whether a graph G contains an (s,t)-path that is exactly k longer than a shortest (s,t)-path. For this problem, we obtain a randomized algorithm with running time about 2.746^k, and a deterministic algorithm with running time about 6.745^k, showing that this problem is FPT as well. Our algorithms for Exact Detour apply to both undirected and directed graphs.

3.8 Average-Case Analysis of Parameterized Problems

Tobias Friedrich (Hasso-Plattner-Institut – Potsdam, DE)

Joint work of Karl Bringmann, Tobias Friedrich, Danny Hermelin, Christian Hercher, Nikolaos Fountoulakis

Many computational problems are NP-hard and are therefore generally believed not to be solvable in polynomial time. Additional assumptions on the inputs are necessary to solve such problems efficiently. Two typical approaches are (i) parameterized complexity where we assume that a certain parameter of the instances is small, and (ii) average-case complexity where we assume a certain probability distribution on the inputs. There is a vast literature on both approaches, but very little about their intersection. Nevertheless, combining these two approaches seems natural and potentially useful in practice. The talk presents the following line of results:

- A hierarchy of parameterized average-case complexity classes [2].
- The W[1]-complete problem k-clique drops to an average-case analog of FPT for homogeneous Erdős-Rényi random graphs of all densities [2] and for inhomogeneous Chung-Lu random graphs with power-law exponent $\gamma > 2$ [4, 5].
- The bounded search tree paradigm allows analyzing average-case run times for a very relaxed graph model that only assumes stochastic independence of the edges. This is demonstrated for the parameterized problems k-Clique, Vertex Cover, and Hitting Set [unpublished].
- The Edge Cover Problem has no kernel of subexponential size in the worst-case (unless P = NP). We study a well-known set of reduction rules and prove that random intersection graphs are reduced completely by these rules [3].
- The geometric problem of computing the hypervolume indicator is W[1]-hard in the worst-case, but can be solved in expected FPT-time if the input is distributed at random on a d-dimensional simplex [1].

References

1 Karl Bringmann and Tobias Friedrich. Parameterized average-case complexity of the hypervolume indicator. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 575–582. ACM, 2013.

2 Nikolaos Fountoulakis, Tobias Friedrich, and Danny Hermelin. On the average-case complexity of parameterized clique. *Theoretical Computer Science*, 576:18–29, 2015.

3 Tobias Friedrich and Christian Hercher. On the kernel size of clique cover reductions for random intersection graphs. *Journal of Discrete Algorithms*, 34:128–136, 2015.

4 Tobias Friedrich and Anton Krohmer. Parameterized clique on scale-free networks. In *International Symposium on Algorithms and Computation (ISAAC)*, volume 7676 of *Lecture Notes in Computer Science*, pages 659–668. Springer, 2012.

5 Tobias Friedrich and Anton Krohmer. Parameterized clique on inhomogeneous random graphs. *Discrete Applied Mathematics*, 184:130–138, 2015.

3.9 Spanning Circuits in Regular Matroids

Petr A. Golovach (University of Bergen, NO)

Joint work of Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, Saket Saurabh
Main reference F. V. Fomin, P. A. Golovach, D. Lokshtanov, S. Saurabh, "Spanning Circuits in Regular Matroids", in Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017), pp. 1433–1441, SIAM, 2017.
URL http://dx.doi.org/10.1137/1.9781611974782.93

We consider the fundamental Matroid Theory problem of finding a circuit in a matroid spanning a set T of given terminal elements. For graphic matroids this corresponds to the problem of finding a simple cycle passing through a set of given terminal edges in a graph. The algorithmic study of the problem on regular matroids, a superclass of graphic matroids, was initiated by Gavenčiak, Král', and Oum [ICALP'12], who proved that the case of the problem with $|T| = 2$ is fixed-parameter tractable (FPT) when parameterized by the length of the circuit. We extend the result of Gavenčiak, Král', and Oum by showing that for regular matroids

- the Minimum Spanning Circuit problem, deciding whether there is a circuit with at most ℓ elements containing T, is FPT parameterized by $k = \ell - |T|$;
- the Spanning Circuit problem, deciding whether there is a circuit containing T, is FPT parameterized by $|T|$.

We note that extending our algorithmic findings to binary matroids, a superclass of regular matroids, is highly unlikely: Minimum Spanning Circuit parameterized by ℓ is W[1]-hard on binary matroids even when $|T| = 1$. We also show a limit to how far our results can be strengthened by considering a smaller parameter. More precisely, we prove that Minimum Spanning Circuit parameterized by $|T|$ is W[1]-hard even on cographic matroids, a proper subclass of regular matroids.

3.10 Parameterized Traveling Salesman Problem: Beating the Average

Gregory Z. Gutin (Royal Holloway University of London, GB)

Joint work of Gregory Z. Gutin, Viresh Patel
Main reference G. Gutin, V. Patel, "Parameterized Traveling Salesman Problem: Beating the Average", SIAM J. Discrete Math., 30(1):220–238, SIAM, 2016.
URL http://dx.doi.org/10.1137/140980946

In the traveling salesman problem (TSP), we are given a complete graph K_n together with an integer weighting w on the edges of K_n, and we are asked to find a Hamilton cycle of K_n of minimum weight. Let $h(w)$ denote the average weight of a Hamilton cycle of K_n for the weighting w. Vizing in 1973 asked whether there is a polynomial-time algorithm which always finds a Hamilton cycle of weight at most $h(w)$. He answered this question in the affirmative and subsequently Rublineckii, also in 1973, and others described several other TSP heuristics satisfying this property. We prove a considerable generalization of Vizing's result: for each fixed k, we give an algorithm that decides whether, for any input edge weighting w of K_n, there is a Hamilton cycle of K_n of weight at most $h(w) - k$ (and constructs such a cycle if it exists). For k fixed, the running time of the algorithm is polynomial in n, where the degree of the polynomial does not depend on k (i.e., the generalized Vizing problem is fixed-parameter tractable with respect to the parameter k).

3.11 How proofs are prepared at Camelot

Petteri Kaski (Aalto University, FI)

Joint work of Andreas Björklund, Petteri Kaski
Main reference A. Björklund, P. Kaski, "How Proofs are Prepared at Camelot: Extended Abstract", in Proc. of the 2016 ACM Symposium on Principles of Distributed Computing (PODC 2016), pp. 391–400, ACM, 2016.
URL http://dx.doi.org/10.1145/2933057.2933101

We study a design framework for robust, independently verifiable, and workload-balanced distributed algorithms working on a common input. The framework builds on recent noninteractive Merlin–Arthur proofs of batch evaluation of Williams [31st IEEE Colloquium on Computational Complexity (CCC'16, May 29-June 1, 2016, Tokyo), 2:117] with the basic observation that Merlin's magic is not needed for batch evaluation: mere Knights can prepare the independently verifiable proof, in parallel, and with intrinsic error-correction.

As our main technical result, we show that the k-cliques in an n-vertex graph can be counted and verified in per-node $O(n(\omega+\epsilon)\frac{k}{6})$ time and space on $O(n(\omega+\epsilon)\frac{k}{6})$ compute nodes, for any constant $\epsilon > 0$ and positive integer k divisible by 6, where $2 \leq \omega < 2.3728639$ is the exponent of square matrix multiplication over the integers. This matches in total running time the best known sequential algorithm, due to Nešetřil and Poljak [Comment. Math. Univ. Carolin. 26 (1985) 415–419], and considerably improves its space usage and parallelizability. Further results include novel algorithms for counting triangles in sparse graphs, computing the chromatic polynomial of a graph, and computing the Tutte polynomial of a graph.

3.12 Improved algebraic algorithms for out-branchings problems

Yiannis Koutis (University of Puerto Rico – Rio Piedras, PR)

We present an $O^*(2^k)$ algorithm for deciding if a directed graph contains an out-branching with at least k internal nodes. We also present an algorithm for detecting out-branchings with at least k leaves and at most s internal nodes with out-degree greater than 1. The algorithm runs in time $O^*(2^{k+s})$, and for certain values of s it improves upon the previous upper bounds for the k-leaf problem. The algorithms are algebraic and work via reductions to two non-standard problems concerning monomial detection in multivariate polynomials.

3.13 Improving TSP tours using dynamic programming over tree decomposition

Łukasz Kowalik (University of Warsaw, PL)

Joint work of Marek Cygan, Łukasz Kowalik, Arkadiusz Socała

Given a traveling salesman problem (TSP) tour H in graph G a k-move is an operation which removes k edges from H, and adds k edges of G so that a new tour H' is formed. The popular k-OPT heuristics for TSP finds a local optimum by starting from an arbitrary tour H and then improving it by a sequence of k-moves.

Until 2016, the only known algorithm to find an improving k-move for a given tour was the naive solution in time $O(n^k)$. At ICALP'16 de Berg, Buchin, Jansen and Woeginger showed an $O(n^{\lfloor \frac{2}{3k} \rfloor + 1})$-time algorithm.

We show an algorithm which runs in $O(n^{(\frac{1}{4}+\epsilon_k)k})$ time, where $\lim \epsilon_k = 0$. We are able to show that it improves over the state of the art for every $k = 5, \ldots, 10$. For the most practically relevant case $k = 5$ we provide a slightly refined algorithm running in $O(n^{3.4})$ time. We also show that for the $k = 4$ case, improving over the $O(n^3)$-time algorithm of de Berg et al. would be a major breakthrough: an $O(n^{3-\epsilon})$-time algorithm for any $\epsilon > 0$ would imply an $O(n^{3-\delta})$-time algorithm for the APSP problem, for some $\delta > 0$.

3.14 A Randomized Polynomial Kernelization for Vertex Cover with a Smaller Parameter

Stefan Kratsch (Universität Bonn, DE)

Main reference S. Kratsch, "A Randomized Polynomial Kernelization for Vertex Cover with a Smaller Parameter", in Proc. of the 24th Annual European Symposium on Algorithms (ESA 2016), LIPIcs, Vol. 57, pp. 59:1-59:17, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ESA.2016.59

In the Vertex Cover problem we are given a graph $G = (V, E)$ and an integer k and have to determine whether there is a set $X \subseteq V$ of size at most k such that each edge in E has at least one endpoint in X. The problem can be easily solved in time $O^*(2^k)$, making it fixed-parameter tractable (FPT) with respect to k. While the fastest known algorithm takes only time $O^*(1.2738^k)$, much stronger improvements have been obtained by studying *parameters that are smaller than k*. Apart from treewidth-related results, the arguably best algorithm for Vertex Cover runs in time $O^*(2.3146^p)$, where $p = k - LP(G)$ is only the excess of the solution size k over the best fractional vertex cover (Lokshtanov et al. TALG 2014). Since $p \leq k$ but k cannot be bounded in terms of p alone, this strictly increases the range of tractable instances.

Recently, Garg and Philip (SODA 2016) greatly contributed to understanding the parameterized complexity of the Vertex Cover problem. They prove that $2LP(G) - MM(G)$ is a lower bound for the vertex cover size of G, where $MM(G)$ is the size of a largest matching of G, and proceed to study parameter $\ell = k - (2LP(G) - MM(G))$. They give an algorithm of running time $O^*(3^\ell)$, proving that Vertex Cover is FPT in ℓ. It can be easily observed that $\ell \leq p$ whereas p cannot be bounded in terms of ℓ alone. We complement the work of Garg and Philip by proving that Vertex Cover admits a randomized polynomial kernelization in terms of ℓ, i.e., an efficient preprocessing to size polynomial in ℓ. This improves over parameter $p = k - LP(G)$ for which this was previously known (Kratsch and Wahlström FOCS 2012).

3.15 Gap Amplification using Bipartite Random Graphs

Bingkai Lin (National Institute of Informatics – Tokyo, JP)

Gap amplification transformation plays an important role in proving hardness of approximation results. This talk presents a new method to construct gap amplification reduction for parameterized optimization problems. First, I will review the threshold phenomenon of random graphs $G(n, p)$ containing a bipartite complete subgraph . Then I will show its application on ruling out super-polynomial time algorithms for approximating Maximum k-Set Intersection and Minimum Set Cover to some ratios.

3.16 Lossy Kernelization I

M. S. Ramanujan (TU Wien, AT)

Joint work of Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh
Main reference D. Lokshtanov, F. Panolan, M. S. Ramanujan, S. Saurabh, "Lossy Kernelization", arXiv:1604.04111v2 [cs.DS], 2016.
URL https://arxiv.org/abs/1604.04111v2

Introductory talk on a new framework for analyzing the performance of preprocessing algorithms. This framework builds on the notion of kernelization from parameterized complexity. However, as opposed to the original notion of kernelization, this framework combines very well with approximation algorithms and heuristics.

3.17 Lossy Kernelization II: Cycle Packing

Fahad Panolan (University of Bergen, NO)

Joint work of Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh
Main reference D. Lokshtanov, F. Panolan, M. S. Ramanujan, S. Saurabh, "Lossy Kernelization", arXiv:1604.04111v2 [cs.DS], 2016.
URL https://arxiv.org/abs/1604.04111v2

In this talk we see an example of Lossy Kernelization – Disjoint Factors. Disjoint Factors problem is closely related to Cycle Packing. We prove that Disjoint Factors admits a Polynomial Sized Approximate Kernelization Scheme (PSAKS).

3.18 Lossy Kernelization, III: Lower Bounds

Daniel Lokshtanov (University of Bergen, NO)

Joint work of Daniel Lokshtanov, Fahad Panolan, M. S. Ramanujan, Saket Saurabh
Main reference D. Lokshtanov, F. Panolan, M.S. Ramanujan, S. Saurabh, "Lossy Kernelization", arXiv:1604.04111v2 [cs.DS], 2016.
URL https://arxiv.org/abs/1604.04111v2

We show how to combine the tecniques for showing kernelization lower bounds with the methods for showing hardness of approximation to rule out approximate kernels of polynomial size for concrete problems. We outline proofs that the longest path problem parameterized by solution size, and the set cover problem parameterized by the size of the universe do not admit constant factor approximate kernels of polynomial size.

3.19 Exponential Time Paradigms Through the Polynomial Time Lens

Jesper Nederlof (TU Eindhoven, NL)

Joint work of Andy Drucker, Jesper Nederlof, Rahul Santhanam
Main reference A. Drucker, J. Nederlof, R. Santhanam, "Exponential Time Paradigms Through the Polynomial Time Lens", in Proc. of the 24th Annual European Symposium on Algorithms (ESA 2016), LIPIcs, Vol. 57, pp. 36:1-36:14, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ESA.2016.36

We propose a general approach to modelling algorithmic paradigms for the exact solution of NP-hard problems. Our approach is based on polynomial time reductions to succinct versions of problems solvable in polynomial time. We use this viewpoint to explore and compare the power of paradigms such as branching and dynamic programming, and to shed light on the true complexity of various problems.

In this talk I will mainly talk about lower bounds for OPP algorithms. For example, if there is a polynomial time algorithm that, given a planar graph, outputs a maximum independent set of n vertices with probability $\mathsf{exp}(-n^{1-\epsilon})$ for some $\epsilon > 0$, then $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. I will also outline connections with "AND-compositions" from kernelization theory.

3.20 Faster Space-Efficient Algorithms for Subset Sum, k-Sum and Related Problems

Jesper Nederlof (TU Eindhoven, NL)

Joint work of Nikhil Bansal, Shashwat Garg, Jesper Nederlof, Nikhil Vyas
Main reference N. Bansal, S. Garg, J. Nederlof, N. Vyas, "Faster Space-Efficient Algorithms for Subset Sum, k-Sum and Related Problems", arXiv:1612.02788v1 [cs.DS], 2016.
URL https://arxiv.org/abs/1612.02788v1

We present a randomized Monte Carlo algorithm that solves a given instance of Subset Sum on n integers using $O^*(2^{0.86n})$ time and $O^*(1)$ space, where $O^*()$ suppresses factors polynomial in the input size. The algorithm assumes random access to the random bits used. The same result can be obtained for Knapsack on n items, and the same methods also have consequences for the k-Sum problem.

3.21 Subexponential Parameterized Algorithms for Planar Graphs, Apex-Minor-Free Graphs and Graphs of Polynomial Growth via Low Treewidth Pattern Covering

Marcin Pilipczuk (University of Warsaw, PL) and Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

Joint work of Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, Saket Saurabh
Main reference F. V. Fomin, D. Lokshtanov, D. Marx, Ma. Pilipczuk, Mi. Pilipczuk, S. Saurabh, "Subexponential Parameterized Algorithms for Planar and Apex-Minor-Free Graphs via Low Treewidth Pattern Covering", in Proc. of the 57th Annual Symposium on Foundations of Computer Science (FOCS 2016), pp. 515–524, IEEE, 2016.
URL http://dx.doi.org/10.1109/FOCS.2016.62

We prove the following theorem. Given a planar graph G and an integer k, it is possible in polynomial time to randomly sample a subset A of vertices of G with the following properties:

- A induces a subgraph of G of treewidth $O(\sqrt{k}\log k)$, and
- for every connected subgraph H of G on at most k vertices, the probability that A covers the whole vertex set of H is at least $(2^{O(\sqrt{k}\log^2 k)} \cdot n^{O(1)})^{-1}$, where n is the number of vertices of G.

Together with standard dynamic programming techniques for graphs of bounded treewidth, this result gives a versatile technique for obtaining (randomized) subexponential parameterized algorithms for problems on planar graphs, usually with running time bound $2^{O(\sqrt{k}\log^2 k)}n^{O(1)}$. The technique can be applied to problems expressible as searching for a small, connected pattern with a prescribed property in a large host graph; examples of such problems include Directed k-Path, Weighted k-Path, Vertex Cover Local Search, and Subgraph Isomorphism, among others. Up to this point, it was open whether these problems can be solved in subexponential parameterized time on planar graphs, because they are not amenable to the classic technique of bidimensionality. Furthermore, all our results hold in fact on any class of graphs that exclude a fixed apex graph as a minor, in particular on graphs embeddable in any fixed surface. We also provide a similar statement for graph classes of polynomial growth.

3.22 Exact Algorithms via Monotone Local Search

Saket Saurabh (The Institute of Mathematical Sciences, India, IN)

Joint work of Daniel Lokshtanov, Serge Gaspers, Fedor Fomin, Saket Saurabh

In a vertex subset problem we are given as input a universe U of size n, and a family F of subsets of the universe defined implicitly from the input. The task is to find a subset S in F of smallest possible size. For an example the Vertex Cover problem is a subset problem where input is a graph G, the universe is the vertex set of G, and the family F is the family of all vertex covers of G. Here a vertex set S is a vertex cover of G if every edge of G has at least one endpoint in S. Many problems, such as Vertex Cover, Feedback Vertex Set, Hitting Set and Minimum Weight Satisfiability can be formalized as vertex subset problems. The trivial algorithm for such problems runs in time 2^n. We show that (essentially) any vertex subset problem that admits an FPT algorithm with running time $c^{k}n^{O(1)}$, where c is a constant and

k is the size of the optimal solution, also admits an algorithm with running time $(2 - \frac{1}{c})^n$. In one stroke this theorem improves the best known exact exponential time algorithms for a number of problems, and gives tighter combinatorial bounds for several well-studied objects. The most natural variant of our algorithm is randomized, we also show how to get a deterministic algorithm with the same running time bounds, up to a sub-exponential factor in the running time. Our de-randomization relies on a new pseudo-random construction that may be of independent interest.

3.23 Backdoors for Constraint Satisfaction

Stefan Szeider (TU Wien, AT)

Joint work of Robert Ganian, Serge Gaspers, Neeldhara Misra, Sebastian Ordyniak, M. S Ramanujan, Stefan Szeider, Stanislav Živný
Main reference S. Gaspers, S. Ordyniak, S. Szeider, "Backdoor Sets for CSP", in The Constraint Satisfaction Problem: Complexity and Approximability, Dagstuhl Follow-Ups, Vol. 7, pp. 137–157, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017.
URL http://dx.doi.org/10.4230/DFU.Vol7.15301.137

We will review some recent parameterised complexity results for the Constraint Satisfaction Problem (CSP), considering parameters that arise from strong backdoor sets into CSP classes defined by tractable constraint languages. The language restrictions have recently stepped into the spotlight because of the recently claimed solution of the long-standing Dichotmy Conjecture. One of the results we will present is based on a novel combination of backdoor sets and treewidth.

3.24 Parameterized Algorithms for Matrix Factorization Problems

David P. Woodruff (IBM Almaden Center – San Jose, US)

I will give a survey on parameterized algorithms for matrix factorization problems, focusing on non-negative matrix factorization, ℓ_1 low rank factorization, tensor factorization, and weighted low rank approximation.

3.25 k-Path of Algorithms

Meirav Zehavi (University of Bergen, NO)

An overview of several algorithms for the k-Path problem and the tools employed to derandomize them, including a presentation of a simple algorithm for the Longest Cycle problem.

4 Open problems

4.1 FPT-approximation of bandwidth

Daniel Lokshtanov (University of Bergen, NO)

Bandwidth
Input: An undirected graph $G = (V, E)$, integer k.
Question: Is there an ordering (injective function) $\pi : V \to \{1, \ldots, |V|\}$, such that $\max_{uv \in E} |\pi(u) - \pi(v)| \leq k$.

▶ **Open Problem 1.** *Is there an FPT-approximation on general graphs parameterized by k? In particular none of the following is known:*

- *Is there $(1 + \epsilon)$-approximation in FPT time?*
- *Is there constant approximation in FPT time?*
- *Is there $f(k)$-approximation in FPT time?*

Relevant reference: In [11] a polynomial time $k^{O(k)}$-approximation is shown for trees and graphs of bounded treelength.

