Report on BCTCS 2009

Artur Czumaj, Sara Kalvala, Steve Matthews

The British Colloquium for Theoretical Computer Science (BCTCS) is an annual meeting for researchers to present their work in a wide range of areas including algorithms, logic, and the design of programming languages (www.bctcs.ac.uk) This event has always been particularly special for the emphasis it gives to supporting PhD students to present their own work and meet internationally respected researchers. For many years the UK Engineering and Physical Research Council (EPSRC) has supported students attending, and the London Mathematical Society (LMS) support a special invited speaker.

BCTCS 2009 took place from April 6-9th, hosted by the Computer Science Department of the University of Warwick and the Centre for Discrete Mathematics and its Applications (DIMAP). 97 participants produced the high quality programme of research in theoretical computer science for which the BCTCS is well known, enriched by talks from the invited distinguished speakers:

- Noga Alon, Tel Aviv University (LMS Keynote Speaker in Discrete Maths),
- Paul Goldberg, University of Liverpool,
- Andy Gordon, Microsoft Research,
- Jane Hillston, University of Edinburgh,
- Alistair Sinclair, University of California at Berkeley,
- Bill Wadge, University of Victoria

For more information on the event itself please see the web site www.dcs.warwick.ac.uk/events/bctcs. Included below are abstracts for each of the invited and contributed talks.

BCTCS 2010 will take place at the University of Edinburgh, from April 6-9th, hosted by the Computer Science Department of the University of Edinburgh, with lead organiser Dr. Julian Bradfield.

Bulletin of the EATCS no 98, pp. 181–211, February 2009

Abstracts: Invited Talks

COMBINATORIAL REASONING IN INFORMATION THEORY

Noga Alon, Tel Aviv University

Combinatorial arguments have played a crucial role in the investigation of several surprising phenomena in Information Theory. After a brief discussion of some of these results I will describe a recent example, based on joint papers with Lubetzky and Stav, and with Hassidim and Weinstein, in which properties of graph powers, colorings of Cayley graphs, and the chromatic numbers of Kneser graphs are applied in the study of a broadcasting problem with side information.

PRINCIPLES AND APPLICATIONS OF REFINEMENT TYPES

Andy Gordon, Microsoft Research

A refinement type is a type qualified by a logical constraint; an example is the type of even numbers, that is, the type of integers qualified by the is-an-even-number constraint. Although this idea has been known in the research community for some time, it has been assumed impractical, because of the difficulties of constraint solving. But recent advances in automated reasoning have overturned this conventional wisdom, and transformed the idea into a practical design principle. I will present a primer on the design, implementation, and application of refinement types. I will explain:

- How a range of diverse features may be unified as instances of the general idea of refinement types.
- How a static checker for the Oslo modeling language M allows us to check for security errors in server configurations; intended constraints on configurations are expressed with refinement types, so that configuration validation reduces to type checking.
- How we statically check integrity and secrecy properties of security critical code, such as an implementation of the CardSpace security protocol, using a system of refinement types for the F# programming language.

The lectures in this series are based on recent research with my esteemed colleagues Karthik Bhargavan, Gavin Bierman, and Cédric Fournet of MSR Cambridge, and David Langworthy of the Microsoft Connected Systems Division; much of our work relies on the excellent Z3 automated theorem prover developed by Nikolaj Bjorner and Leonardo de Moura of MSR Redmond.

RECENT PROGRESS IN COMPUTING APPROXIMATE NASH EQUILIBRIA

Paul Goldberg, University of Liverpool

In Game Theory, the Nash Equilibrium is the most prominent and well-known solution concept, mainly due to the famous result of John Nash showing that every game is guaranteed to have such a solution. Recent results suggest however that in the worst case, Nash equilibria may be computationally hard to find. A subsequent line of research has considered the problem of searching for a weaker version of this solution, an "approximate" Nash equilibrium. In this talk I will give an overview of some of these results, and discuss future directions.

PHASE TRANSITIONS AND MIXING TIMES

Alistair Sinclair, University of California at Berkeley

Recent work on random satisfiability and other problems has raised the possibility of deep connections between phase transitions (as studied in physics) and computational complexity. Markov chain Monte Carlo algorithms provide one of the most compelling examples to date of this connection. Roughly speaking, the physical notion of a phase transition frequently has a computational manifestation in the form of a sudden jump in the mixing time.

In this talk I will illustrate various aspects of the above phenomenon, with special emphasis on the classical Ising model. No knowledge of statistical physics will be assumed.

STOCHASTIC PROCESS ALGEBRA: BRINGING PERFORMANCE TO LIFE

Jane Hillston, University of Edinburgh

Stochastic process algebras emerged in the early 1990s as a novel formal description technique for performance modelling based on continuous time Markov chains (CTMCs). Enhancing classical process algebras with information about the expected duration of actions, stochastic process algebras have a clear CTMC semantics yet offer compositionality and formal manipulation techniques which are not available when performance models are constructed directly at the CTMC level. Over the ensuing decade stochastic process algebras such as PEPA enjoyed significant success in modelling and analysing a range of computer systems including software and communication systems. In the last eight years there has been considerable interest in applying these formalisms to biological applications, particularly for modelling the dynamics of intracellular processes. In this tutorial I will give an overview of stochastic process algebras and explain the attractions and challenges of using them in these diverse areas of application.

INFINITESIMAL LOGIC

Bill Wadge, University of Victoria

Infinitesimal logic is a multivalued logic in which there are discrete levels of truth and falsehood. At the top we have the Gold standard of absolute truth, but also, directly below it, Silver truth as well. Silver truth is much less true than Gold truth - in fact, infinitely less true. But Silver truth is still true, and infinitely more true than Bronze truth, just below it; and so on. At the bottom we have Gold (complete) falsity, then the much less false Silver falsity, then Bronze falsity, and so on, with a neutral value right in the middle.

This logic was first developed to give a semantics for negation as failure—for example, the negation-as-failure of Silver truth is Bronze falsity. More generally, infinitesimal logic allows us to avoid many of the paradoxes of self-reference.

On the practical side, this logic gives a semantics to constraints (such as database queries) with preferences. The most preferable answers to a query are those that satisfy the constraints at the Gold level, followed by those that satisfy it at the Silver level, then those that satisfy it at the Bronze level, and so on. Queries can be constructed with the usual Boolean connectives but also with preference sensitive alternatives; two of which can be thought of as "and, if possible" and "or, failing that".

Abstracts: Contributed Talks

PTAS FOR THE k-TOUR COVER PROBLEM ON THE EUCLIDEAN PLANE FOR MODERATELY LARGE VALUES OF k

Anna Adamaszek in joint work with Artur Czumaj and Andrzej Lingas

We are given a set P of n points in the Euclidean plane and a distinguished point O called the origin. A k-tour cover