4.2 Time and space complexity of k-LCS

Michał Pilipczuk (University of Warsaw, PL)

k-Longest Common Subsequence (k-LCS)
Input: alphabet Σ, strings $s_1, \ldots, s_k \in \Sigma^*$.
Question: what is the longest common subsequence of all the strings s_i.

The standard dynamic programming has running time and space complexity $O(n^k)$. By Savitch's theorem we can reduce the space complexity to $\mathsf{poly}(k, n)$ at the cost of increasing the running time to $n^{O(k \log n)}$.

▶ **Open Problem 2.** *Is k-LCS solvable in $n^{f(k)}$ time and FPT space?*

Relevant reference: in [21] a connection is proved between this open problem and the question of space efficient algorithms for bounded treewidth graphs. Other relevant reference: [12].

4.3 Fine-grained complexity of k-LCS

Karl Bringmann (MPI für Informatik – Saarbrücken, DE)

We can solve k-LCS (defined above) in time $O(n^k)$, but under SETH there is no $O(n^{k-\epsilon})$ time algorithm $|\Sigma| = \Omega(k)$ [1]. On the other hand we know for $|\Sigma| = O(1)$ the problem is W[1]-hard and there is no $n^{o(k)}$ time algorithm [19].

► **Open Problem 3.** *Is there an $O(n^{(1-\epsilon_\Sigma)k})$ time algorithm?*

4.4 Fine-grained complexity of Hitting Set w.r.t. VC dimension

Karl Bringmann (MPI für Informatik – Saarbrücken, DE)

Hitting Set
Input: a set family $\mathcal{F} \subseteq 2^U$, integer k.
Question: Is there a set $X \subseteq U$ of size at most k, such that X interesects each set in $\mathcal{F}$.

We know that Hitting Set can be solved in time $n^{k+o(1)}$ (for $k \geq 7$), and under the Strong Exponential Time Hypothesis (SETH) no $O(n^{k-\epsilon})$ time algorithm exists [18].

► **Definition 1.** We say that a set $X \subseteq U$ is *shattered* by a set family $\mathcal{F} \subseteq 2^U$ if the family $\{X \cap S : S \in \mathcal{F}\}$ contains all the subsets of X. The VC dimension of $\mathcal{F}$ is the largest cardinality of a set X, such that X is shattered by $\mathcal{F}$.

It is known that for VC $= 1$ the Hitting Set problem is polynomial time solvable, while for VC $= 2$ the problem becomes W[1]-hard and does not admit $n^{o(\frac{k}{\log k})}$ time algorithm [6].

► **Open Problem 4.** *Is there $O(n^{(1-\epsilon_{VC})k})$ time algorithm for the Hitting Set problem?*

4.5 FPT-approximation of VC dimension

Bingkai Lin (National Institute of Informatics – Tokyo, JP)

► **Open Problem 5.** *Is there a constant-factor FPT-time approximation algorithm for VC dimension (defined above)?*

4.6 Better approximation of Dominating Set

Bingkai Lin (National Institute of Informatics – Tokyo, JP)

Dominating Set
Input: an undirected graph G, an integer k.
Question: is there a set $X \subseteq V(G)$ of size at most k, such that each vertex of G is in X or has a neighbour in X?

It is well known that Dominating Set admits polynomial time $\ln(n)$-approximation algorithm as well as $n^{O(k)}$ time exact algorithm.

▶ **Open Problem 6.** *Is there an $o(\ln n)$-approximation algorithm for the Dominating Set problem running in time $n^{k-\epsilon}$?*

4.7 Orthogonal Vectors for Subset Sum

Jesper Nederlof (TU Eindhoven, NL)

Orthogonal Vectors for Subset Sum (OVSS)
Input: $\mathcal{A}, \mathcal{B} \subseteq \binom{[d]}{d/4}$.
Question: is there $A \in \mathcal{A}$, $B \in \mathcal{B}$ such that $A \cap B = \emptyset$?

We are satisfied with any algorithm with constant error probability. For an integer d, denote $[d] = \{1, \dots, d\}$ and $\binom{[d]}{d/4}$ for the set of all subsets of $[d]$ of size $d/4$. Let $h(\cdot)$ denote the binary entropy function and $\tilde{O}$ omit factors polynomial in d.

▶ **Open Problem 7.** *Solve OVSS in time $\tilde{O}\left((|\mathcal{A}| + \mathcal{B}|) \cdot \frac{2^{(1-\epsilon)d}}{\binom{d}{d/4}}\right)$ for $\epsilon > 0$.*

Observations: Let $\alpha = 1 - h(1/4) \approx 0.1888$. Note that $2^{\alpha d} = 2^d / \binom{d}{d/4}$.

- There is an $\tilde{O}((|\mathcal{A}| + |\mathcal{B}|)2^{\alpha d})$ time algorithm based on representative sets (see [16] for an extended version of this open problem statement outlining the algorithm).
- If $|\mathcal{A}| \leq 2^{\alpha' d}$ or $|\mathcal{B}| \leq 2^{\alpha' d}$ for $\alpha' < \alpha$, then trivial enumeration works. Moreover, by directly using the improvements over this trivial enumeration from [2, 7, 13], we may in fact assume $|\mathcal{A}|, |\mathcal{B}| \geq 2^{(\alpha+\delta)d}$ for some $\delta > 0$.
- If $|\mathcal{A}| > 2^{\beta d}$, where $\beta > h(1/4) - (1 - h(1/4)) \approx 0.6223$, an algorithm of Björklund et al. [5] works: it runs in time $\tilde{O}((|\downarrow \mathcal{A}| + |\downarrow \mathcal{B}|)) \leq \tilde{O}(2^{h(1/4)d})$, where for a set family $\mathcal{F}$, $\downarrow \mathcal{F}$ denotes the sets of subsets of elements of $\mathcal{F}$.
- In fact, $|\downarrow \mathcal{A}|$ can be upper bounded by $\tilde{O}(\max_\lambda \min\{|\mathcal{A}|\binom{d/4}{\lambda d}, \binom{d}{\lambda d}\})$. After a small calculation, this gives that the algorithm from [5] is fast enough whenever $\beta > 0.525$.

▶ **Open Problem 8.** *Does there exist for some constant $c > 0$ an algorithm that, given $z = 2^{cd}$ instances $(\mathcal{A}_1, \mathcal{B}_1), \dots, (\mathcal{A}_z, \mathcal{B}_z)$ of OVSS, detects whether any instance is a YES-instance in time $(\sum_{i=1}^{z}(|\mathcal{A}_i| + |\mathcal{B}_i|))2^{(\alpha-\epsilon)d}$, for $\epsilon > 0$?*

Note Open Problem 8 relaxes Open Problem 7 as it asks whether exponentially many instances of OVSS can be solved fast in an amortized sense.

Motivation: Following the approach of [4], a positive answer would imply an $\tilde{O}(2^{(.5-\epsilon)n})$ time algorithm for n-integer subset sum for some $\epsilon > 0$.

4.8 Fixed parameter tractability of Weighted Low Rank Approximation

David P. Woodruff (IBM Almaden Center – San Jose, US)

Weighted Low Rank Approximation
Input: $n \times n$ matrix A over reals, rank bound $r = O(1)$, weight matrix $W \in \mathbb{R}^{n \times n}$
Goal: find a rank r matrix B such that the weighted Frobenius norm of the difference $|W \circ (A - B)|_F = \sum(W_{i,j} \cdot (A_{i,j} - B_{i,j})^2)$ is small, i.e., at most $1.01 \cdot OPT$

We assume the entries of A and W are integers in the range $\{-M, -M+1, \dots, M\}$ for an integer $M \leq 2^{poly(n)}$, i.e., that the entries of A and W can be specified using $poly(n)$ bits.

▶ **Open Problem 9.** *Is there an FPT algorithm for this problem when parameterized by the rank of the weight matrix W?*

It is known [22] that there is an $n^{O(k)}$ upper bound and conditional $2^{\Omega(k)}$ lower bound.

4.9 Short resolution refutations for SAT when parameterized by treewidth

Stefan Szeider (TU Wien, AT)

We consider propositional formulas in conjunctive normal form (CNF), given as a set of clauses, where each clause is a set of literals, e.g., $F = \{\{x,y\}, \{x,\bar{y},z\}, \{\bar{x},y\}, \{\bar{x},\bar{y}\}, \{\bar{z}\}\}$.

▶ **Definition 1.** A clause C is the *resolvent* of clauses C_1 and C_2 if there is exactly one variable x such that $x \in C_1$, $\bar{x} \in C_2$, and $C = (C_1 \setminus \{x\}) \cup (C_2 \setminus \{\bar{x}\})$.

A *resolution refutation* of a formula F is a vertex-labeled dag with exactly one sink where each vertex has in-degree 0 or 2. Each node is labeled with a clause as follows: (i) each source is labeled with a clause from F, (ii) each non-source is labeled with the resolvent of the clauses labeling its predecessors, and (iii) the clause which labels the sink is empty.

The *size* of a resulution refutation is the number of its vertices.

It is known that a formula is unsatisfiable if and only if it has a resolution refutation.

▶ **Definition 2.** The *primal graph* $P(F)$ of a formula F is the graph whose vertices are the variables of F, where two vertices are connected by an edge iff the corresponding variables appear together (negated or unnegated) in some clause.

The *incidence graph* $I(F)$ is the bipartite graph between variables and clauses where two vertices are connected by an edge iff the corresponding variable appears (negated or unnegated) in the corresponding clause.

It is known that for any formula F the treewidth of its incidence graph is at most the treewidth of its primal graph plus one:

$$\mathsf{tw}(I(F)) \leq tw(P(F)) + 1\,.$$

Also, it is known that #SAT is FPT when parameterized by $\mathsf{tw}(I(F))$ and $\mathsf{tw}(P(F))$. Further, it is known that every unsatisfiable formula F has a resolution refutation of FPT size when parameterized by $\mathsf{tw}(P(F))$.

▶ **Open Problem 10.** *Is there always a resolution refutation of FPT size when parameterized by* $\mathsf{tw}(I(F))$*?*

4.10 Small universal Steiner tree covers

Marcin Pilipczuk (University of Warsaw, PL)

Let G be a graph embedded on the plane in such a manner that the outerface of G, denoted henceforth ∂G, is a simple cycle of length k. For a set $T \subseteq V(\partial G)$ and $A \subseteq V(G)$, we say that A *covers* an optimal Steiner tree for T if there exists an optimum Steiner tree in G with terminals T, such that every vertex of degree at least three in this tree lies in A. A set A is a *universal Steiner tree cover* in G if A covers an optimal Steiner tree for every $T \subseteq V(\partial G)$.

In [20] we have shown an existence of a universal Steiner tree cover of size bounded polynomially in k, but the degree of the bound is above 100. On the other hand, we do not know any example that is significantly worse than a grid of perimeter k.

▶ **Open Problem 11.** *Prove or disprove the following statement: for every such G, there exists a universal Steiner tree cover of size $\widetilde{O}(k^2)$.*

4.11 Even Set

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

Even Set
Input: Set system $\mathcal{S}$ over a universe U, integer k.
Find: A *nonempty* set $X \subseteq U$ of size at most k such that $|X \cap S|$ is even for every $S \in \mathcal{S}$.

Essentially equivalent formulations:

- With graphs and neighborhoods.
- Minimum circuit in a binary matroid.
- Minimum distance in a linear code over a binary alphabet.

▶ **Open Problem 12.** *What is the parameterized complexity of Even Set? Is it fixed parameter tractable?*

4.12 FPT-approximation of Maximum Clique and Minimum Dominating Set

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

▶ **Open Problem 13.** *Can Maximum Clique (Minimum Dominating Set) be approximated in FPT time? I.e., is there an algorithm running in time $f(k) \cdot n^{O(1)}$ that, given a graph G and an integer k, finds a $g(k)$-clique (dominating set of size $g(k)$) for some unbounded nondecreasing function g or correctly states that there is no k-clique (dominating set of size k) in G?*

4.13 Polynomial (Turing) Kernels

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

▶ **Open Problem 14.** *Do the following problems have polynomial kernels?*

- *Directed Feedback Vertex Set*
- *Multiway Cut (with arbitrary number t of terminals)*
- *Planar Vertex Deletion*

Does k-Path have a polynomial Turing kernel?

4.14 Directed Odd Cycle Traversal

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

Directed Odd Cycle Traversal
Input: Directed graph G, integer k.
Find: A set $X \subseteq U$ of at most k vertices such that $G - X$ has no directed cycle of odd length.

This problem generalizes

- Directed Feedback Vertex Set [9]
- Odd Cycle Transversal [23]
- Directed S-Cycle Transversal [10]

▶ **Open Problem 15.** *What is the parameterized complexity of Directed Odd Cycle Traversal?*

4.15 Square root phenomenon

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

▶ **Open Problem 16.** *Are there $2^{O(\sqrt{k}\cdot\mathrm{polylog}(k))}n^{O(1)}$ time FPT algorithms for planar problems?*

Natural targets are
- Steiner Tree
- Directed Steiner Tree
- Directed Subset TSP

What about counting problems?
- k-path
- k-mathching
- k disjoint triangles
- k independent set

4.16 Disjoint paths / minor testing

Dániel Marx (Hungarian Academy of Sciences – Budapest, HU)

The best known parameter dependence for the k-disjoint paths problem and H-minor testing seems to be triple exponential [15] using [8]. For planar graphs [3] gave an $2^{2^{\mathsf{poly}(k)}}n^{O(1)}$ algorithm.

▶ **Open Problem 17.** *Are there $2^{\mathsf{poly}(k)}n^{O(1)}$ time algorithms for planar or general graphs?*

References

1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. Tight hardness results for LCS and other sequence similarity measures. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 59–78. IEEE Computer Society, 2015. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7352273, doi:10.1109/FOCS.2015.14.

2 Amir Abboud, Richard Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In Indyk [14], pages 218–230. doi:10.1137/1.9781611973730.17.

3 Isolde Adler, Stavros G Kolliopoulos, Philipp Klaus Krause, Daniel Lokshtanov, Saket Saurabh, and Dimitrios Thilikos. Tight bounds for linkages in planar graphs. In *International Colloquium on Automata, Languages, and Programming*, pages 110–121. Springer, 2011.

4 Per Austrin, Petteri Kaski, Mikko Koivisto, and Jesper Nederlof. Dense subset sum may be the hardest. In Ollinger and Vollmer [17], pages 13:1–13:14. doi:10.4230/LIPIcs.STACS.2016.13.

5 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Counting paths and packings in halves. In Amos Fiat and Peter Sanders, editors, *Algorithms – ESA 2009, 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009. Proceedings*, volume 5757 of *Lecture Notes in Computer Science*, pages 578–586. Springer, 2009. doi:10.1007/978-3-642-04128-0_52.

6 Karl Bringmann, László Kozma, Shay Moran, and N. S. Narayanaswamy. Hitting set for hypergraphs of low vc-dimension. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, volume 57 of *LIPIcs*, pages 23:1–23:18. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016. URL: http://www.dagstuhl.de/dagpub/978-3-95977-015-6, doi:10.4230/LIPIcs.ESA.2016.23.

7 Timothy M. Chan. Speeding up the four russians algorithm by about one more logarithmic factor. In Indyk [14], pages 212–217. doi:10.1137/1.9781611973730.16.

8 Chandra Chekuri and Julia Chuzhoy. Degree-3 treewidth sparsifiers. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 242–255. Society for Industrial and Applied Mathematics, 2015.

9 Jianer Chen, Yang Liu, Songjian Lu, Barry O'sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *Journal of the ACM (JACM)*, 55(5):21, 2008.

10 Rajesh Chitnis, Marek Cygan, Taghi Hajiaghayi, Mohammad, Marcin Pilipczuk, and Michal Pilipczuk. Designing fpt algorithms for cut problems using randomized contractions. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 460–469. IEEE, 2012.

11 Markus Sortland Dregi and Daniel Lokshtanov. Parameterized complexity of bandwidth on trees. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming – 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 405–416. Springer, 2014. doi:10.1007/978-3-662-43948-7_34.

12 Michael Elberfeld, Christoph Stockhusen, and Till Tantau. On the space and circuit complexity of parameterized problems: Classes and completeness. *Algorithmica*, 71(3):661–701, 2015. doi:10.1007/s00453-014-9944-y.

13 Russell Impagliazzo, Shachar Lovett, Ramamohan Paturi, and Stefan Schneider. 0-1 integer linear programming with a linear number of constraints. *CoRR*, abs/1401.5512, 2014. URL: http://arxiv.org/abs/1401.5512.

14 Piotr Indyk, editor. *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. SIAM, 2015. doi:10.1137/1.9781611973730.

15 Ken-ichi Kawarabayashi and Paul Wollan. A shorter proof of the graph minor algorithm: the unique linkage theorem. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 687–694. ACM, 2010.

16 Jesper Nederlof. Faster subset sum via improved orthogonal vectors? http://www.win.tue.nl/ jnederlo/problem.pdf.

17 Nicolas Ollinger and Heribert Vollmer, editors. *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, volume 47 of *LIPIcs*. Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.

18 Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.

19 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *J. Comput. Syst. Sci.*, 67(4):757–771, 2003. doi:10.1016/S0022-0000(03)00078-3.

20 Marcin Pilipczuk, Michal Pilipczuk, Piotr Sankowski, and Erik Jan van Leeuwen. Network sparsification for steiner problems on planar and bounded-genus graphs. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 276–285. IEEE Computer Society, 2014. URL: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6975722, doi:10.1109/FOCS.2014.37.

21 Michal Pilipczuk and Marcin Wrochna. On space efficiency of algorithms working on structural decompositions of graphs. In Ollinger and Vollmer [17], pages 57:1–57:15. doi:10.4230/LIPIcs.STACS.2016.57.

22 Ilya P. Razenshteyn, Zhao Song, and David P. Woodruff. Weighted low rank approximations with provable guarantees. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 250–263. ACM, 2016. URL: http://dl.acm.org/citation.cfm?id=2897518, doi:10.1145/2897518.2897639.

23 Bruce Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

Participants

- Amir Abboud
Stanford University, US
- Arturs Backurs
MIT – Cambridge, US
- Andreas Björklund
Lund University, SE
- Édouard Bonnet
Middlesex University, GB
- Cornelius Brand
Universität des Saarlandes, DE
- Karl Bringmann
MPI für Informatik –
Saarbrücken, DE
- Yixin Cao
Hong Kong Polytechnic
University, CN
- Radu Curticapean
Hungarian Academy of Sciences –
Budapest, HU
- Marek Cygan
University of Warsaw, PL
- Holger Dell
Universität des Saarlandes, DE
- Fedor V. Fomin
University of Bergen, NO
- Tobias Friedrich
Hasso-Plattner-Institut –
Potsdam, DE
- Petr A. Golovach
University of Bergen, NO
- Gregory Z. Gutin
Royal Holloway University of
London, GB
- Danny Hermelin
Ben Gurion University –
Beer Sheva, IL
- Petr Hlineny
Masaryk University – Brno, CZ
- Petteri Kaski
Aalto University, FI
- Eun Jung Kim
University Paris-Dauphine, FR
- Yiannis Koutis
University of Puerto Rico –
Rio Piedras, PR
- Łukasz Kowalik
University of Warsaw, PL
- Stefan Kratsch
Universität Bonn, DE
- Bingkai Lin
National Institute of Informatics –
Tokyo, JP
- Daniel Lokshtanov
University of Bergen, NO
- Dániel Marx
Hungarian Academy of Sciences –
Budapest, HU
- Jesper Nederlof
TU Eindhoven, NL
- Fahad Panolan
University of Bergen, NO
- Christophe Paul
CNRS – Montpellier, FR
- Geevarghese Philip
Chennai Mathematical
Institute, IN
- Marcin Pilipczuk
University of Warsaw, PL
- Michal Pilipczuk
University of Warsaw, PL
- M. S. Ramanujan
TU Wien, AT
- Peter Rossmanith
RWTH Aachen, DE
- Marc Roth
Universität des Saarlandes, DE
- Saket Saurabh
The Institute of Mathematical
Sciences, India, IN
- Ildiko Schlotter
Budapest University of
Technology & Economics, HU
- Stefan Szeider
TU Wien, AT
- Dimitrios M. Thilikos
University of Athens, GR
- Magnus Wahlström
Royal Holloway University of
London, GB
- Gerhard J. Woeginger
RWTH Aachen, DE
- David P. Woodruff
IBM Almaden Center –
San Jose, US
- Meirav Zehavi
University of Bergen, NO

Report from Dagstuhl Seminar 17042

From Characters to Understanding Natural Language (C2NLU): Robust End-to-End Deep Learning for NLP

Organized by

Phil Blunsom[1], Kyunghyun Cho[2], Chris Dyer[3], and Hinrich Schütze[4]

1 **University of Oxford, GB,** `phil.blunsom@cs.ox.ac.uk`
2 **New York University, US,** `kyunghyun.cho@nyu.edu`
3 **Carnegie Mellon University - Pittsburgh, US,** `cdyer@cs.cmu.edu`
4 **LMU München, DE,** `hs2016@cislmu.org`

Edited by

Heike Adel[5] and Yadollah Yaghoobzadeh[6]

5 **LMU München, DE,** `heike@cis.lmu.de`
6 **LMU München, DE,** `yadollah@cis.lmu.de`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 17042 “From Characters to Understanding Natural Language (C2NLU): Robust End-to-End Deep Learning for NLP”. The seminar brought together researchers from different fields, including natural language processing, computational linguistics, deep learning and general machine learning. 31 participants from 22 academic and industrial institutions discussed advantages and challenges of using characters, i.e., “raw text”, as input for deep learning models instead of language-specific tokens. Eight talks provided overviews of different topics, approaches and challenges in current natural language processing research. In five working groups, the participants discussed current natural language processing/understanding topics in the context of character-based modeling, namely, morphology, machine translation, representation learning, end-to-end systems and dialogue. In most of the discussions, the need for a more detailed model analysis was pointed out. Especially for character-based input, it is important to analyze what a deep learning model is able to learn about language – about tokens, morphology or syntax in general. For an efficient and effective understanding of language, it might furthermore be beneficial to share representations learned from multiple objectives to enable the models to focus on their specific understanding task instead of needing to learn syntactic regularities of language first. Therefore, benefits and challenges of transfer learning were an important topic of the working groups as well as of the panel discussion and the final plenary discussion.

Seminar January 22–27, 2017 – http://www.dagstuhl.de/17042
1998 ACM Subject Classification I.2 Artificial Intelligence, I.2.7 Natural Language Processing
Keywords and phrases Natural Language Understanding, Artificial Intelligence, Deep Learning, Natural Language Processing, Representation Learning
Digital Object Identifier 10.4230/DagRep.7.1.129

From Characters to Understanding Natural Language (C2NLU): Robust End-to-End Deep Learning for NLP,
Dagstuhl Reports, Vol. 7, Issue 1, pp. 129–157
Editors: Phil Blunsom, Kyunghyun Cho, Chris Dyer, and Hinrich Schütze
Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Executive Summary

Phil Blunsom
Kyunghyun Cho
Chris Dyer
Hinrich Schütze
Yadollah Yaghoobzadeh

Deep learning is currently one of most active areas of research in machine learning and its applications, including natural language processing (NLP). One hallmark of deep learning is *end-to-end learning*: all parameters of a deep learning model are optimized *directly for the learning objective*; e.g., for the objective of accuracy on the binary classification task: is the input image the image of a cat? Crucially, the set of parameters that are optimized includes "first-layer" parameters that connect the raw input representation (e.g., pixels) to the first layer of internal representations of the network (e.g., edge detectors). In contrast, many other machine learning models employ hand-engineered features to take the role of these first-layer parameters.

Even though deep learning has had a number of successes in NLP, research on true end-to-end learning is just beginning to emerge. Most NLP deep learning models still start with a hand-engineered layer of representation, the level of tokens or words, i.e., the input is broken up into units by manually designed tokenization rules. Such rules often fail to capture structure both within tokens (e.g., morphology) and across multiple tokens (e.g., multi-word expressions). Given the success of end-to-end learning in other domains, it is likely that it will also be widely used in NLP to alleviate these issues and lead to great advances.

The seminar brought together researchers from deep learning, general machine learning, natural language processing and computational linguistics to develop a research agenda for the coming years. The goal was to combine recent *advances in deep learning architectures and algorithms* with *extensive domain knowledge about language* to make *true end-to-end learning for NLP* possible.

Our goals were to make progress on answering the following research questions.

- C2NLU approaches so far fall short of the state of the art in cases where token structures can easily be exploited (e.g., in well-edited newspaper text) compared to word-level approaches. What are promising avenues for developing C2NLU to match the state of the art even in these cases of text with well-defined token structures?
- Character-level models are computationally more expensive than word-level models because detecting syntactic and semantic relationships at the character-level is more expensive (even though it is potentially more robust) than at the word-level. How can we address the resulting challenges in scalability for character-level models?
- Part of the mantra of deep learning is that domain expertise is no longer necessary. Is this really true or is knowledge about the fundamental properties of language necessary for C2NLU? Even if that expertise is not needed for feature engineering, is it needed to design model architectures, tasks and training regimes?
- NLP tasks are diverse, ranging from part-of-speech tagging over sentiment analysis to question answering. For which of these problems is C2NLU a promising approach, for which not?

- More generally, what characteristics make an NLP problem amenable to be addressed using tokenization-based approaches vs. C2NLU approaches?
- What specifically can each of the two communities involved – natural language processing and deep learning – contribute to C2NLU?
- Create an NLP/deep learning roadmap for research in C2NLU over the next 5–10 years.

2 Table of Contents

Panel discussions

3 Introduction: C2NLU

Phil Blunsom (University of Oxford, GB), Kyunghyun Cho (New York University, US), Chris Dyer (Carnegie Mellon University – Pittsburgh, US), and Hinrich Schütze (LMU München, DE)

This section contains the motivation for the seminar given in the proposal submitted by the organizers in 2015 to the Schloss Dagstuhl – Leibniz-Zentrum für Informatik. It has been slightly edited.

3.1 Introduction

Deep learning is currently one of most active areas of research in machine learning and its applications, including natural language processing (NLP). One hallmark of deep learning is *end-to-end learning*: all parameters of a deep learning model are optimized *directly for the learning objective*; e.g., for the objective of accuracy on the binary classification task: is the input image the image of a cat? Crucially, the set of parameters that are optimized includes "first-layer" parameters that connect the raw input representation (e.g., pixels) to the first layer of internal representations of the network (e.g., edge detectors). In contrast, many other machine learning models employ hand-engineered features to take the role of these first-layer parameters.

Even though deep learning has had a number of successes in NLP, research on true end-to-end learning is just beginning to emerge. Most NLP deep learning models still start with a hand-engineered layer of representation, the level of tokens or words, i.e., the input is broken up into units by manually designed tokenization rules. Such rules often fail to capture structure both within tokens (e.g., morphology) and across multiple tokens (e.g., multi-word expressions). Given the success of end-to-end learning in other domains, it is likely that it will also be widely used in NLP to alleviate these issues and lead to great advances.

3.2 Goals of the Seminar

The seminar brought together researchers from deep learning, general machine learning, natural language processing and computational linguistics to develop a research agenda for the coming years. The goal was to combine recent *advances in deep learning architectures and algorithms* with *extensive domain knowledge about language* to make *true end-to-end learning for NLP* possible.

Our goals were to make progress on answering the following research questions.

- C2NLU approaches so far fall short of the state of the art in cases where token structures can easily be exploited (e.g., in well-edited newspaper text) compared to word-level approaches. What are promising avenues for developing C2NLU to match the state of the art even in these cases of text with well-defined token structures?
- Character-level models are computationally more expensive than word-level models because detecting syntactic and semantic relationships at the character-level is more expensive (even though it is potentially more robust) than at the word-level. How can we address the resulting challenges in scalability for character-level models?

- Part of the mantra of deep learning is that domain expertise is no longer necessary. Is this really true or is knowledge about the fundamental properties of language necessary for C2NLU? Even if that expertise is not needed for feature engineering, is it needed to design model architectures, tasks and training regimes?
- NLP tasks are diverse, ranging from part-of-speech tagging over sentiment analysis to question answering. For which of these problems is C2NLU a promising approach, for which not?
- More generally, what characteristics make an NLP problem amenable to be addressed using tokenization-based approaches vs. C2NLU approaches?
- What specifically can each of the two communities involved – natural language processing and deep learning – contribute to C2NLU?
- Create an NLP/deep learning roadmap for research in C2NLU over the next 5–10 years

3.3 Detailed Description of the Topic

C2NLU, i.e., for approaches to end-to-end deep learning in which either the input or the output or both are character streams.

The arguments for C2NLU we present below are closely related and can be thought of as different perspectives on the same underlying problems of token-based approaches.

3.3.1 Robustness against noise

Human natural language processing (NLP) is robust in the sense that small perturbations of the input do not affect processing negatively. Such perturbations include letter insertions, deletions, substitutions and transpositions and the insertion of spaces ("guacamole" → "gua camole") and the deletion of spaces ("ran fast" → "ranfast"). Such perturbations can cause complete failure of token-based processing. C2NLU has the potential of being robust against this type of noise.

3.3.2 Robust morphological processing

There currently does not exist an approach to morphological processing that handles both inflectional and derivational morphology and works across languages with typologically different systems of morphology. If we give up the notion that a token is an opaque symbol and instead model the sequence of characters it is made up of, then we can in principle learn all morphological regularities: inflectional and derivational regularities as well as a wide typological range of morphological processes such as vowel harmony, agglutination, reduplication and nonconcatenativity (as in Arabic and Hebrew). Further, this information can be integrated at the sequence level to permit the learning of agreement and other morpho-syntactic phenomena.

Two caveats are in order. First, it is clear that C2NLU has the potential of successful acquisition of morphology, but whether it can do so in practice is a big question. Second, adopting C2NLU does not mean that linguistically informed models will be replaced with "linguistically ignorant" models. Instead, C2NLU can be a complement to traditional morphological processing in computational linguistics, for example, in a system combination approach. C2NLU is also a promising framework for incorporating linguistic knowledge about morphology as an inductive bias into statistical models. The latter has been done with only limited success in standard statistical NLP models.

3.3.3 Orthographic productivity

It is clear that the truth is between two extreme positions: (i) the character sequence of a token is arbitrary and uninformative and (ii) the character sequence of a token perfectly and compositionally predicts its linguistic properties. Morphology is the most important phenomenon of limited predictability that lies between these two extremes. But there are many other less prominent phenomena that taken together have the potential of improving NLP models and performance of NLP systems considerably if they could be handled at the level of human competence.

- Properties of names predictable from character patterns, e.g., "Yaghoobazadeh" is identifiable as a Farsi surname, "Darnique" and "Delonda" are identifiable as names of girls (in a US context), "osinopril" is most likely a medication
- Orthographic blends and modifications of existing words, e.g., "staycation", "Obamacare", "mockumentary", "dramedy", "cremains"
- Non-morphological orthographic productivity in certain registers, domains and genres: character repetition in tweets ("cooooooooooool"), shm-reduplication ("fancy-shmancy"), the pseudo-derivational suffix "-gate" signifying "scandal" ("Watergate", "Irangate", "Dieselgate")
- Sound symbolism, phonesthemes, e.g., "gl-" ("glitter", "gleam", "glint", "glisten", "glow")
- Onomatopoeia, e.g., "oink", "sizzle", "tick tock"

3.3.4 OOV analysis

One big advantage of character models is that the problem of out-of-vocabulary (OOV) words disappears.[1] Of course, if the NLP system encounters a character string that was never observed before and that would be opaque even to a human reader, then the situation is not better than it would be for a token-based model encountering an OOV. However, as discussed above, in many cases a great deal can be predicted from the character string of an OOV.

So this argument – character models are a promising approach to OOV analysis – is a summary of the last three arguments. Character models are more robust against noise (which may lead to OOVs), have the potential for more robust morphological processing (many OOVs are due to inflection and derivation) and can handle orthographic productivity better ("Yaghoobazadeh", "osinopril" and "staycation" are relevant examples for words that a token-based system may not have observed in the training set).

3.3.5 OOV generation

There is currently no principled and general way for token-based end-to-end systems to generate tokens that are not part of the training vocabulary. Since a token is represented as a vocabulary index and parameters governing system behavior affecting this token are referring to this vocabulary index, a token that does not have a vocabulary index cannot easily be generated in end-to-end systems.

One application in which this is a critical problem is the generation of names in end-to-end machine translation. In the simplest case, a name like "Obama" must be copied from source sentence to target sentence. More complicated cases involve transliteration (English "Putin"

[1] However, there will occasionally be out-of-alphabet characters if special cases like emoji and rare diacritics occur in the input.

becomes French "Poutine") and number variation ("4.12 million" may become "4 million" in a summary). In token-based systems, these cases are often handled separately with external mechanisms, such as an external transliteration system coupled with named-entity detection. However, this is inherently limited as it requires a separate system, which is not jointly tuned together with the main system, for each and every special case.

Character-based systems do not have a problem with OOV generation in principle. However, as with other potential advantages, the problem of practical feasibility arises: specific proposals as to how C2NLU systems can learn to accurately generate OOVs are needed.

3.3.6 Tokenization-free models

One of the drawbacks of token-based models is that they usually tokenize text early on and it is difficult to correct these early tokenization decisions later on. While it is theoretically possible to generate all possible tokenizations and pass any tokenization ambiguity through the entire NLP pipeline (e.g., by using lattices), this is inefficient and often incompatible with the requirements of subsequent processing modules. For this reason, text-to-text machine translation systems usually only consider a single tokenization of source and target.

Tokenization causes limited damage in English although even in English there are difficult cases like "Yahoo!", "San Francisco-Los Angeles flights", "The Iowa campaign manager was selected 'The Apprentice'-style." and hashtags like "#starwars". In other languages, tokenization is even more problematic. In Chinese, tokens are not separated by spaces or other typesetting conventions. For most NLP applications, German compounds should be split. Tokens in agglutinative languages like Turkish are difficult to process as unanalyzed symbols.

3.3.7 Direct models of the data

Traditional machine learning and especially statistical natural language processing rely heavily on feature engineering. In contrast, the philosophy of deep learning is to use any knowledge about the domain for careful design of the architecture of models and for task definitions and training regimes that optimally exploit available data. This careful modeling should then obviate the need for manual feature engineering.

One motivation for this approach is the view that a good set of features can best be found by training a well designed model in a well designed experimental setup. In contrast, manual feature design is prone to errors and omissions.

The success of deep learning in speech, vision and machine translation demonstrates the potential of giving models direct access to the data as opposed to through the intermediary of human-designed features.

However, in natural language processing – and even in neural machine translation – there is still one set of manually designed features that is almost universally used: tokens. Given the success of the raw-data approach for other applications, an approach that takes it to the extreme – character-based models – seems worth investigating for natural language processing. Character-based models are the most faithful to deep learning philosophy: they model the data as it comes in without any alteration through manually designed features. It is an open question whether character-based models will ultimately turn out to be superior to token-based models, but it is an important question that we want to investigate in this workshop.

4 Overview of Talks

4.1 C2NLU: An Overview

Heike Adel (LMU München, DE)

Natural language processing (NLP) and natural language understanding (NLU) often build on a pipeline of different preprocessing modules, such as tokenization, sentence segmentation as well as syntactic and semantic analysis. This pipeline is prone to subsequent errors. However, recovering from those errors is often impossible or not feasible. Alternatively, NLU models could directly model the raw input data, i.e., sequences of characters. Especially with deep learning models, it is possible that the models learn their own representation of the data without the need of tokenizing the character sequence into words. In fact, character-based features have a long history in NLP/NLU, especially in information retrieval, grapheme-to-phoneme conversion or language identification. Recently, more and more studies use characters as input to (hierarchical) neural networks.

This talk provides an overview of character-based NLP/NLU systems. It motivates the usage of characters as features for machine learning systems and/or as input to deep learning models. Furthermore, it presents existing studies on character-based NLU, clustering them into three categories: tokenization-based models, bag-of-n-gram models and end-to-end models.

4.2 Should Model Architecture Reflect Linguistic Structure?

Chris Dyer (Carnegie Mellon University – Pittsburgh, US)

Sequential recurrent neural networks (RNNs) over finite alphabets are remarkably effective models of natural language. RNNs now obtain language modeling results that substantially improve over long-standing state-of-the-art baselines, as well as in various conditional language modeling tasks such as machine translation, image caption generation, and dialogue generation. Despite these impressive results, such models are a priori inappropriate models of language. One point of criticism is that language users create and understand new words all the time, challenging the finite vocabulary assumption. A second is that relationships among words are computed in terms of latent nested structures rather than sequential surface order (Chomsky, 1957; Everaert, Huybregts, Chomsky, Berwick, and Bolhuis, 2015).

In this talk I discuss two models that explore the hypothesis that more (a priori) appropriate models of language will lead to better performance on real-world language processing tasks. The first composes sub word units (bytes, characters, or morphemes) into lexical representations, enabling more naturalistic interpretation and generation of novel word forms. The second, which we call recurrent neural network grammars (RNNGs), is a new generative model of sentences that explicitly models nested, hierarchical relationships among words and phrases. RNNGs operate via a recursive syntactic process reminiscent of probabilistic context-free grammar generation, but decisions are parameterized using RNNs that condition on the entire (top-down, left-to-right) syntactic derivation history, greatly relaxing context-free independence assumptions. Experimental results show that RNNGs obtain better results in generating language than models that don't exploit linguistic structures.

4.3 From Bayes Decision Rule to Neural Networks for Human Language Technology (HLT)

Hermann Ney (RWTH Aachen, DE)

The last 40 years have seen dramatic progress in machine learning and statistical methods for speech and language processing like speech recognition, handwriting recognition and machine translation. Most of the key statistical concepts were originally developed for speech recognition. Examples of such key concepts are the Bayes decision rule for minimum error rate and probabilistic approaches to acoustic modeling (e.g., hidden Markov models) and language modeling. Recently, the performance of HLT systems for speech recognition were improved significantly by the use of artificial neural networks, such as deep feedforward multi-layer perceptrons and recurrent neural networks (including long short-term memory extension). We will discuss these approaches in detail and how they fit into the probabilistic approach to HLT.

4.4 Cross-Token Character N-Gram Modeling: The Other Shoe To Drop?

Hinrich Schütze (LMU München, DE) and Yadollah Yaghoobzadeh (LMU München, DE)

Representation learning in NLP is usually performed on the level of tokens, which requires segmentation or tokenization of input sentences. Here, we introduce an alternative that is completely nonsymbolic: random segmentation of the input (within-token or cross-token) into character n-grams. This applies to training the parameters of the model on a training corpus as well as to applying it when computing the representation of a new text. We give linguistic motivation as well as reporting experimental results, hypothesizing that this way of representation learning could be effective and overcome shortcomings of other methods.

4.5 Modeling Multiple Sequences: Explorations, Consequences and Challenges

Orhan Firat (Middle East Technical University – Ankara, TR)

Deep (recurrent) neural networks have been shown to successfully learn complex mappings between arbitrary length input and output sequences in varying input granularity, such as words, sub-words, characters or even unicode bytes. We investigate extensions of this effective framework, encoder-decoder networks, that handle multiple sequences at the same time. This reduces to the problem of multi-lingual machine translation (MLNMT), as we explore and discuss the applicability and benefits of using finer tokens. Further we discuss future directions that are enabled by using finer tokens, from multi-modal processing and multi-task learning perspectives. We finally discuss observed problems and future challenges for multi-sequence modeling with finer tokens.

4.6 Robsut Wrod Reocginiton via semi-Character Recurrent Neural Network

Kevin Duh (Johns Hopkins University – Baltimore, US)

The Cmabrigde Uinervtisy (Cambridge University) effect from the psycholinguistics literature has demonstrated a robust word processing mechanism in humans, where jumbled words (e.g., Cmabrigde / Cambridge) are recognized with little cost. Inspired by the findings from the Cmabrigde Uinervtisy effect, we propose a word recognition model based on a semi-character level recursive neural network (scRNN). In our experiments, we demonstrate that scRNN has significantly more robust performance in word spelling correction (i.e., word recognition) compared to existing spelling checkers. Furthermore, we demonstrate that the model is cognitively plausible by replicating a psycholinguistics experiment about human reading difficulty using our model. (This is joint work with Keisukue Sakaguchi, Matt Post, and Ben Van Durme)

4.7 Inducing Morpho-syntactic Lexicons and Morphological Inflections

Manaal Faruqui (Carnegie Mellon University – Pittsburgh, US)

Morphology of a word can help determine different aspects of its meaning such as tense, mood, voice, aspect, person, gender, number and case. Such morpho-syntactic information about word meaning provides crucial information while training models for downstream NLP tasks. In this talk we are going to discuss two different problems involving morphology. In the first part of the talk I will show how morphological information can be used to construct large-scale morpho-syntactic lexicons for a large number of languages. In the second part of the talk I will show how different possible inflected forms of a word can be generated using encoder-decoder neural network models in a language-independent manner.

4.8 (Neural) Graphical Models Over Strings

Ryan Cotterell (Johns Hopkins University – Baltimore, US)

Natural language processing must sometimes consider the internal structure of words, e.g., in order to understand or generate an unfamiliar word. Unfamiliar words are systematically related to familiar ones due to linguistic processes such as morphology, phonology, abbreviation, copying error, and historical change. We will show how to build joint probability models over many strings. These models are capable of predicting unobserved strings, or predicting the relationships among observed strings. However, computing the predictions of these models can be computationally hard. We outline approximate algorithms based on Markov chain Monte Carlo, expectation propagation, and dual decomposition. We give results on some NLP tasks.

5 Working groups

5.1 Morphology

Kristina Toutanova (Google – Seattle, US)

Joint work of Fabienne Cap, Ryan Cotterell, Chris Dyer, Kevin Duh, Manaal Faruqui, Vladimir Golkov, Adam Lopez, Laura Rimmel, Kristina Toutanova, François Yvon

5.1.1 Introduction

This is a report that summarizes the discussions and project ideas that originated in the morphology working group. Our group had productive discussions on several topics that deal with character-level and subword-level modeling of natural language.

The discussion started with "what is morphology?" or rather what is a universally accepted theory of how words are formed. Most people agreed that probably at some level there exists a morpheme and then inflection generation on the morpheme leads to the creation of new words. However, we did not delve deep into whether this is correct theory or not, as we wanted to discuss more practical problems.

A common agreement in the team was that derivational morphology is often ignored in modeling, even though around 50% of words in English are constructed through derivational processes. An example of this phenomenon is: *unquaffability*, which is composed of *un* + *quaff* + *able* + *ity*. In derivational morphology since the semantic meaning of the word changes, it is more important to have segmentation of the word that can reveal its composition. Derivational morphology is also different from inflectional morphology in the sense that we often need to extract knowledge from the sentence context to be able to analyze the word.

An advantage of character-level models over word or morpheme-level models is that it allows the generation and analysis of out-of-vocabulary words. Using character-level models of course raises the question whether morphology is useful at all practically or it is something that can be ignored. To people whose interest lies in understanding how language works, this is an interesting question. But for those who only care about whether information within a word can help NLP systems, just proceeding with a character-level model seems like the best route.

5.1.2 Research Questions

Now we provide a list of different topics that we came up with for future research. We will describe one particular model in detail in Section 5.1.4.

- **RQ1:** Is there a relation between traditional morphology and character-level representations?
- **RQ2:** Does morphological modeling help downstream tasks?
- **RQ3:** How do we incorporate morphology in neural machine translation?
- **RQ4:** How do we know if the characters of segments in a word are compositional?
- **RQ5:** How do we analyze what segments/characters of the word is the model selecting?
- **RQ6:** How does character-level NLP handle compounding and multi-word expressions?
- **RQ7:** How does character-level NLP handle non-standard morphological processes (e.g., word analogies, phonesthemes (glow-glitter-glide), apocope, portmanteau words (Oxbridge, Bennifer, Brexit), diminutives, etc.
- **RQ8:** Given a word form, can we predict its morpho-syntactic properties?
- **RQ9:** Can we use data with perturbed character sequences within a word to generate more training data and improve model performance?
- **RQ10:** Do character and segment-level models learn the exact same thing?

5.1.3 Analyzing distributed representations

Character-based models deliver distributed representations for words, which are varying dependent on the context of their occurrences. An interesting issue is to try and diagnose the kind of morphological knowledge that is (or not) encoded in these representations.

Expectations regarding these representations are that they should for instance encode:

- Ambiguity (in inflection and derivation): This is important because ambiguity is pervasive in NLP, and can sometimes have some systematic (i.e., in conjugation) nature. Therefore, looking at how ambiguity is encoded in distributional representations has a bearing on what we can expect from continuous representations.
- Another issue is with exceptional patterns: if we look at string patterns, we see regular and exceptional / coincidental patterns, e.g., dearly is dear+ly but early is not ear+ly. Can we detect regularity vs. exceptions in distributed representations? In a similar fashion, can we detect transparent vs. opaque derivatives based on these representations?

In order to evaluate these models and representations, a set of systematic tests, applicable to a wide variety of languages and morphological systems needs to be designed. Possible tasks, along the lines of the recent work by [1, 2] are the following:

- Predict the correct agreement / case markings, for increasing complex (remote) dependencies. This could be implemented in a number of ways, such as compare the likelihood of correct vs. incorrect forms.

  Another interesting test would a be akin to the “wug” test. Given a prefix sentence and the prefix of a word, see whether the model can generate the right suffix. For instance:

  > I think that writing this short example really wug?

  If the model can also generate POS tags, we can make the task more interesting by also providing POS information (in the former example how to generate given .... (past,wug)). This kind of exercise should be designed for more inflectionally interesting languages than English.

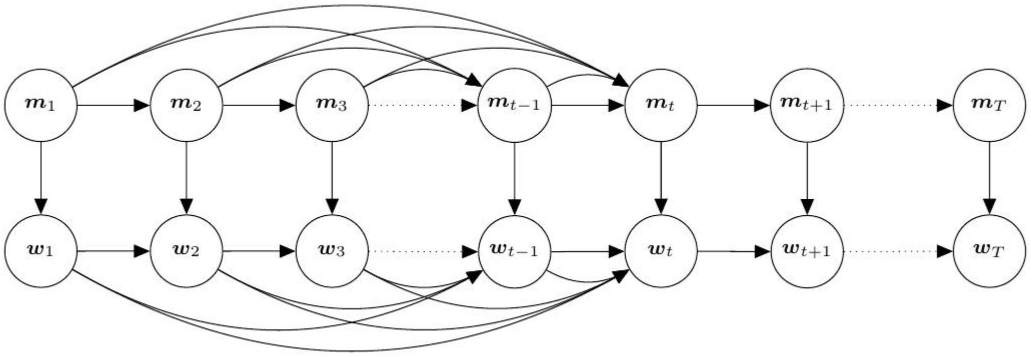

Figure 1 A graphical represention of the joint model.

- Testing derivational knowledge is more challenging. A possible source of inspiration is the definition generation task, which provides a way to explicitly stimulate lexical creativity. Alternatively, one could also try sentence completion exercises with contexts that should both constrain a given POS and a given derivative (if it exists)

  > This model is really [Markovian / *Markovable / Markovist / *?Markoving / *Markovability / Markovesque / *Markovism] etc.

  and test phenomena such as morphotactics, morpho-phonological processes (or rather their orthographical expression), such as e.g., vowel harmony, vowel reduction, consonant assimilation, etc).

As regards the evaluation of these representations, various ideas have popped up, such as:

- Correlate the distribution of representation of a word with the number of morphological analyses;
- Evaluate the ability of the representations to identify derivational families;
- Contrast character-based vs. word-based embeddings and see whether they can be made to agree[2]

The next step, if these properties are not satisfied and actually matter for downstream applications, is to try to enforce these properties by jointly learning word sequences and their morphology. Such ideas are developed in the following section.

5.1.4 Proposed Model

In this section, we sketch out a language model that jointly models words ($\vec{w}$) and morphological features ($\vec{m}$). Intuitively, each token has a set of morphological features, and the generative process first selects the morphological features at time t and then, conditional on that and the history of words, selects a word. A graphical representation is in Figure 1.

[2] This might be useful in the following setting: (1) learn word based embeddings for frequent words; (2) train a character-based model to reproduced these; (3) use the resulting character embeddings to generate embeddings for unknown words. This could also be used as a way to speed up the training of character-level embeddings.

$$
\begin{aligned}
p(\vec{w}) &= \sum_{\boldsymbol{m}} p(\boldsymbol{w}, \boldsymbol{m}) \\
p(\boldsymbol{w}, \boldsymbol{m}) &= \prod_{t=1}^{N} p(w_t, m_t \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) \\
p(\vec{w}) &= \sum_{\vec{m}} \prod_{i=1}^{N} p(w_t, m_t \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) \\
p(w_t, m_t \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) &= p(m_t \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) \times p(w_t \mid \boldsymbol{w}_{<t}, m_t) \qquad (1) \\
&= p(m_t \mid \boldsymbol{m}_{<t}) \times p(w_t \mid \boldsymbol{w}_{<t}, m_t) \qquad (2) \\
&= p(m_t \mid m_{t-1}) \times p(w_t \mid \boldsymbol{w}_{<t}, m_t) \qquad (3) \\
&= p(m_t \mid \boldsymbol{w}_{<t}, m_{t-1}) \times p(w_t \mid \boldsymbol{w}_{<t}, m_t) \qquad (4)
\end{aligned}
$$

In Equations (1) and (2), supervised training is tractable, but marginalizing $\boldsymbol{m}$ is intractable. However, the models given by Equations (3) and (4) are reasonable and the forward algorithm can be used to marginalize $\boldsymbol{m}$ which can be used for likelihood evaluation or unsupervised training.

$$
\begin{aligned}
p(w_t = k \mid \boldsymbol{w}_{<t}, m_t) &= \mathrm{softmax}_k \left(\mathbf{U} \tanh \left[\boldsymbol{\varphi}_{m_t}; \mathbf{h}^w_{t-1}\right]\right) \\
p(m_t = \ell \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) &= \mathrm{softmax}_\ell \left(\mathbf{V} \tanh \left[\mathbf{h}^m_{t-1}; \mathbf{h}^w_{t-1}\right]\right)
\end{aligned}
$$

Possible extensions of the baselines:

1. generate words with a character model (instead of the softmax layer). It would be especially interesting to see whether the additional morphological “tier” helps to improve the morphological tasks;
2. add an attention model over past words / past morphs / both;

$$
\begin{aligned}
p(w_t = k \mid \boldsymbol{w}_{<t}, m_t) &= \mathrm{softmax}_k \left(\mathbf{U} \tanh \left(\boldsymbol{\varphi}_{m_t} + \sum_{i=1}^{t-1} \beta^m_i \mathbf{h}^w_i\right)\right) \\
p(m_t = \ell \mid \boldsymbol{w}_{<t}, \boldsymbol{m}_{<t}) &= \mathrm{softmax}_\ell \left(\mathbf{V} \tanh \left(\sum_{i=1}^{t-1} (\alpha^m_i \mathbf{h}^m_i + \beta^m_i \mathbf{h}^w_i)\right)\right)
\end{aligned}
$$

3. use this as an additional model on the target side for MT tasks.

5.1.5 Comparing Characters and Words

Character-level modeling allows sharing of statistical strength among words that have common character sequences. Word-level modeling integrates an inductive bias towards architectures that construct and reason with representations of words. Prior work has built architectures using these different units of modeling, but has not isolated their impact in controlled comparative studies.

We propose several experiments to evaluate the capacity and practical performance of character and word-based architectures. A simple experiment is to start with two architectures for word representations used in a language modeling task for a simple n-gram feed-forward language model. Architecture W embeds all words using a one-hot representation of words and a second architecture C uses a bi-directional character-level RNN with one or more layers to derive a word embedding as the last layer.

Given a model trained on a given dataset D using architecture W and word embedding size d, theory suggests that an architecture of type C exists that can obtain the same optimal value of the training loss function. Theory does not provide guidance on the dimensionality or number of hidden layers in C and similarly, nothing is known about the training time and difficulty of the two optimization problems.

We suggest to experiment with architectures of growing size in C, to discover patterns in the relationship between dimensionality d in W and dimensionality and architectures in C. We will experiment with multiple layers and hidden unit size in C, a final layer mapping to dimensionality d. CNN in addition to RNN architectures will be interesting to evaluate. For every candidate architecture, we will measure the loss function value obtained, as well as dimensionalities, time, and learning curve.

An alternative to training character-level architectures from scratch is to first train a word-level architecture and then train a character level sub-model which for each word w aims to fit its word embedding v via the character-level sub-network. In theory, this approach would result in a similar value of the loss function (if the word embeddings can be approximated very well), but would be much faster to train since type-level as opposed to token-level updates will be required in training. The authors of [3] have already shown that character-level models are able to fit word vectors well, but have not studied the relative dimensionality and efficiency of the two representations.

References

1 Tal Linzen, Emmanuel Dupoux, Yoav Goldberg: Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. TACL 2016.

2 Rico Sennrich: How Grammatical is Character-level Neural Machine Translation? Assessing MT Quality with Contrastive Translation Pairs. EACL 2017.

3 Ryan Cotterell, Hinrich Schütze, Jason Eisner: Morphological Smoothing and Extrapolation of Word Embeddings. ACL 2016.

5.2 Machine Translation

Parnia Bahar (RWTH Aachen, DE)

Joint work of Parnia Bahar, Ondrej Bojar, Desmond Elliot, Orhan Firat, Alex Fraser, Georg Heigold, Jan Niehues, Laura Rimell, Helmut Schmid

Machine translation (MT) is perhaps the biggest success of deep learning in NLP. This working group was concerned with research questions and challenges for character-level MT. In particular, the working group dealt with three issues in Neural Machine Translation (NMT) [1, 2]: appropriate units for an NMT system; efficiency of NMT with different units; and how to treat multilinguality and multiple modalities.

5.2.1 Appropriate Units for NMT

Since the vocabulary size needs to be shortened to the most frequent tokens in word-level NMT, it suffers from out-of-vocabulary (OOV) problems. Possible solutions are subword units or characters as the atomic units of translation.

Byte Pair Encoding (BPE) [3] is a prominent approach which segments words into sequences of subwords. The most frequent words are kept in their original forms while the rare words are divided into pieces. This appproach is more effective than using words. However, it requires purely offline segmentation in advance and the number of merges needs to be predefined. An alternative idea would be to learn the merge operations inside the network for machine translation. Another solution which does not require explicit segmentation or word boundaries is character-level NMT.

Although using character-level NMT reduces the computational complexity in the softmax operation and can handle the OOV problem, several requirements and challenges need to be considered. If the source and target sentences are encoded in characters, the sequence length is longer. Hence, for each target token, the decoder needs to attend to all characters in the source sequence. This can lead to an extreme computational time in the attention layer. One solution to overcome this problem is to have a shorter source representation. The authors of [4], for example, propose a convolutional neural network with different filter sizes and max-pooling to have a shorter representation in the encoder. In particular, a representation for every five character positions is created and then forwarded to a highway network and the encoder, which is a bidirectional gated recurrent neural network. Thus, the network calculates features on character n-grams including cross-token n-grams. The working group agreed that it should be possible to get slightly better results without max pooling if there is enough training data and time available.

Character-based models provide advantages for multilingual models. However, they require a shared idea of characters. It is, for example, not clear if the model of [4] would work for both German and Chinese input. Coming up with a good BPE inventory for multilingual input/output is in general difficult.

5.2.2 Efficiency of NMT with Different Units

An advantage of character-based models is the possibility of attending to characters and, therefore, implicitly splitting compounds or words from agglutinative languages. However besides translation quality, efficiency of training and memory plays an important role. Based on experimental results reported in the literature, character-level NMT is three times slower than BPE-NMT. Therefore, utilizing larger strides in max pooling, reducing the number of computations of attention weights, faster convergence by means of momentum-based optimization algorithms and finally reducing the precision could speed up training and make memory usage more efficient. The decoder can be sped up by recomputing attention only after every three characters or by only computing attention for a few relevant positions. For this purpose, the working group proposed to run a recurrent neural network over the input embeddings to compute a relevance score (sigmoid output) for each position. Closely located high relevance scores are penalized (for example, by adding a sum of pairwise products of scores to the loss function). This additional constraint should force the model to identify a small set of highly relevant positional embeddings. The machine translation decoder can then only attend on the relevant positions. Having only a small number of relevant positions speeds up the decoder attention considerably.

5.2.3 Multilinguality and Multi-Modality

In the last session, the working group discussed how to use two input modalities, such as a character-level text and an image. An example for image-to-text decoding is the image captioning task. Because sharing information between character-level text, like the plural morpheme "+s" and a picture of two cats is difficult, the group first considered two-source character-level translation as in multilingual systems. The BLEU score of an NMT system which is trained on two sources (French and Spanish) for translating into English is considerably higher than the BLEU scores of two single-source systems. The working group assumed that this implies that the NMT systems learns a shared representation (something like an interlingua). Alternatively, it might just be an ensembling effect of two co-existing systems, leading to improvements because more target side data was seen effectively. To test whether anything is shared or not, the working group created two setups. The first one is to train two separate NMT systems: French→English and Spanish→English on the same set of English sentences and apply multi-source translation at test time (by ensemble decoding). In the second setup, there are two encoders (one for French and one for Spanish) and one shared decoder for English, trained by alternating samples. If the second scenario works better than the first one, there is some sharing in the joint decoder. This would be beneficial, because it means that one can train two separate encoders (for different modalities, such as character-level text and pictures) with a single shared output decoder. For such a multi-modal system, the same test could be applied to figure out whether there is sharing between the two modalities. An alternative approach for training shared representations is an adversarial setup which includes a second objective in order to make it hard to tell from which encoder the representation came.

5.2.4 Research Questions

The working group came up with the following research directions:

- **RQ1:** What is a proper translation unit in terms of performance, efficiency and availability of reasonable computational resources?
  - Possible units: characters, BPE, subwords, words, phrases, ...
  - Investigation of BPE (offline segmentation) vs. character-based (online segmentation) models.
    The space of possible offline segmentations is huge.
  - Reduction of BPE size towards characters to get a better generalization.
    Sennrich, for example, showed that a 2-gram character consecutive segmentation plus a word short list work well. [3]
  - For efficiency, it might be better to have a hybrid representation instead of a fully character-based one.
    An important question is what sort of hybrid representations are possible. A possible hybrid would be BPE + characters in an encoder-decoder network with two attention layers.
  - Comparison of character-based word representations and extra-token character-n-gram representations
  - Is it possible to predict bytes or even bits?
  - Investigation of multiple encoders at different levels (on words, BPE, characters).
    A two-level attention mechanism might be used which first decides which encoder to use. Another possibility is the concatenation of the results of the different encoders. Techniques from multi-source translations might be employed as well.
- **RQ2:** How can we inspect models?
  - An interesting analysis would be the comparison of attention at character level and attention at segment level.
  - Is it possible to use attention on character-to-character models to get BPE-like units (for evaluating character-to-character models)?
- **RQ3:** Can we use large amounts of monolingual source data to improve the encoder?
  One idea is to have multiple decoders with a shared encoder in a multi-task setting.

- **RQ4:** How to investigate whether there is sharing between a character-level text encoder and a picture encoder for image caption translation?
  Is there a good representation like the red box proposed by the representation-learning working group?

References

1 Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio: Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015.

2 Ilya Sutskever, Oriol Vinyals, Quoc V. Le: Sequence to Sequence Learning with Neural Networks. NIPS 2014.

3 Rico Sennrich, Barry Haddow, Alexandra Birch: Neural Machine Translation of Rare Words with Subword Units. ACL 2016.

4 Jason Lee, Kyunghyun Cho, Thomas Hofmann: Fully Character-Level Neural Machine Translation without Explicit Segmentation. arXiv 2016.

5 Junyoung Chung, Kyunghyun Cho, Yoshua Bengio: A character-level decoder without explicit segmentation for neural machine translation. ACL 2016.

6 Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, Koray Kavukcuoglu: Neural Machine Translation in Linear Time. arXiv 2017.

7 Jonas Gehring, Michael Auli, David Grangier, Yann N. Dauphin: A Convolutional Encoder Model for Neural Machine Translation. ICLR 2017.

5.3 Representation Learning

Yadollah Yaghoobzadeh (LMU München, DE)

Joint work of Heike Adel, Phil Blunsom, Fabienne Cap, Vera Demberg, Chris Dyer, Jan Hajicč, Karl-Moritz Hermann, Adam Lopez, Sien Moens, Martin Schmitt, Hinrich Schütze, Yadollah Yaghoobzadeh

5.3.1 Introduction

Starting from scratch and doing end-to-end learning for each individual task is not effective. In end-to-end NLP, we need to start from characters/bytes and therefore lots of training data is required. However, sparsity is always a reality. Therefore, we need to somehow transfer our knowledge (specifically, the generic knowledge) from one task to others. Representation learning is one way to do so: we learn a generic representation of the input and use that in the end tasks. We argue that segmentation of inputs to the sequence of characters/bytes makes representation learning ineffective.

A representation should contain all general linguistic knowledge, including morphological, syntactic, semantic, sound/phonetic, orthographic, distributional and discourse knowledge.[3] Then, for the specific NLP tasks, mappings are learned on top of these general representations. The advantage of representations is that less training data is necessary for a target task. However, there are also some caveats, e.g., how to fit coreference resolution and discourse dependencies, even though these require general linguistic knowledge.

There are some important (unanswered) questions about representation learning in NLP: (i) What is the level of granularity? (From morphemes to idioms). (ii) Are overlapping units

[3] The possibility of extraction of world knowledge from text is limited; e.g., "black sheep" is far more common than "white sheep" in text.

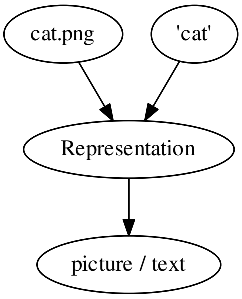

Figure 2 Multimodal representation.

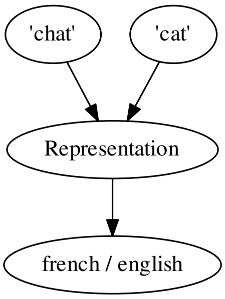

Figure 3 Multilingual representation.

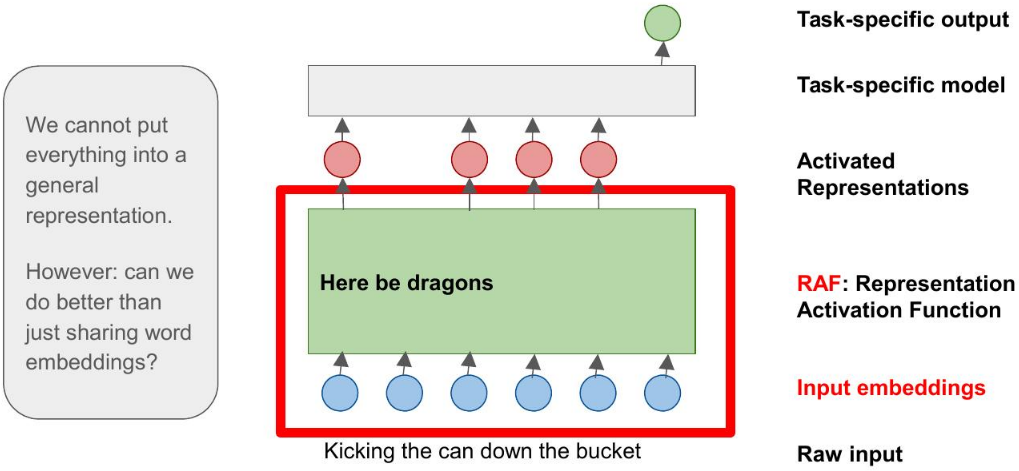
We cannot put everything into a general representation.
However: can we do better than just sharing word embeddings?
Here be dragons
Kicking the can down the bucket
Task-specific output
Task-specific model
Activated Representations
RAF: Representation Activation Function
Input embeddings
Raw input

Figure 4 Schematic diagram of The Red-Box.

fine? Example: "seven", "heaven" and "seventh heaven". (iii) What should be the content of the representations? Example types of content: phonetic similarity, visual similarity (radicals, strokes, pixels, bits), Affordances, physical properties, (iv) Are the contents of representations based on the target use case? (v) Should we use better learning objectives for representations? (vi) How can we learn disentangled representations? (vii) How to trade off memorization against generalization? (Saussure vs. Frege).

5.3.2 The Red-Box

The motivation and schematic diagram of the model we proposed for representation learning is shown in Figure 4.

There are some key considerations in modeling the Red-Box: (i) Activated representations (AR) = standard embeddings + some context + some syntax + some semantics. (ii) The aim is to facilitate learning task-specific models. (iii) Is the representation activation function (RAF) equivalent to the NLP pipeline? (iv) Training: multi-task setup with joint core up to AR. (v) RAF evaluation objective: minimize sample complexity for downstream tasks. (vi) Duplication / Parameter Sharing: "In seventh heaven" is a unit that we can have an embedding for. But to understand "the highest of seven heavens according to Islam

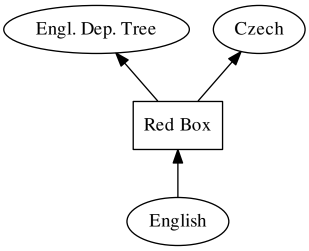

Figure 5 Example tasks for the Red-Box: dependency parsing and machine translation.

and Judaism" we need access to the notion of seventh heaven downstream. In traditional approaches, all relevant info is in one place (the lexicon). In contrast in the Red-Box model, knowledge is distributed over two places: embeddings & RAF. How is knowledge shared between embeddings and RAF?

5.3.3 Research Questions

Apart from theoretical considerations, there are also some practical challenges and research questions that need to be addressed:

- **RQ1: How to train the Red-Box?** (i) on which tasks? (ii) with which kind of weighting regime and (iii) how to avoid catastrophic forgetting?
  We need to choose the tasks that we want to train the Red-Box on. Should these tasks be multi-modal, or just text-based?
  - Example tasks A (see Figure 5). Task 1: dependency parsing, an English sentence as input and the English dependency parse of the sentence as output. Task 2: machine translation, an English sentence as input and the Czech translation as output. The Red-Box possibly benefits in this scenario as follows: Task 1 helps identify "grandma" as object of "take care" in "he took care of grandma". Task 2 (transfer task) can then learn more easily that "grandma" needs to be generated in the accusative case.
  - Example tasks B: Task 1: machine translation, a Finnish sentence as input and the English translation as output. Task 2 (transfer task): coreference resolution, a Finnish sentence as input and the resolved coreference chains as output. The Red-Box possibly benefits in this scenario as follows: Task 1 should learn coreference resolution since it is required for correct Finnish-to-English translation. Task 2 just has to tap into what Task 1 has already learned.

  To avoid catastrophic forgetting, we might need to include some sort of memory in the Red-Box – Human memory prevents forgetting.
- **RQ2: What information should be in the output of the Red-Box?** How to draw the line between core linguistic competence and non-linguistic information? What about: common-sense knowledge, world knowledge, logical inference? And: are vectors going to be the representation of outputs? Vectors are the "universal language" of deep learning, so an easy combination of modules is possible with them. However, they are hard to interpret.
- **RQ3: How can the Red-Box and its output be interpreted?** There are two questions in this regard: (i) Which parts of the sentence are responsible for a particular output? (ii) What is the interpretation proper of the output? There are some possible ways: (i) Nearest neighbors; (ii) Attention; (iii) Train special "task" modules that produce an

interpretable output; (iv) If we train the Red-Box on machine translation, how do we find out what it has learned? Maybe it has not learned anything interesting?

- **RQ4: For which tasks can the Red-Box be used?** There are more open research questions that need to be discussed: (i) Which tasks can a Red-Box trained on machine translation perform well for? Question answering, textual entailment, inference? (ii) Which combination of tasks should the Red-Box be trained on, so that it is usable across a broad range of applications? (similar to the classical NLP pipeline).

5.4 End-to-End

Heike Adel (LMU München, DE)

Joint work of Heike Adel, Jan Hajič, Thomas Hofmann, Hang Li, Hermann Ney, Ngoc Thang Vu

5.4.1 Introduction

There are several interpretations of the term “end-to-end”, such as “trainability of the system as a whole”, “joint training of all system components” or “training without explicit feature design”. While the term “end-to-end” is probably young, the concept of end-to-end learning is more than 20 years old although back then people refered to it rather as a joint learning of modules. While the computational complexity of such a system was too high in the past, current studies are getting closer to building “real” end-to-end systems.

5.4.2 What are the “Two Ends”?

One characteristic of end-to-end systems is that the whole system is trained as a whole – from the processing of the input to the prediction of the output.

The typical input to end-to-end systems is raw data. While in speech, this corresponds to the acoustic signal or spectral features, and computer vision uses the pixels of an image, the question in NLP is whether to use words, characters or bytes. Mixtures of those are also possible. The working group agreed that using pre-trained embeddings to represent the input does not contradict the notion of end-to-end. One particular possibility is the joint (hybrid) training of embeddings and the final task. Especially due to continuous vector representations in neural network and end-to-end training, it is also possible to create representations for inputs of different modalities in the same continuous space. However, this increases the complexity of the model. Slightly different to multi-modal is cross-modal training. In this case, the modality of the input is different to the modality of the output. An example is the generation of an output in a different modality (such as image captioning). One research question in this area is how to integrate a prior of the target modality into the end-to-end model (such as language model probabilities for image captioning).

For training end-to-end models, the correct choice of the objective function is crucial. The objective function should be related to the end performance / final application to be able to train every system component jointly. However, the end performance might be hard to express mathematically or might not be differentiable. An example is a machine translation system that should be used by human beings. The reaction of humans to the system and its output can differ a lot from typical machine translation metrics, such as the BLEU score. In training end-to-end systems, there might also be multiple objectives. For example, an intermediate layer could be trained to predict part-of-speech (POS) tags

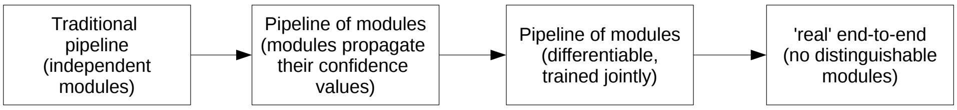
Traditional pipeline (independent modules)
Pipeline of modules (modules propagate their confidence values)
Pipeline of modules (differentiable, trained jointly)
'real' end-to-end (no distinguishable modules)

Figure 6 Evolution from pipeline systems to end-to-end systems.

using a side-objective in order to bias the model to learn POS tag-like features. This allows integrating linguistic or domain knowledge in a differentiable way into an end-to-end system. It can improve the final performance and lead to a faster convergence of the system.

5.4.3 End-to-End Systems

End-to-end learning refers to the joint training of individual system components. There is an important tradeoff between end-to-end training and model interpretability. Between traditional pipeline systems and "real" end-to-end systems, there are other hybrid possibilities. Figure 6 shows the evolution from pipeline systems to end-to-end systems. In traditional pipeline systems, each system component (module) is independent from the other modules. In a first hybrid step, the pipeline modules are still independent but propagate their confidence values to the following modules. Between these types of systems and "real" end-to-end systems which might not have distinguishable modules any more, there are pipeline systems with different modules but all modules are differentiable and can be trained jointly.

Due to the complexity of end-to-end systems, a good implementation is necessary. There is statistical efficiency (i.e., how many training examples a system needs until it gets reasonable results) and computational complexity (i.e., how and after which training time the optimal parameters are found). In contrast to the past, it is now possible to apply backpropagation training through the whole end-to-end system.

The working group discussed specific deep learning models and their strengths and weaknesses in terms of learning long-distance dependencies (which is a crucial challenge in text processing, especially when the input is a character sequence). While recurrent neural networks (RNNs and especially LSTMs) are more powerful, also for modeling long-range dependencies, they are also harder to train and computationally more expensive. While LSTMs actively manage their memory (hidden layers), attention is a recently introduced mechanism that avoids the need of a memory. Convolutional neural networks (CNNs), on the other hand, are good at (low-level) feature extraction (they are usually applied in combination with fully connected layers afterwards) and computationally more efficient but they have limitations in capturing long range dependencies.

Hierarchical models have both advantages and disadvantages: Since documents are also built hierarchically, hierarchical models might be appropriate for modeling documents. However, processing a document without a hierarchical structure could factor out syntax and help focus on the semantic meanings.

5.4.4 Research Questions

The working group proposed the following set of research questions:

- **RQ1: Interpretation of results:** When papers report improvements for a specific task, it is central to show whether those improvements are also reproducible on other tasks and/or domains. It is important to see whether the improvement comes from a superior model or rather from a different (and maybe larger) training data set. In fact, systematic errors in the model or architecture might only be recognizable if the amount of training

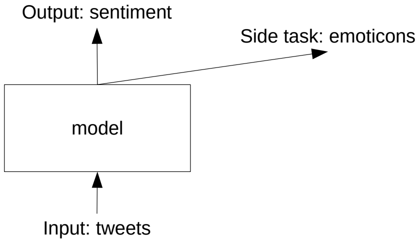

Figure 7 Example for data augmentation.

data is very large (infinite). This means that a more careful design of experiments as well as a detailed error analysis is necessary. A possibility to judge end-to-end models might be shared tasks because in shared tasks the competing systems are trained and evaluated on exactly the same dataset. Thus, direct comparisons of models are possible.

- **RQ2: Understanding the models:** Deep learning models and especially end-to-end models are often hard to interpret. Understandable models might be structured in a way that intermediate representations have meanings. For example, a model could have an intermediate layer that is trained to learn POS tag-like features. Similar to analyzing the results and errors of a model (see RQ1), the model itself should be inspected, too. Attention provides a straightforward way of visualizing attention weights. The question is whether this is enough to understand the model. Would it be possible, e.g., to visualize connections between words that are far away from each other in the input? In LSTMs, it is possible to analyze how the gradients flow through the different intermediate layers. In CNNs, most analysis focuses on extracting the most important n-gram features. An open question is whether these n-gram features are generalizable to other tasks and domains. In general, visualizing models or model components always comes with the theoretical question of what we actually expect.
- **RQ3: Integrating knowledge into end-to-end systems:** When training end-to-end systems, the system should learn everything by itself, i.e., without explicit feature design. Nevertheless, by designing the model, we implicitly use (and should use) knowledge about the task: For designing the input, it is crucial to know which information is needed for the task. When creating the model structure, we use our knowledge by deciding how the different modules should interact. For coming up with an appropriate objective function, it is important to know which function is most closely related to the application. That means although there is no explicit feature design in end-to-end learning, the models still reflect domain knowledge. Especially end-to-end systems are hard to train with little data. On the other hand, getting/annotating data can be very expensive. A cheap way of obtaining additional data for end-to-end systems in order to train them more robustly, is data augmentation. The concept is similar to transfer learning of domain information or multitask learning. Figure 7 shows how a side task can be used to better train a model for an application. In this example, the input to the model is tweets and the task is the prediction of sentiment. Beside the labeled data for the task (which might be little), there is a huge amount of unlabeled tweets available. Therefore, the model is extended with an additional output layer which predicts the emoticons in the unlabeled tweets. This allows a more robust training of the model part that creates a representation of the input tweet. The data augmentation process is another way of integrating domain knowledge since the selection of the side task needs a good understanding of the main task and domain.
- **RQ4: Adaptivity of end-to-end systems:** The adaptivity of end-to-end systems, e.g., to new domains is a very important challenge. An example would be self-driving cars in a

new city. A particular problem of end-to-end models is that they are very specialized to a particular task and domain. Furthermore, it is arguable whether there are still individual modules inside an end-to-end system that can be re-trained or exchanged with modules for the new domain (similar to "plug-and-play"). Another question that arises is how efficient adaptation can be ensured. A possibility might be defining and measuring an "adaptation distance". When re-training the whole end-to-end system for the new domain, it would be interesting to analyze which parts of the network have changed because of the adaptation (see RQ2).

5.5 Dialogue

Heike Adel (LMU München, DE)

Joint work of Heike Adel, Karl Moritz Hermann, Hang Li, Hermann Ney, Kristina Toutanova

5.5.1 Introduction

There are different variants of dialogue systems. The most important categorizations differ between task-independent vs. task-dependent systems, between single-turn or multi-turn dialogues and between retrieval-based vs. generation-based systems. Examples for task-independent models are chat bots. Task-dependent dialogue systems are, for example, question answering systems. The answer can be either found in structured data, e.g., templates, databases, knowledge graphs, or in unstructured data, e.g., a forum.

Character-based models are a great opportunity to improve dialogue systems since they provide the possibility of generating unseen words.

There are different data sets available in different languages [1]. STC [2], for example, consists of Chinese single-turn dialogue data from Weibo. Ubuntu [3] is an English multi-turn dialogue data set from a chat room. Reddit [4] contains English comment data from Reddit. Unfortunately, there is usually not enough data to train robust systems.

5.5.2 Research Topics

The working group formulated the following research topics:

- Sample Complexity
- Robust Question Answering
- Multi-source Question Answering
- Inference in Question Answering

References

1 Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, Joelle Pineau: A survey of available corpora for building data-driven dialogue systems. arXiv preprint (2017). arXiv:1512.05742
2 http://61.93.89.94/Noah_NRM_Data
3 https://github.com/rkadlec/ubuntu-ranking-dataset-creator
4 https://www.reddit.com/r/datasets/comments/3bxlg7/i_have_every_publicy_available_reddit_comment

6 Panel discussions

6.1 Panel discussion: Character-based models

Heike Adel (LMU München, DE)

Moderator: Manaal Faruqui
Participants: Ryan Cotterell, Kristina Toutanova, Chris Dyer, Jan Hajič, Phil Blunsom

The decision whether to model words, characters, bytes or even bits is often an engineering question depending on the bias of the models and the amount of available data. While characters are a part of our language, bytes have been defined by humans. Since there is a deterministic mapping between bytes and characters, it is a purely engineering decision whether to learn that mapping or not. While it might be possible to capture some additional signals or regularities when modeling bytes (e.g., capital letters have a fixed distance to their lowercase counterpart), going beyond that (and, e.g., modeling bits) might not be advantageous in general but could help for special tasks like pronunciation modeling.

Since we cannot give our models access to everything, it is useful to help them with linguistic information. (There might be an analogy to the epicycle history: Before the observation that the sun was in the center, it was very hard to predict the planet movements.) Linguistic features could, for example, be learned with multitask learning (as a side-knowledge for the main task) and they might also be useful to better interpret the model after training. However, it is an important question which kind of linguistics the model should learn: linguistic theories? Some notions of morphology? Or just something like tokenization? A promising solution could be to focus on phenomena described in most linguistic theories, i.e., phenomena most linguists agree on.

Modeling morphology is an important part of natural language processing/understanding. Sequences of characters do not capture all information. So, external knowledge about morphology can help the model to solve its task. While morphology can be useful from the empirical and engineering point of view, it is also an important theoretical topic (for example, to answer questions like how children learn morphemes).

For short-term artificial intelligence, end-to-end models are the quickest solution. However, for long-term solutions, multitask and transfer learning techniques will get more important because there is not enough data for end-to-end learning in general. Transfer learning is similar to how children learn: They learn something from task A and then leverage it for task B. Therefore, it is essential to separate representation learning and language understanding from specific applications. An example for successes of transfer learning are zero-shot translations. While deep learning facilitates the transfer of representations (even across modalities), getting it to work is not straightforward. For example, it is not clear which tasks we should use to learn to understand language and to figure out whether the models have learned anything about language or linguistics. For example, if the models just follow the traditional NLP pipeline, they do not learn anything about language but just about our own understanding of it. In fact, models should be able to induce such kind of knowledge as in, e.g., alignment learning in neural machine translation.

The prospects of sharing character-level encoders among tasks might be task dependent. However, at the character or morpheme level it should be possible to learn something general and task-independent. A character-based model that is trained across tasks can benefit from a large amount of training data. Examples are the success of multilingual models.

6.2 Plenary discussion: Where do we go from here?

Heike Adel (LMU München, DE)

Moderator: Adam Lopez

So far, we only know empirically that transfer learning works (example: word embeddings) but we do not have a theoretical basis for this (yet). It is an open question whether it is provable at all. There are similarites to semi-supervised learning for which we know criteria when it works and when it does not work. Transfer learning could be analyzed in a similiar way. A reason why transfer or multitask learning works in machine translation is that it prevents the decoder from only doing language modeling: Instead, the focus shifts more towards considering the source language. As a result, the system learns when to consider the source and when to consider the target language.

Transfer or multitask learning can cause "catastrophic forgetting". To mitigate this problem, examples from every task should be put into each batch and the weights should be updated based on the gradients gathered from all of them. This heuristic is used in machine translation research.

An open question is how to find out whether a model has gathered linguistic knowledge. In contrast to vision, analyzing the internal features of a neural network is hard for language. Possible ways to answer this question could be (i) to do intrinsic queries (e.g., the model should have learned that "glasses" refers to a single object although it is a plural word), (ii) to generate text based on the internal representations, (iii) to alter the input in a systematic way, (iv) to permute the predictions and differentiate them with the input (to see what is forcing the prediction and where the gradients come from), or (v) to break down a big task (such as neural machine translation) into smaller incremental tasks and analyze the neural network on those.

In general, there is a lot of data available. Rather than needing more data, we need more tasks and automatic evaluation methods.

Especially in the context of end-to-end learning, there are different objectives, such as language engineering vs. gaining linguistic insights, or building intelligent agents vs. understanding the human brain / language / behavior.

Participants

- Heike Adel
LMU München, DE
- Parnia Bahar
RWTH Aachen, DE
- Phil Blunsom
University of Oxford, GB
- Ondřej Bojar
Charles University – Prague, CZ
- Fabienne Cap
Uppsala University, SE
- Ryan Cotterell
Johns Hopkins University – Baltimore, US
- Vera Demberg
Universität des Saarlandes, DE
- Kevin Duh
Johns Hopkins University – Baltimore, US
- Chris Dyer
Carnegie Mellon University – Pittsburgh, US
- Desmond Elliott
University of Amsterdam, NL
- Manaal Faruqui
Carnegie Mellon University – Pittsburgh, US
- Orhan Firat
Middle East Technical University – Ankara, TR
- Alexander M. Fraser
LMU München, DE
- Vladimir Golkov
TU München, DE
- Jan Hajič
Charles University – Prague, CZ
- Georg Heigold
DFKI – Kaiserslautern, DE
- Karl Moritz Hermann
Google DeepMind – London, GB
- Thomas Hofmann
ETH Zürich, CH
- Hang Li
Huawei Technologies – Hong Kong, HK
- Adam Lopez
University of Edinburgh, GB
- Marie-Francine Moens
KU Leuven, BE
- Hermann Ney
RWTH Aachen, DE
- Jan Niehues
KIT – Karlsruher Institut für Technologie, DE
- Laura Rimell
University of Cambridge, GB
- Helmut Schmid
LMU München, DE
- Martin Schmitt
LMU München, DE
- Hinrich Schütze
LMU München, DE
- Kristina Toutanova
Google – Seattle, US
- Ngoc Thang Vu
Universität Stuttgart, DE
- Yadollah Yaghoobzadeh
LMU München, DE
- François Yvon
LIMSI – Orsay, FR

Report from Dagstuhl Seminar 17051

Theory and Applications of Behavioural Types

Edited by

Simon Gay[1], Vasco T. Vasconcelos[2], Philip Wadler[3], and Nobuko Yoshida[4]

1 **University of Glasgow, GB,** simon.gay@glasgow.ac.uk
2 **University of Lisbon, PT,** vmvasconcelos@ciencias.ulisboa.pt
3 **University of Edinburgh, GB,** wadler@inf.ed.ac.uk
4 **Imperial College London, GB,** yoshida@doc.ic.ac.uk

Abstract

This report documents the programme and the outcomes of Dagstuhl Seminar 17051 “Theory and Applications of Behavioural Types”. Behavioural types describe the dynamic aspects of programs, in contrast to data types, which describe the fixed structure of data. Perhaps the most well-known form of behavioural types is session types, which are type-theoretic specifications of communication protocols. More generally, behavioural types include typestate systems, which specify state-dependent availability of operations; choreographies, which specify collective communication behaviour; and behavioural contracts.

In recent years, research activity in behavioural types has increased dramatically, in both theoretical and practical directions. Theoretical work has explored new relationships between established behavioural type systems and areas such as linear logic, automata theory, process calculus testing theory, dependent type theory, and model-checking. On the practical side, there are several implementations of programming languages, programming language extensions, software development tools, and runtime monitoring systems, which are becoming mature enough to apply to real-world case studies.

The seminar brought together researchers from the established, largely European, research community in behavioural types, and other participants from outside Europe and from related research topics such as effect systems and actor-based languages. The questions that we intended to explore included:

- How can we understand the relationships between the foundations of session types in terms of linear logic, automata, denotational models, and other type theories?
- How can the scope and applicability of behavioural types be increased by incorporating ideas and approaches from gradual typing and dependent type theory?
- What is the relationship, in terms of expressivity and tractability, between behavioural types and other verification techniques such as model-checking?
- What are the theoretical and practical obstacles to delivering behavioural types to software developers in a range of mainstream programming languages?
- What are the advantages and disadvantages of incorporating behavioural types into standard programming languages or designing new languages directly based on the foundations of session types?
- How can we evaluate the effectiveness of behavioural types in programming languages and software development?

Seminar January 29–3, 2017 – http://www.dagstuhl.de/17051

1998 ACM Subject Classification D.1.1 Applicative (Functional) Programming, D.1.3 Concurrent Programming, D.2.4 Software/Program Verification, D.3.1 Formal Definitions and Theory, D.3.3 Language Constructs and Features, F.3.2 Semantics of Programming Languages

Theory and Applications of Behavioural Types, *Dagstuhl Reports*, Vol. 7, Issue 1, pp. 158–189
Editors: Simon Gay, Vasco T. Vasconcelos, Philip Wadler, and Nobuko Yoshida
DAGSTUHL REPORTS Dagstuhl Reports
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Keywords and phrases Behavioural Types, Programming Languages, Runtime Verification, Type Systems
Digital Object Identifier 10.4230/DagRep.7.1.158
Edited in cooperation with Tzu-Chun Chen

1 Executive Summary

Simon Gay
Vasco T. Vasconcelos
Philip Wadler
Nobuko Yoshida

Behavioural types describe dynamic aspects of a program, in contrast to data types, which describe the fixed structure of data. Behavioural types include session types, typestate, choreographies, and behavioural contracts. Recent years have seen a substantial increase in research activity, including theoretical foundations, design and implementation of programming languages and tools, studies of the relationships between different forms of behavioural types, and studies of the relationships between behavioural types and more general type-theoretic ideas such as gradual typing and dependent typing. The aim of this seminar was to bring together researchers on behavioural types and related topics, in order to understand and advance the state of the art.

Many of the participants have been active in COST Action IC1201: Behavioural Types for Reliable Large-Scale Software Systems (BETTY), a European research network on behavioural types. Other participants were invited from related research areas and from outside Europe, in order to broaden the scope of the seminar and to make connections between communities.

The programme for the first half of the week was planned in advance, with priority given to two kinds of presentation: (1) demonstrations of programming language implementations and tools, and (2) presentations by participants from outside the BETTY community. The programme for the second half of the week evolved during the seminar, with more emphasis on group discussion sessions.

The seminar was judged to be a success by all the participants. At least one conference submission resulted from collaboration started during the week, other existing collaborations made substantial progress, and several participants planned a submission to the EU RISE funding scheme. We intend to propose a follow-on seminar on a similar topic in the future.

This report contains the abstracts of the talks and software demonstrations, and summaries of the group discussion sessions.

2 Table of Contents

Executive Summary

Overview of Talks

Working groups

Open problems

3 Overview of Talks

3.1 Towards Inferring Session Types

Gul Agha (University of Illinois – Urbana-Champaign, US)

In sequential systems, programmers are responsible for specifying a total order of events in a system. This results in overly constraining when events may occur. In contrast, concurrent systems allow nondeterministic interleaving of actions at autonomous actors. Without additional constraints on the order of events at participating actors, an interleaving may lead to incorrect operations – for example, one that results in a deadlock. Moreover, the correct order of events at an actor is dependent on what interaction it is participating in. For example, an actor may be in the role of a client in one interaction protocol and the role of a backup server in another. To facilitate such flexibility, synchronization should be specified separately from the functional behavior of an actor – in terms of its interface rather than its representation. I will argue for the use of synchronization constraints as a user friendly language whose semantics is given by multiparty session types. Moreover, I propose that it is possible to infer session types with a degree of confidence by analyzing ordering patterns in traces of program execution: if an ordering pattern is repeatedly observed in such traces, we can impose the ordering to avoid Heisenbugs that may occur from rarer schedules that violate the observed order.

3.2 Effects as Capabilities

Nada Amin (EPFL – Lausanne, CH)

Joint work of Fengyun Liu, Nicolas Stucki, Sandro Stucki, Martin Odersky

It seems quite natural that one should track effects by means of a static typing discipline, similarly to what is done for arguments and results of functions. After all, to understand a function's contract and how it can be composed, knowing its effects is just as important as knowing the types of its arguments and result. Yet after decades of research [3, 4, 6, 5, 7, 8, 11, 13], why are effect systems not as mainstream as type systems?

The static effect discipline with the most widespread use is no doubt Java's system of checked exceptions. Ominously, they are now widely regarded as a mistake [2]. One frequent criticism is about the notational burden they impose. Throws clauses have to be laboriously threaded through all call chains. All too often, programmers make the burden go away by catching and ignoring all exceptions that they think cannot occur in practice. In effect, this disables both static and dynamic checking, so the end result is less safe than if one started with unchecked exceptions only. Another common problem of Java's exceptions is lack of polymorphism: Often we would like to express that a function throws the same exceptions as the (statically unknown) functions it invokes. Effect polymorphism can be expressed in Java only at the cost of very heavy notation, so it is usually avoided. Java's system of checked exceptions may be an extreme example, but it illustrates the general pitfalls of checking effects by shifting the burden of tracking effects to the programmer.

We are investigating a new approach to effect checking, that flips the requirements around. The central idea is that instead of talking about effects we talk about capabilities. For instance, instead of saying a function "throws an IOException" we say that the function "needs the capability to throw an IOException". Capabilities are modeled as values of some capability type. For instance, the aforementioned capability could be modeled as a value of type CanThrow[IOException]. A function that might throw an IOException needs to have access to an instance of this type. Typically it takes an argument of the type as a parameter.

It turns out that that the treatment of effects as capabilities gives a simple and natural way to express "effect polymorphism" – the ability to write a function once, and to have it interact with arguments that can have arbitrary effects. Since capabilities are just function parameters, existing language support for polymorphism, such as type abstraction and subtyping, is readily applicable to them. But there are two areas where work is needed to make capabilities as effects sound and practical.

First, when implemented naively, capabilities as parameters are even more verbose than effect declarations such as throws clauses. Not only do they have to be declared, but they also have to be propagated as additional arguments at each call site. We propose to make use of the concept of implicit parameters [9, 10, 14] to cut down on the boilerplate. Implicit parameters make call-site annotations unnecessary, but they still have to be declared just like normal parameters. To avoid repetition, we propose to investigate a way of abstracting implicit parameters into implicit function types. With implicits, the approach provides the common case of propagation for free, and an easy migration path from impure to pure.

Second, there is one fundamental difference between the usual notions of capabilities and effects: capabilities can be captured in closures. This means that a capability present at closure construction time can be preserved and accessed when the closure is applied. Effects on the other hand, are temporal: it generally does make a difference whether an effect occurs when a closure is constructed or when it is used. We propose to address this discrepancy by introducing a "pure function" type, instances of which are not allowed to close over effect capabilities.

In this talk, we report on work in progress, exploring the idea of effects as capabilities in detail. We have worked on minimal formalizations for implicit parameters and pure functions and studied encodings of higher-level language constructs into these calculi. Based on the theoretical modelization we are developing a specification for adding effects to Scala.

References

1 Lewis, Jeffrey R and Launchbury, John and Meijer, Erik and Shields, Mark B. *Implicit parameters: Dynamic scoping with static types.* Proceedings of POPL, 2000.

2 Thomas Whitmore. *Checked exceptions, Java 's biggest mistake.* Literal Java Blog, 2015.

3 Gifford, David K and Lucassen, John M. *Integrating functional and imperative programming.* Proceedings of POPL, 1986.

4 Lucassen, John M and Gifford, David K. *Polymorphic effect systems.* Proceedings of POPL, 1988.

5 Talpin, Jean-Pierre and Jouvelot, Pierre. *The type and effect discipline.* Information and computation, 1994.

6 Talpin, Jean-Pierre and Jouvelot, Pierre. *Polymorphic type, region and effect inference.* Journal of Functional Programming, 1992.

7 Wadler, Philip and Thiemann, Peter. *The marriage of effects and monads.* ACM Transactions on Computational Logic, 2003.

8 Filinski, Andrzej. *Monads in Action.* Proceedings of POPL, 2010.

9 Odersky, Martin. *Poor Man's Typeclasses*. Presentation to IFIP WG 2.8, 2006. http://lampwww.epfl.ch/~odersky/talks/wg2.8-boston06.pdf

10 Oliveira, Bruno CdS and Moors, Adriaan and Odersky, Martin. *Type classes as objects and implicits*. Proceedings of OOPSLA, 2010.

11 Rytz, Lukas and Odersky, Martin and Haller, Philipp. *Lightweight polymorphic effects*. Proceedings of ECOOP, 2012.

12 Ben Lippmeier. *Type Inference and Optimisation for an Impure World*. PhD Thesis, Australian National University, 2010.

13 Andrej Bauer and Matija Pretnar. *Programming with algebraic effects and handlers*. J. Log. Algebr. Meth. Program. 2015.

14 Oliveira, Bruno C.d.S. and Schrijvers, Tom and Choi, Wontae and Lee, Wonchan and Yi, Kwangkeun. *The Implicit Calculus: A New Foundation for Generic Programming*. Proceedings of PLDI, 2012.

3.3 Observed Communication Semantics for Classical Processes

Robert Atkey (University of Strathclyde – Glasgow, GB)

URL http://materials.dagstuhl.de/files/17/17051/17051.RobertAtkey.Slides.pdf

Classical Linear Logic (CLL) has long inspired readings of its proofs as communicating processes. Wadler's CP calculus is one of these readings. Wadler gave CP an operational semantics by selecting a subset of the cut-elimination rules of CLL to use as reduction rules. This semantics has an appealing close connection to the logic, but does not resolve the status of the other cut-elimination rules, and does not admit an obvious notion of observational equivalence. We propose a new operational semantics for CP based on the idea of observing communication, and use this semantics to define an intuitively reasonable notion of observational equivalence. To reason about observational equivalence, we use the standard relational denotational semantics of CLL. We show that this denotational semantics is adequate for our operational semantics. This allows us to deduce that, for instance, all the cut-elimination rules of CLL are observational equivalences.

3.4 Stateful Programming in Idris

Edwin Brady (University of St. Andrews, GB)

I present a library for giving precise types to interactive, stateful programs in Idris, a dependently typed pure functional programming language. I show how to describe state transition systems in types, capturing pre- and post-conditions of operations, and dealing with errors and feedback from the environment. I demonstrate with socket programming, and an asynchronous server for a simple network protocol.

3.5 Behavioral Types, Type Theory, and Logic

Luis Caires (New University of Lisbon, PT)

Joint work of Luis Caires, Frank Pfenning, Bernardo Toninho, Jorge Perez, Joao Seco

We review a collection of recent work providing a logical Curry-Howard foundation to the notion of behavioural type, useful to describe intensional usage protocols for state-full objects such as e.g., sessions. In particular we show how the basic linear logic interpretation discovered by Caires and Pfenning can be naturally extended to incorporate dependent types, allowing us to express higher order processes, value dependent behaviour, assertions, and proof carrying code; polymorphic types, allowing us to express behavioural genericity, and sums, allowing us to express non-determinism, and other typing constructs, relevant for typing shared state concurrency. We conclude by arguing that such linear logic interpretations provide a way of rooting the notion of behavioural type, and the notion of session type in particular, in the common house of Type Theory, from which the most fundamental programming language typing concepts have also emerged.

3.6 Session Types for Fault-tolerant Distributed Systems

Patrick Thomas Eugster (TU Darmstadt, DE)

Distributed systems are hard to get right, due to the possibility of partial failures where certain components or participants fail while others continue to operate. Session types are an appealing approach to aid programmers in reasoning about complex interaction in the presence of partial failures, yet have so far focused more on high-level programming models such as Web Services, where many failures are abstracted. Our contributions to address the problem in this talk are twofold. First we propose a set of abstractions allowing programmers to describe the handling of failures of different kinds. Together with information about the underlying system model we infer how and where to notify participants of failures in order to achieve a consistent failure handling as described by programmers. Second, we discuss the integration of failure handling mechanisms with failure masking approaches. In the latter context, we focus on supporting different broadcast models in order to support redundancy.

3.7 Statically Detecting (Dead)locks in the Linear Pi-calculus

Adrian Francalanza (University of Malta – Msida, MT)

Joint work of Adrian Francalanza, Marco Giunti, Antonio Ravara

We propose an alternative approach to the study of type-based (dead)lock analysis in the context of the linear pi-calculus. Instead of targeting the class of (dead)lock-free processes, we study type-based techniques for statically approximating the class of (dead)locked processes. We develop type-based analyses that return lists of problematic channels on which (dead)locks occur once the analysed program is executed. Such information is arguably more useful in the case of erroneous programs, because it directs the programmer to the source of the error. Another distinguishing aspect of our work is that the semantic guarantees of our type-based analysis ensure verdict precision (i.e. the absence of false negatives), but allow for occasionally classifying erroneous programs as bug-free. This differs from more mainstream static analysis approaches that tend to favour soundness, but is more useful for automated error resolution procedures where, ideally, the analysed programs are not be modified unnecessarily.

3.8 Gradual Typing

Ronald Garcia (University of British Columbia – Vancouver, CA)

Programming language design has recently exhibited a recurring trend: languages perceived as "statically typed" are beginning to exhibit "dynamic typing" features, while "dynamically typed" languages are exhibiting the converse. The theory of Gradual Typing has been developed to help provide a foundation for languages that wish to exhibit similar combinations while ensuring sound reasoning principles. This talk gives a high-level introduction to the concepts underlying gradual typing, with some historical context, some recent work on developing a general framework for developing gradually typed languages, and a list of open challenges that pertain to the behavioural types community.

3.9 Practical Affine Types and Typestate-Oriented Programming

Philipp Haller (KTH Royal Institute of Technology – Stockholm, SE)

URL http://materials.dagstuhl.de/files/17/17051/17051.PhilippHaller.Slides1.pdf

Aliasing is a known source of challenges in the context of imperative object-oriented languages, which have led to important advances in type systems for aliasing control. However, their large-scale adoption has turned out to be a surprisingly difficult challenge. While new language designs show promise, they do not address the need of aliasing control in existing languages.

This talk presents a new approach to isolation and uniqueness in an existing, widely-used language, Scala. The approach is unique in the way it addresses some of the most important obstacles to the adoption of type system extensions for aliasing control. First, adaptation of existing code requires only a minimal set of annotations. Only a single bit of information is required per class. Surprisingly, the talk shows that this information can be provided by the object-capability discipline, widely-used in program security. The type system is implemented for the full Scala language, providing, for the first time, a sound integration with Scala's local type inference. Finally, we present an ongoing effort to generalize the type system to typestates.

3.10 DCR Tools

Thomas Hildebrandt (IT University of Copenhagen, DK)

Joint work of Thomas Hildebrandt, Søren Debois, Tijs Slaats, Morten Marquard
Main reference S. Debois, T. T. Hildebrandt, M. Marquard, T. Slaats, "The DCR Graphs Process Portal", in Proc. of the BPM Demo Track 2016, pp. 7–11, 2016.
URL http://ceur-ws.org/Vol-1789/bpm-demo-2016-paper2.pdf

The presentation give a quick tour of the tools for modelling and simulating Dynamic Condition Response (DCR) graphs. DCR graphs is a declarative process notation for the modelling of flexible adaptable choreographies developed through a number of research projects jointly with industry with the aim to support the design, analysis and execution of flexible and adaptable workflow and business processes. Formally, DCR graphs generalise labelled event structures to allow (1) finite descriptions of infinite behaviour, (2) represent mandatory (pending) events that must eventually happen or become in conflict with events that happened, (3) allow dynamic, asymmetric conflict. Regarding expressiveness, the core model can express exactly the languages that are a union of regular and omega-regular languages (if one ignore true concurrency) – but the model maps to true concurrency models such as event structures. The presentation focus on the tools that have been developed at the IT University of Copenhagen (http://dcr.tools) and the industry partner Exformatics (http://dcrgraphs.net). The development of the two tools also demonstrate a model for transferring research to industry, where the academic tool serves as a means to demonstrate new developments that later are transferred to the industrial tool.

3.11 Using Session Types for Reasoning About Boundedness in the Pi-Calculus

Hans Hüttel (Aalborg University, DK)

Depth-bounded and name-bounded processes are pi-calculus processes for which some of the decision problems that are undecidable for the full calculus become decidable. P is depth-bounded at level k if every reduction sequence for P contains successor processes with at most k active nested restrictions. P is name-bounded at level k if every reduction sequence for P contains successor processes with at most k active bound names. We use binary session types to formulate two type systems that give sound characterizations of these properties: If a process is well-typed, it is depth-bounded, respectively name-bounded.

3.12 Session-ocaml: A Session-based Library with Polarities and Lenses

Keigo Imai (Gifu University, JP), Nobuko Yoshida (Imperial College London, GB), and Shoji Yuen (Nagoya University, JP)

Main reference K. Imai, N. Yoshida, S. Yuen, "Session-ocaml: a session-based library with polarities and lenses", Manuscript, 2017.
URL http://www.ct.info.gifu-u.ac.jp/~keigoi/session-ocaml/

We propose session-ocaml, a novel library for session-typed concurrent/distributed programming in OCaml. Our technique is based only on the parametric polymorphism, hence common to various statically-typed programming languages. The key ideas are follows: (1) The polarised session types gives an alternative formulation of duality enabling OCaml to infer the appropriate session type in a session with a reasonable notational overhead. (2) A parameterized monad with lenses enables full session type implementation including delegation. We show an application of session-ocaml including an SMTP client and a database server.

3.13 Lightweight Functional Session Types

J. Garrett Morris (University of Edinburgh, GB)

Joint work of Sam Lindley, J. Garrett Morris
Main reference S. Lindley, J. G. Morris, "Lightweight Functional Session Types", in "Behavioural Types: from Theory to Tools", River Publishers, 2017; pre-print available from author's webpage.
URL http://homepages.inf.ed.ac.uk/slindley/papers/fst.pdf

Row types provide an account of extensibility that combines well with parametric polymorphism and type inference. We discuss the integration of row types and session types in a concurrent functional programming language, and how row types can be used to describe extensibility in session-typed communication.

3.14 Composable Actor Behaviour

Roland Kuhn (Actyx AG – München, DE)

URL http://materials.dagstuhl.de/files/17/17051/17051.RolandKuhn.Slides.pdf

This presentation focuses on the composition of the behavior of distributed components–modeled using Actors–from reusable pieces. Allowing abstraction and type-safety to be applied within these components for operations that are fundamentally non-local is seen as a prerequisite for offering safe construction of distributed systems in a widely and practically applicable programming tool.

Please see the linked article for an introduction to the tool (based on Scala and Akka); pointers to the source code and how to try it out are given towards the end.

3.15 Adaptive Interaction-Oriented Choreographies in Jolie

Ivan Lanese (University of Bologna, IT)

We will give a demo of AIOCJ, Adaptive Interaction-Oriented Choreographies in Jolie. The tool is composed by an Eclipse plugin and a running environment to program distributed applications, and to adapt them at runtime by replacing pre-selected pieces of code with new code coming from outside the application. Notably, a single program describes the whole distributed application, and a single adaptation may involve many components. The application is free from communication races and deadlocks by construction, both before and after the adaptation.

3.16 Failure-Aware Protocol Programming

Hugo-Andrés López (Technical University of Denmark – Lyngby, DK)

URL http://materials.dagstuhl.de/files/17/17051/17051.Hugo-Andr%C3%A9sL%C3%B3pez.Slides.pdf

Motivated by challenging scenarios in Cyber-Physical Systems (CPS), we study how choreographic programming can cater for dynamic infrastructures where not all endpoints are always available. We introduce the Global Quality Calculus (GCq), a variant of choreographic programming for the description of communication systems where some of the components involved in a communication might fail. GCq features novel operators for multiparty, partial and collective communications. In this talk I will study the nature of failure-aware communication: First, we introduce GCq syntax, semantics and examples of its use. The interplay between failures and collective communications in a choreography can lead to choreographies that cannot progress due to absence of resources. In our second contribution, we provide a type system that ensures that choreographies can be realized despite changing availability conditions. A specification in GCq guides the implementation of distributed endpoints when paired with global (session) types. Our third contribution provides an endpoint-projection based methodology for the generation of failure-aware distributed processes. We show the correctness of the projection, and that well-typed choreographies with availability considerations enjoy progress.

3.17 Chaperone Contracts for Higher-Order Sessions

Hernán Melgratti (University of Buenos Aires, AR)

URL http://materials.dagstuhl.de/files/17/17051/17051.Hern%C3%A1nMelgratti.Slides.pdf

Sessions in concurrent programs play the same role of functions and objects in sequential ones. This calls for a way to describe properties and relationships of messages exchanged in sessions using behavioral contracts, in the spirit of the design-by-contract approach to software development. Unlike functions and objects, however, the kind, direction, and properties of messages exchanged in a session may vary over time, as the session progresses. This feature of sessions enriches the "behavioral" qualification of session contracts, which must evolve along with the session they describe.

In this work, we extend to sessions the notion of chaperone contract (roughly, a contract that applies to a mutable object) and investigate the ramifications of contract monitoring in a higher-order calculus equipped with a session type system. We give a characterization of correct module, one that honors the contracts of the sessions it uses, and prove a blame theorem. Guided by the calculus, we describe a lightweight and portable implementation of monitored sessions as an OCaml module with which programmers can benefit from static session type checking and dynamic contract monitoring using an off-the-shelf version of OCaml.

3.18 Static Deadlock Detection for Go

Nicholas Ng (Imperial College London, GB) and Nobuko Yoshida (Imperial College London, GB)

Joint work of Julien Lange, Nicholas Ng, Bernardo Toninho, Nobuko Yoshida
Main reference J. Lange, N. Ng, B. Toninho, N. Yoshida, "Fencing off go: liveness and safety for channel-based programming", in Proc. of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2017), pp. 748–761, ACM, 2017.
URL http://dx.doi.org/10.1145/3009837.3009847

Go is a production-level statically typed programming language whose design features explicit message-passing primitives and lightweight threads, enabling (and encouraging) programmers to develop concurrent systems where components interact through communication more so than by lock-based shared memory concurrency. Go can only detect global deadlocks at runtime, but provides no compile-time protection against all too common communication mismatches or partial deadlocks.

In this talk we present a static verification framework for liveness and safety in Go programs, able to detect communication errors and partial deadlocks in a general class of realistic concurrent programs, including those with dynamic channel creation, unbounded thread creation and recursion. Our approach infers from a Go program a faithful representation of its communication patterns as a behavioural type. By checking a syntactic restriction on channel usage, dubbed fencing, we ensure that programs are made up of finitely many different communication patterns that may be repeated infinitely many times. This restriction allows us to implement a decision procedure for liveness and safety in types which in turn statically ensures liveness and safety in Go programs.

Details of our verification tool-chain are available on http://mrg.doc.ic.ac.uk/tools/gong/.

3.19 Session Types with Linearity in Haskell

Dominic Orchard (University of Kent – Canterbury, GB) and Nobuko Yoshida (Imperial College London, GB)

Main reference D. Orchard, N, Yoshida, "Sessions types with linearity in Haskell", in Behavioural Types: from Theory to Tools, River Publishers, 2017.

Type systems with parametric polymorphism can encode communication patterns over channels, providing part of the power of session types. However, statically enforcing linearity properties of session types is more challenging. Haskell provides various features that can overcome this challenge. However, current approaches lead to a programming style which is either non-idiomatic for Haskell, or types which are too hard to write and read. I'll demo an early version of a Haskell library for session types that does it all: session-typed, linear, idiomatic Haskell with easy-to-read-and-write types.

3.20 A Simple Library Implementation of Binary Sessions

Luca Padovani (University of Turin, IT)

Main reference L. Padovani, "A simple library implementation of binary sessions", Journal of Functional Programming, Vol. 27:e4, 2017.
URL http://dx.doi.org/10.1017/S0956796816000289

This demo is about FuSe, a simple OCaml implementation of binary sessions that supports delegation, equi-recursive, polymorphic, context-free session types, session subtyping, and allows the OCaml compiler to perform session type checking and inference.

3.21 Concurrent TypeState-Oriented Programming

Luca Padovani (University of Turin, IT)

Joint work of Silvia Crafa, Luca Padovani
Main reference S. Crafa, L. Padovani, "The chemical approach to typestate-oriented programming", in Proc. of the 2015 ACM SIGPLAN Int'l Conf. on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA 2015), pp. 917–934, ACM, 2015.
URL http://dx.doi.org/10.1145/2814270.2814287

This demo is about CobaltBlue, a tool for the static behavioural analysis of Objective Join Calculus scripts. The tool checks that concurrent objects and actors (modelled as terms in the Objective Join Calculus) are consistent with – and are used according to – their protocol.

3.22 Precise Subtyping

Jovanka Pantovic (University of Novi Sad, RS)

Joint work of Mariangiola Dezani-Ciancaglini, Silvia Ghilezan, Svetlana Jaksic, Jovanka Pantovic, Alceste Scalas, Nobuko Yoshida
Main reference M. Dezani-Ciancaglini, S. Ghilezan, S. Jaksic, J. Pantovic, N. Yoshida, "Precise subtyping for synchronous multiparty sessions", in Proc. of PLACES 2015, pp. 29–43, 2015.
URL http://dx.doi.org/10.4204/EPTCS.203.3

A subtyping relation is operationally precise if both the soundness and the completeness with respect to type safety are satisfied. Soundness provides safe replacement of a term of a smaller type when a term of a bigger type is expected. Is such a relation is the greatest one, we get the completeness. We discuss the notion of operational preciseness, methodology for proving the completeness and show how it works on the example of multiparty session subtyping.

3.23 Concurrent C0

Frank Pfenning (Carnegie Mellon University – Pittsburgh, US)

Joint work of Max Willsey, Rokhini Prabhu, Frank Pfenning
Main reference M. Willsey, R. Prabhu, F. Pfenning, "Design and Implementation of Concurrent C0", in Proc. LINEARITY 2016, EPTCS 238, pp. 73–82, 2017.
URL http://dx.doi.org/10.4204/EPTCS.238.8

We give a demo of Concurrent C0, an imperative language extended with session-typed message-passing concurrency. C0 is a type-safe and memory-safe subset of C, extended with a layer of contracts, and has been used in teaching introductory programming at Carnegie Mellon University since 2010. The extension follows the Curry-Howard interpretation of intuitionistic linear sequent calculus, adapted to the linear setting. Considerable attention has been paid to programmer-friendly features such as good error messages from the lexer, parser, and (linear) type-checker. Access to Concurrent C0 can be obtain from the author. The live-coded of a concurrent append function is available in the additional materials.

3.24 Manifest Sharing with Session Types

Frank Pfenning (Carnegie Mellon University – Pittsburgh, US)

Joint work of Stephanie Balzer, Frank Pfenning
Main reference S. Balzer, F. Pfenning, "Manifest Sharing with Session Types", Technical Report CMU-CS-17-106, Carnegie Mellon University, 2017.
URL http://materials.dagstuhl.de/files/17/17051/17051.FrankPfenning.Slides.pdf

We report on work in progress to reconcile sharing of resources in logically based session typed languages. The key idea is to decompose the exponential modality of linear logic into two adjoint modalities and then give a nonstandard operational interpretation of the shared layer. As a side effect, it seems we can faithfully interpret the (untyped) asynchronous pi-calculus, answering a question by Wadler.

3.25 Detecting Concurrency Errors of Erlang Programs via Systematic Testing

Konstantinos Sagonas (Uppsala University, SE)

Main reference P. Abdulla, S. Aronis, B. Jonsson, K. Sagonas, "Optimal Dynamic Partial Order Reduction", in Proc. of the 41st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 373–384, ACM, 2014. Extended and significantly revised version to appear in the Journal of the ACM.
URL https://dx.doi.org/10.1145/2535838.2535845

Testing and verification of concurrent programs is an important but also challenging problem. Effective techniques need to faithfully model the semantics of the language primitives and have a way to combat the combinatorial explosion of the possible different ways that threads may interleave (scheduling non-determinism). In this talk we will focus on a particular verification technique known as stateless model checking (a.k.a. systematic concurrency testing) and we will present Concuerror, a state-of-the-art tool for finding and reproducing errors in concurrent Erlang programs. Time permitting, we will briefly review the algorithms that Concuerror employs in order to examine only an optimal (but sound) subset of all interleavings.

More information about the tool can be found at http://www.concuerror.com.

3.26 Lightweight Session Programming in Scala

Alceste Scalas (Imperial College London, GB)

Joint work of Alceste Scalas, Nobuko Yoshida
Main reference A. Scalas, N. Yoshida, "Lightweight Session Programming in Scala", in Proc. of the 30th European Conf. on Object-Oriented Programming (ECOOP 2016), LIPIcs, Vol. 56, pp. 21:1–21:28, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2016.
URL http://dx.doi.org/10.4230/LIPIcs.ECOOP.2016.21

Designing, developing and maintaining concurrent applications is an error-prone and time-consuming task; most difficulties arise because compilers are usually unable to check whether the inputs/outputs performed by a program at runtime will adhere to a given protocol specification. To address this problem, we propose lightweight session programming in Scala: we leverage the native features of the Scala type system and standard library, to introduce (1) a representation of session types as Scala types, and (2) a library, called lchannels, with a convenient API for session-based programming, supporting local and distributed communication. We generalise the idea of Continuation-Passing Style Protocols (CPSPs), studying their formal relationship with session types. We illustrate how session programming can be carried over in Scala: how to formalise a communication protocol, and represent it using Scala classes and lchannels, letting the compiler help spotting protocol violations. We attest the practicality of our approach with a complex use case, and evaluate the performance of lchannels with a series of benchmarks.

3.27 Programming Protocols with Scribble and Java

Alceste Scalas (Imperial College London, GB)

Joint work of Raymond Hu, Alceste Scalas, Nobuko Yoshida
Main reference R. Hu, N. Yoshida, "Hybrid Session Verification Through Endpoint API Generation", in Proc. of the Int'l Conf. on Fundamental Approaches to Software Engineering (FASE 2016), LNCS, Vol. 9633, pp. 401–418, Springer, 2016.
URL http://dx.doi.org/10.1007/978-3-662-49665-7_24

I will provide a brief introduction to Scribble – http://www.scribble.org/.

Scribble is both a language for defining global protocols involving multiple participants, and a tool that can verify the properties of such protocols (in particular: absence of deadlocks and orphan messages). Scribble can also automatically generate APIs that simplify the implementation of protocol-abiding programs. Its approach is based on the Multiparty Session Types framework.

During the talk I will illustrate the Scribble description of the SMTP protocol, and an SMTP client based on Scribble-generated APIs for Java.

3.28 Partial Type Equivalences for Verified Dependent Interoperability

Nicolas Tabareau (Ecole des Mines de Nantes, FR)

Joint work of Pierre-Évariste Dagand, Nicolas Tabareau, Éric Tanter
Main reference P.-E. Dagand, N. Tabareau, E. Tanter, "Partial type equivalences for verified dependent interoperability", in Proc. of the 21st ACM SIGPLAN Int'l Conf. on Functional Programming (ICFP 2016), pp. 298–310, ACM, 2016.
URL http://dx.doi.org/10.1145/2951913.2951933

Full-spectrum dependent types promise to enable the development of correct-by-construction software. However, even certified software needs to interact with simply-typed or untyped programs, be it to perform system calls, or to use legacy libraries. Trading static guarantees for runtime checks, the dependent interoperability framework provides a mechanism by which simply-typed values can safely be coerced to dependent types and, conversely, dependently-typed programs can defensively be exported to a simply-typed application. In this paper, we give a semantic account of dependent interoperability. Our presentation relies on and is guided by a pervading notion of type equivalence, whose importance has been emphasized in recent works on homotopy type theory. Specifically, we develop the notion of partial type equivalences as a key foundation for dependent interoperability. Our framework is developed in Coq; it is thus constructive and verified in the strictest sense of the terms. Using our library, users can specify domain-specific partial equivalences between data structures. Our library then takes care of the (sometimes, heavy) lifting that leads to interoperable programs. It thus becomes possible, as we shall illustrate, to internalize and hand-tune the extraction of dependently-typed programs to interoperable OCaml programs within Coq itself.

3.29 Gradual Session Types

Peter Thiemann (Universität Freiburg, DE)

URL http://materials.dagstuhl.de/files/17/17051/17051.PeterThiemann.Slides.pdf

Session types describe structured communication on heterogeneously typed channels at a high level. They lift many of the safety claims that come with sound type systems to operations on communcation channels.

The use of session types requires a fairly rich type discipline including linear types in the host language. However, web-based applications and micro services are often written on purpose in a mix of languages, with very different type disciplines in the spectrum between static and dynamic typing.

Effective use of session typing in this setting requires a mix of static and dynamic type checking. Gradual session types address this mixed setting by providing a framework which grants seamless transition between statically typed handling of sessions and any required degree of dynamic typing.

We propose GradualGV as an extension of the functional session type system GV with dynamic types and casts. We use AGT as a guideline to obtain a consistent static semantics which conservatively extends GV. We demonstrate type and communication safety as well as blame safety, thus extending previous results to functional languages with session-based communication. Our system differs from previous gradually typed systems in two respects: the interplay of linearity and dynamic types as well as the necessity to deal with changing type state requires a novel approach to specifying the dynamics of the language.

3.30 Choreographies, Modularly: Components for Communication Centred Programming

Hugo Torres Vieira (IMT – Lucca, IT)

Joint work of Marco Carbone, Fabrizio Montesi, Hugo Torres Vieira

As communicating systems are becoming evermore complex it is crucial to conceive programming abstractions that support modularity in the development of communicating systems. In this talk we present a new model for the modular development of component-based software, following the reactive style, i.e., computations in a component are triggered by the availability of new data. The key novelty is the mechanism for composing components, which is based on multiparty protocols given as choreographies. We show how our model can be compiled to a fully-distributed implementation by translating our terms into a process calculus, and present a type system for ensuring communication safety, deadlock-freedom, and liveness.

3.31 From Communicating Machines to Graphical Choreographies

Emilio Tuosto (University of Leicester, GB)

I will showcase ChorGram (https://bitbucket.org/emlio_tuosto/chorgram/wiki/Home), a tool for the reconstruction of choreographies from systems consisting of a class of communicating automata. After a very brief and lightweight introduction to the underlying theory, I will demonstrate how ChorGram can help in designing and analyse communication-centric applications.

3.32 Fencing off Go

Nobuko Yoshida (Imperial College London, GB)

URL http://materials.dagstuhl.de/files/17/17051/17051.NobukoYoshida.Preprint.pdf

Go is a production-level statically typed programming language whose design features explicit message-passing primitives and lightweight threads, enabling (and encouraging) programmers to develop concurrent systems where components interact through communication more so than by lock-based shared memory concurrency. Go can only detect global deadlocks at runtime, but provides no compile-time protection against all too common communication mis-matches or partial deadlocks. This work develops a static verification framework for liveness and safety in Go programs, able to detect communication errors and partial deadlocks in a general class of realistic concurrent programs, including those with dynamic channel creation, unbounded thread creation and recursion. Our approach infers from a Go program a faithful representation of its communication patterns as a behavioural type. By checking a syntactic restriction on channel usage, dubbed fencing, we ensure that programs are made up of finitely many different communication patterns that may be repeated infinitely many times. This restriction allows us to implement a decision procedure for liveness and safety in types which in turn statically ensures liveness and safety in Go programs. We have implemented a type inference and decision procedures in a tool-chain and tested it against publicly available Go programs.

3.33 Undecidability of Asynchronous Session Subtyping

Nobuko Yoshida (Imperial College London, GB)

URL http://materials.dagstuhl.de/files/17/17051/17051.NobukoYoshida1.Preprint.pdf

Asynchronous session subtyping has been studied extensively and applied in the literature. An open question was whether this subtyping relation is decidable. This paper settles the question in the negative. To prove this result, we first introduce a new sub-class of two-party communicating finite-state machines (CFSMs), called asynchronous duplex (ADs), which we show to be Turing complete. Secondly, we give a compatibility relation over CFSMs, which is sound and complete wrt. safety for ADs, and is equivalent to the asynchronous subtyping. Then we show that checking whether two CFSMs are in the relation reduces to the halting problem for Turing machines. In addition, we show the compatibility relation to be decidable for three sub-classes of ADs.

4 Working groups

4.1 Group Discussion: Integrating Static and Dynamic Typing

Laura Bocchi (University of Kent – Canterbury, GB)

Joint work of Approximately half of the seminar participants.

The starting point for the discussion was that most of us have worked or are working on static typing, some of us on dynamic typing and monitoring, or even on the combination of static and dynamic verification in a network (but not in the same node), and only a few have direct experience of gradual and hybrid typing. Gradual/hybrid behavioural typing is quite a new thread.

The motivation is that run-time monitoring is critical in several contexts (e.g. when addressing security issues, in untrusted networks, in real-time scenarios where it is harder to make precise predictions). Run-time mechanisms provide programmers with better access to the current state of objects, which is often unclear at compile-time. Usability is also a motivation.

What does it mean to monitor? The notion of monitors is strictly related to the notion of contract. Monitors are contracts, which are used to check interactions. There are two main aspects of monitoring: verification (check behaviour and determine blame) and enforcement (e.g. suppress bad messages).

What does it mean to "go right and wrong"? There is a need for systematic construction of, and reasoning about, monitors.

Gradual/hybrid typing require a more complex notion of correctness than the usual type safety given by static typing. Critical to this aim is the role of blame. There are several views of blame, including at least the following, and any points in between. Blame the the less precisely-typed code (e.g., when hybrid typing) [1]. Blame anybody who has violated the contract [2] Blame who originated the first contract violation (implicitly assumed in [3]).

Blame in the case of sharable resources is not obvious. Shared resources may be linked to "something" linear which makes it not obvious to assign blame. This is a problem that comes with linearity (affinity would be ok).

There is a general interest in a mathematical model of blame.

Blame is also useful as it introduces a "social process" in the sense that it makes programmers want to work hard to satisfy their contracts (assuming that contract violations throws blame on others). This may promote the use of contracts in practice.

There are some limitations of dynamic types. One critical problem is to define the boundaries between static and dynamic typing. Both static and dynamic typing have advantages and disadvantages. We focused, in our discussion, on the limitations of dynamic typing: worse performance (due to overhead), no progress guarantees, in some cases more expressive but in some other cases less expressive (e.g, when using parametricity or talking about multiple runs, to check branching-time properties, limitations like monitorability [3] when having assertions on message content, loss of transparency in timed scenarios)

References

1 Philip Wadler, Robby Findler. *Well-typed programs can't be blamed.* Proceedings of ESOP, 2009.

2 Massimo Bartoletti, Alceste Scalas, Emilio Tuosto, Roberto Zunino. *Honesty by Typing.* Logical Methods in Computer Science 12(4), 2016.

3 Laura Bocchi, Tzu-chun Chen, Romain Demangeon, Kohei Honda, Nobuko Yoshida. *Monitoring Networks through Multiparty Session Types.* Proceedings of FORTE, 2013.

4 Limin Jia, Hannah Gommerstadt, Frank Pfenning *Monitors and Blame Assignment for Higher-Order Session Types.* Proceedings of POPL, 2016.

5 Christos Dimoulas, Robert Bruce Findler, Cormac Flanagan, Matthias Felleisen. *Correct blame for contracts: no more scapegoating* Proceedings of POPL, 2011.

6 Cameron Swords, Amr Sabry, Sam Tobin-Hochstadt. *Expressing Contract Monitors as Patterns of Communication.* Proceedings of ICFP, 2015.

7 Tim Disney, Cormac Flanagan, Jay McCarthy. *Temporal Higher-Order Contracts.* Proceedings of ICFP, 2011.

4.2 Group Discussion: Behavioural Types in Non-Communication Domains

Simon Gay (University of Glasgow, GB)

Joint work of Approximately half of the seminar participants

The discussion focused on three main topics.

What is a behavioural type?

Hans Hüttel quoted the definition "... notions of typing that are also able to describe properties associated with the behaviour of programs and in this way also describe how a computation proceeds. This often includes accounting for the notions of causality and choice." [1] Examples include session types, type state, effect types, coeffect types, information flow, intersection types, differential types. In many systems, behavioural types evolve with the reduction of terms, whereas standard types remain the same. However, consider the functional programming style based on the GV calculus: the types don't change with reduction, but instead due to rebinding – use linearity to encode changing types. The simply-typed lambda calculus is not an example of a behavioural type system. Whilst this can be translated into communication [2, 3], simple-types within the lambda calculus are not themselves behavioural. Computations in one language can be translated into communication in another language, capturing intensional aspects of a program [2, 4, 5].

Another view is that non-behavioural types characterise the final value of the computation, whereas behavioural types describe how the computation proceeds. Simple types control termination, but are not seen as inherently behavioural. We could say that behavioural types include everything that's not a simple type. Logical relations give meaning to types, and can have computational content, e.g. due to effects (trace properties).

In what areas, other than communication, do type-state like constraints occur?

It is interesting to infer type-state specifications for a given API to avoid some undesired behaviour, e.g. to avoid dereferencing a null, or to infer sequencing constraints. Jonathan Aldrich's group have done empirical work on the occurrence of type-state in the wild [6]. It relates strongly to notions of identity and state but a linear discipline allows it to be decomposed in a copying semantics (e.g. in a pure functional setting). Another empirical study of programming protocols is in Joshua Sunshine's thesis, Chapter 3: "Quantitative study of API protocol usability".

Several people gave examples.

Hugo López: cyberphysical systems have control events where the connection between events are unknown (cf. shared memory concurrency) and timing plays a part. In message-passing concurrency the links between events are much more clear. This is related to [7].

Keigo Imai: Type-state example: in a smartphone there is a lot of context switching, involving serialising state, on a low-memory device. Applications are in various active/inactive states, and this changes the user's capabilities to interact with each.

Francisco Martins: Related example: different hardware components get turned on and off or have different capabilities for the purposes of battery saving. Programmers could be forced to follow the protocols such that a resource's handles are closed and battery is saved.

Garrett Morris: L4 microkernel off-loads responsibility for memory to user programs. Programs have to ask kernel to subdivide their heap allocation for subthreads, which then they give up some capability. Can't DDOS the kernel by asking for lots of thread control blocks, because now these are within the purview of the programs. Have to manage the capabilities yourself. The state of the capabilities is a key part of the kernel/program interaction; the server can refuse requests that violate previous capability assignments. Microkernel design could benefit from type state definition.

Dimitrios Kouzapas: Data processing, and private data, e.g. camera photographs and recognises number plate, if car is speeding the data is kept, if not the data is dropped. This is a protocol on the data and relates to provenance.

Thomas Hildebrandt: There are legal frameworks for the behaviour of how our data is processed. We want to mediate between the contracts.

Giovanni Bernardi: The idea of effects generalises the idea of communication, and mathematically this works out as a model of some kind of modality.

We considered why behavioural and linear types became linked in the first place. There are behavioural specifications which don't need linearity. Linearity gives us a way to encode the state changes. But is this too much? Bhavioural types could help people who write concurrent data structures. Does this provide a way to explain where locking is and isn't needed?

Areas for future research

- Consider behavioural types for concurrent data structures and algorithms.
- Perhaps we could weaken certain assumptions about shared channels so that protocols of interaction are more easily distributed and non-binary interactions are expressible, on shared channels, which may or may not imply locking/mutual exclusion.
- Look at the work of Beckman et al. [6] to give us a source of case studies for type-state systems and see what features can be captured by our current tools.

- Re-examine the assumption that linearity is necessary, or at least re-examine its realisation, in the light of work such as Frank Pfenning's talk during the seminar.
- Review work on formal theories about the interaction of different behaviours, e.g., communication and hardware schedules in FPGA systems [8], probabilities and exceptions [9], differential types and state.

References

1 Hans Hüttel, Ivan Lanese, Vasco T. Vasconcelos, Luís Caires, Marco Carbone, Pierre-Malo Deniélou, Dimitris Mostrous, Luca Padovani, António Ravara, Emilio Tuosto, Hugo Torres Vieira and Gianluigi Zavattaro. *Foundations of Session Types and Behavioural Contracts*. ACM Computing Surveys 49(1) 3:1–3:36, 2016.

2 Robin Milner. *Functions as processes*. Proceedings of ICALP, 1990.

3 Bernardo Toninho, Luís Caires and Frank Pfenning. *Functions as session-typed processes*. Proceedings of FOSSACS, 2012.

4 Dominic Orchard and Nobuko Yoshida. *Effects as sessions, sessions as effects*. Proceedings of POPL, 2016.

5 Cameron Swords, Amr Sabry and Sam Tobin-Hochstadt. *Expressing Contract Monitors as Patterns of Communication*. Proceedings of ICFP, 2015.

6 Nels Beckman, Duri Kim and Jonathan Aldrich. *An Empirical Study of Object Protocols in the Wild*. Proceedings of ECOOP, 2011.

7 Tim Disney, Cormac Flanagan, Jay McCarthy. *Temporal Higher-Order Contracts*. Proceedings of ICFP, 2011.

8 Xinyu Niu, Nicholas Ng, Tomofumi Yuki, Shaojun Wang, Nobuko Yoshida and Wayne Luk. *EURECA compilation: Automatic optimisation of cycle-reconfigurable circuits*. Proceedings of FPL, 2016.

9 Marco Gaboardi, Shin-ya Katsumata, Dominic Orchard, Flavien Breuvart and Tarmo Uustalu. *Combining effects and coeffects via grading*. Proceedings of ICFP, 2016.

4.3 Group discussion: Dependent Session Types

Simon Gay (University of Glasgow, GB)

Joint work of Approximately half of the seminar participants.

Several researchers have studied dependent session types, in which the types of messages may depend on the values of previous messages. Three lines of work have been produced by different participants in the seminar.

Frank Pfenning, Luis Caires and Bernardo Toninho have included dependent types in the linear logic / Curry-Howard approach to session types, and have shown how they can encode features such as proof-irrelevance.

Conor McBride has developed a general setting for combining dependent types and linear types, by distinguishing between "consumption" and "contemplation", that is, value-level and type-level uses of data. He has used dependent session types as an example of a specific type theory that can be developed in this setting.

Edwin Brady has embedded session types and related concepts of typestate in his general-purpose dependently-typed programming language, Idris.

The early part of the discussion focused on understanding the relationships between these three approaches. Conor McBride emphasised the need to be clear about what it is that a session type can depend on: in his view it is the traffic on a channel, and this is consistent with the other approaches; note that dependence on the identity of a channel would be a different concept.

There was some discussion about the possibility of dependence on the behaviour of a process. Dependence on traffic is one aspect of this idea, but there could be others. It leads to the need to define equivalence between processes.

Towards the end of the discussion, areas for further research were identified.

- More detailed comparisons between the different approaches to dependent session types.
- Further study of process equivalence.
- The relationship between intuitionistic and classical formulations of the logical foundation of session types.
- Understanding the possibility of dependence on channel identity.

4.4 Group Discussion: Future Activities and Funding Possibilities

Simon Gay (University of Glasgow, GB)

Joint work of Approximately half of the seminar participants.

The background to this discussion is that most of the participants in the seminar were involved in COST Action IC1201 (BETTY: Behavioural Types for Reliable Large-Scale Software Systems), which ran for four years from October 2012 to October 2016. The seminar included participants from outside the BETTY group, in order to bring in relevant ideas from broader research topics. The end of the COST Action naturally prompted discussion about future activities for the community, and future funding for research on behavioural types. These two points are closely linked.

First we discussed future activities, independently of funding. We agreed that another Dagstuhl seminar would be worthwhile, with an expanded or different combination of people from related topics. Concurrent Separation Logic was mentioned as a relevant topic. There were several suggestions for different ways of organising a Dagstuhl seminar, especially in relation to the choice of discussion topics and the possibility of specifying discussion topics before the beginning of the seminar. We also noted the possibility of a more focussed meeting on a topic such as programming language design. Dagstuhl is not the only possible location for a similar seminar: we could consider the Shonan centre in Japan, or the Banff centre in Canada. However, Dagstuhl is the most convenient and we agreed to propose another seminar. The organisers of the present seminar said that they would be willing to organise another one.

Discussion moved on to the question of funding. Hans Huttel spoke about Horizon 2020 calls, noting that our community had applied unsuccessfully in recent calls. This led to a discussion about whether we had chosen the right calls, and then the higher-level question of how the topics of the calls are defined. Antonio Ravara argued that we have been too passive, and that we should try to get leaders of our community onto the committees that define the funding calls. Failing that, we should try to influence people who are on the committees. This requires long-term strategy and it's too late for Horizon 2020, but we need to start thinking about the next cycle of research funding. An immediate action, which we agreed on, is to redevelop the BETTY website in order to raise the profile of the research area and community.

There was some discussion about specific areas, which have the possibility of funding, in which to try to apply behavioural types. Some members of our community had applied to Internet of Things calls, without success. Ivan Lanese noted that if we want to introduce behavioural types into IoT applications, then we need to work with people who have more practical experience with IoT; one way to start would be to invite such people to the next Dagstuhl seminar. Giovanni Bernardi mentioned a specific high-profile systems researcher at his institution, who might be a useful contact.

The final topic of discussion focussed on national funding schemes. We should all apply for national projects, and perhaps it would be possible to submit coordinated applications in more than one country to support a collaborative project. Connecting with the earlier discussion about adding session types to mainstream languages, people could apply for national projects to do that. We could also try to systematically take advantage of schemes for visiting professors and researchers, in order to arrange visits within the community. Another possibility is to explore industrial funding, such as Google's faculty grants; it was also mentioned that Mozilla are funding PhD students at Northeastern University in the USA. Finally, we in the community could support each other by sharing successful funding proposals and by providing support for individual fellowship applications from early-career researchers.

4.5 Group Discussion: Session Sharing and Races

Simon Gay (University of Glasgow, GB)

Joint work of Approximately half of the seminar participants.

There was a discussion about the problems posed by allowing sharing and races within session type systems, and various approaches to controlling these generalisations of the classical session type systems. The group produced a list of relevant references.

References

1. Damiano Mazza. *The true concurrency of differential interaction nets*. Mathematical Structures in Computer Science, 2016.
2. Stephen Brookes, Peter O'Hearn. *Concurrent Separation Logic*. ACM SIGLOG News, 2016.
3. Ilya Sergey. *Concurrent Separation Logic family tree*. http://ilyasergey.net/other/CSL-Family-Tree.pdf
4. Ralf Jung et al. *Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning*. Proceedings of POPL, 2015.
5. Tzu-chun Chen, Kohei Honda. *Specifying stateful asynchronous properties for distributed programs*. Proceedings of CONCUR, 2012.
6. Lindsey Kuper, Ryan Newton. *LVars: lattice-based data structures for deterministic parallelism*. Proceedings of FHPC, 2013.
7. Filipe Militão, Jonathan Aldrich, Luís Caires. *Composing Interfering Abstract Protocols*. Proceedings of ECOOP, 2016
8. Filipe Militão, Jonathan Aldrich, Luís Caires. *Rely-Guarantee Protocols*. Proceedings of ECOOP, 2014.
9. Luís Caires, João Costa Seco. *The type discipline of behavioural separation*. Proceedings of POPL, 2013.

4.6 Group Discussion: Standardisation of a Programming Language with Session Types

Simon Gay (University of Glasgow, GB)

Joint work of Most of the seminar participants.

At dinner the previous evening, there was a discussion between the seminar organisers (Phil Wadler, Nobuko Yoshida, Simon Gay, Vasco Vasconcelos) together with Mariangiola Dezani and Frank Pfenning. We agreed to suggest a project to integrate session types into Haskell, OCaml, Rust and Scala. This would include working with the language developers to add support for linear types in order to avoid the need for all the coding tricks we have seen in Haskell, OCaml and Scala.

The general discussion involved 30 people, who between them represented most of the topics and approaches that had been presented during the seminar. Phil Wadler opened the discussion by summarising the situation that led to the development of Haskell as a standard lazy functional language: several languages were being developed by different research groups, and the community decided that it would be productive to adopt a single common language as a platform for exploring and promoting lazy functional programming.

The group agreed that there are two distinct possibilities for developing a standard language that includes session types: (1) we develop a new language, similarly to the development of Haskell; (2) we pick an existing language (or several languages) and work with its developers to put session types into it. There was also discussion about whether standardising a programming language is the right level to work at. An alternative would be to follow the Imperial College group in using Scribble as a standard language-independent formalism for describing protocols.

Each approach has advantages and disadvantages, which were discussed thoroughly. The main advantage of working with an existing language (or languages) is that they already have programmers and communities who would be able to take advantage of session types in a familiar setting. The main disadvantage is that it might not be straightforward to combine session types, especially the necessary linear typing, with a full range of existing language features such as polymorphism. This problem could be reduced by working with Rust, which already has affine types. Frank Pfenning explained that he has already made contact with Mozilla about integrating session types into Rust, although he doesn't yet have a concrete proposal for a language extension. It was also noted that extending an existing language and getting the extension into the main release would require deep involvement in that language's community.

The main advantage of developing a new language is that the design is not constrained by existing features. Frank Pfenning has developed two small languages based on session types: SILL, and Concurrent C0. The latter was demonstrated during the seminar. The main disadvantage of developing a new language is that a great deal of engineering work is required to produce a usable full-spectrum language. This can be reduced to some extent by building on the runtime system of an existing language, as Scala did with Java. Perhaps Erlang woudl be an interesting base. Conor McBride took the view, based on his experience with dependently-typed programming and the relationship between Haskell and languages such as Agda, that the aim should not be to achieve widespread adoption of a new language; instead, success consists of features being stolen by mainstream languages.

There is a question of whether session types are a sufficiently foundational feature to justify a new language design. Concurrency is a cross-cutting concern, orthogonal to the main language paradigm, so it seems that a new language design would have to commit to one of the existing main paradigms and this decision would provoke disagreement before any work is done on session typing features. Countering this point could be the argument that session-typed concurrent programming, based on pi calculus, could be a paradigm for controlling massively parallel architectures.

It was noted that a language design from our community should have a well-specified formal semantics. Derek Dreyer (not present at the seminar) has an ERC grant to formally study the semantics and type system of Rust; we agreed that it would be useful to involve him in future meetings.

There was some discussion about the advantages and disadvantages of full type inference, with significant support for the idea that we should not aim for it. In relation to a session-typed methodology for developing distributed systems, it makes more sense to start with explicitly declared types as part of the system design. Interactive programming guided by session types has some attractions.

In the end, there was enthusiasm from around half of the people present, for the idea of developing a new language based on session types. More detailed discussion will follow in the future, and other interested people will be able to join in. It was noted that Dagstuhl has the possibility of small focussed meetings, as well as full seminars, and this could be a way of proceeding with a language design effort.

4.7 Group Discussion: Behavioural Types for Mainstream Software Development

Philipp Haller (KTH Royal Institute of Technology – Stockholm, SE)

Joint work of Approximately half of the seminar participants.

The starting point of the discussion was the question: how to support software development using behavioral types? Practical software development requires the use of widely-used programming languages, such as Scala, OCaml, or Haskell. Several approaches to encoding session types in the type systems of these languages were presented during the seminar. Thus, a natural question to ask was whether these existing implementations are "enough", or whether they have fundamental limitations that should be addressed in future work by the community. It was noted that current implementations already improve upon programming models used in industry; it would thus be worthwhile to create industrial-strength systems building upon the approaches of existing session type implementations.

Two principal approaches were identified to implementing session types for existing languages: the first approach encodes session types in the type system of an existing language; this approach requires the host language to have a sufficiently powerful type system. The second approach directly extends an existing language. It was mentioned that widely-used languages have developed to include features which support more direct encodings than are possible today in Haskell, OCaml, or Scala. For example, Rust has support for affine types, and there is at least one implementation of session types in Rust.

An important limitation of existing implementations of session types was identified, namely, useful and informative type error messages. Luca Padovani pointed out that in his OCaml implementation, if a developer does not implement an end point correctly, error messages are informative, and error locations are precise. However, when connecting two end points and a type error exists, the error may occur “far away” from where the actual problem is. It was also noted that type inference may be a source of difficulties. Making types explicit typically improves type error messages, for example, in the existing Scala library implementations. Finally, Gul Agha pointed out that most informative would be error traces corresponding to session types.

A group of discussion participants identified development tools as important for the adoption of session types by practitioners. It was noted that session types appear related to UML sequence diagrams, widely used in practice. Simon Gay pointed out that the generation of code templates can guide developers during the implementation of communication protocols. Code generation could also be integrated with modern IDEs.

On a less technical level, it was noted that establishing good feedback channels between software developers and designers of languages and libraries for programming with session types is a challenge. In this context, interaction with open-source maintainers may be an effective way to receive valuable feedback. In addition, integrating session types into open-source projects could demonstrate their value to software developers.

Education and training were also identified as important for the adoption of session types by the broader software development community. It was suggested that curricula at colleges and universities helped adoption of functional programming languages like Haskell. As a possible route discussion participants suggested the collection of patterns, inspired by the Gang-of-Four book, showing how session types or linearity help address common issues in software development. These patterns could then be taught at universities.

Modularity and reuse of session types was identified as a topic requiring further research. While FSMs are often natural for expressing the communication patterns of individual processes, their complexity can easily explode in the context of several participants. Gul Agha had explained this challenge in his presentation earlier during the seminar. Both Gul Agha and Luca Padovani pointed out that in different contexts, the same concurrent object might be used with different protocols/session types. Thus, modular specifications of session types are required which separate protocols from concurrent objects.

The discussion participants identified the following future directions. First, the development of compelling use cases that mirror modern software development; these use cases must be “complex enough” to showcase the power of session types. Second, the development of design patterns in the style of the Gang-of-Four book. Third, work on suitable abstractions, beyond state machines, that are provided to developers. Fourth, the development of a systematic approach to evaluate session types in the context of professional software development (taking open source software into account). Fifth, exchanges between industry and academia, and collaboration with industrial research labs. Finally, development of suitable concepts and curricula for teaching and education.

4.8 Group Discussion: Educational Resources for Behavioural Types

Hugo Torres Vieira (IMT – Lucca, IT)

Joint work of Approximately half of the seminar participants.

The discussion started with a short description of past experiences of teaching courses related to behavioural types.

At Imperial College London a course on concurrent programming uses LTSA (https://www.doc.ic.ac.uk/ltsa/) for the verification of systems modelled as a set of interacting finite state machines, and Java for the actual implementation. A gap between the high-level modelling and the Java implementations was identified as an issue for the students' learning experience, both conceptually and at the level of tool support.

At the University of Leicester more than one course on concurrent programming was mentioned, an optional module based on Java (offered to 3rd year BSc and MSc students) and notably an MSc course (core to several MSc degrees) that uses choreographies for system specification. CFSMs are used the model the global interaction scenario which are then used in a top-down style to obtain the local implementations. Students reacted positively to the inclusion of some encompassing theory in the latest edition, and tool support with ChorGram is a goal for the next one.

At CMU some courses related to behavioural types were mentioned, most of which using C0 http://c0.typesafety.net. In particular, the introductory course on imperative programming uses behavioural type like specifications to provide contracts written in the language itself. Some other courses that address data structures in a concurrency setting were also mentioned.

At IMT Lucca a PhD level module on type-based verification was mentioned, where behavioural types were the topic of the last few lectures.

At the University of British Columbia, at the EPFL, and at the University of Strathclyde the reported experiences with courses related to type-based verification mostly concerned sequential languages, and the relationship with behavioural types is a topic of interest for further developments.

After the report on past and ongoing experiences, some desirable future goals for our community were discussed.

Obtaining language and tool support for reducing the gap between specifications and implementations.

Creating a repository for the exchange of existing solutions to example scenarios in the existing approaches (the repository created by the ABCD project at Glasgow University, Edinburgh University and Imperial College London was mentioned as an existing resource).

Creating a repository with related documentation of courses taught at various places to serve as a reference for complementary courses. For instance, courses on designing/modelling based on behavioural types may refer to courses/material on programming based on behavioural types

Establishing a common agreement on the principles to be taught in courses related to behavioural types so that training can lead to the desired effect of creating a community of programmers with the specialized know-how (a mention to Benjamin Pierce's Software Foundations course was made, as a reference for future courses).

In the discussion it was noted that tool support in a module requires to spend time on the usability of tools. However, this time can be compensated if some tools offer some automatic marking features.

5 Open problems

5.1 A Meta Theory for Testing Equivalences

Giovanni Tito Bernardi (University Paris-Diderot, FR)

Main reference Giovanni Bernardi, Matthew Hennessy, “Mutually Testing Processes”, Logical Methods in Computer Science, 11(2:1), 2015.
URL http://dx.doi.org/10.2168/LMCS-11(2:1)2015

Testing equivalences are an alternative to bisimulation equivalence that provide in a natural way semantic models for session types. In this talk we will recall the chief ideas behind testing equivalences, along with part of the state of the art. We will also present an open problem, hopefully spurring discussion.

Participants

- Gul Agha
University of Illinois – Urbana-Champaign, US
- Nada Amin
EPFL – Lausanne, CH
- Robert Atkey
University of Strathclyde – Glasgow, GB
- Giovanni Tito Bernardi
University Paris-Diderot, FR
- Laura Bocchi
University of Kent – Canterbury, GB
- Edwin Brady
University of St. Andrews, GB
- Luis Caires
New University of Lisbon, PT
- Marco Carbone
IT University of Copenhagen, DK
- Ilaria Castellani
INRIA Sophia Antipolis, FR
- Tzu-chun Chen
TU Darmstadt, DE
- Mariangiola Dezani
University of Turin, IT
- Patrick Thomas Eugster
TU Darmstadt, DE
- Adrian Francalanza
University of Malta – Msida, MT
- Ronald Garcia
University of British Columbia – Vancouver, CA
- Simon Gay
University of Glasgow, GB
- Philipp Haller
KTH Royal Institute of Technology – Stockholm, SE
- Thomas Hildebrandt
IT University of Copenhagen, DK
- Hans Hüttel
Aalborg University, DK
- Keigo Imai
Gifu University, JP
- Dimitrios Kouzapas
University of Glasgow, GB
- Roland Kuhn
Actyx AG – München, DE
- Ivan Lanese
University of Bologna, IT
- Hugo-Andrés López
Technical University of Denmark – Lyngby, DK
- Francisco Martins
University of Lisbon, PT
- Conor McBride
University of Strathclyde – Glasgow, GB
- Hernán Melgratti
University of Buenos Aires, AR
- Fabrizio Montesi
University of Southern Denmark – Odense, DK
- J. Garrett Morris
University of Edinburgh, GB
- Nicholas Ng
Imperial College London, GB
- Dominic Orchard
University of Kent – Canterbury, GB
- Luca Padovani
University of Turin, IT
- Jovanka Pantovic
University of Novi Sad, RS
- Frank Pfenning
Carnegie Mellon University – Pittsburgh, US
- Antonio Ravara
Universidade Nova de Lisboa, PT
- Konstantinos Sagonas
Uppsala University, SE
- Alceste Scalas
Imperial College London, GB
- Nicolas Tabareau
Ecole des Mines de Nantes, FR
- Peter Thiemann
Universität Freiburg, DE
- Hugo Torres Vieira
IMT – Lucca, IT
- Emilio Tuosto
University of Leicester, GB
- Vasco T. Vasconcelos
University of Lisbon, PT
- Philip Wadler
University of Edinburgh, GB
- Nobuko Yoshida
Imperial College London, GB
- Shoji Yuen
Nagoya University, JP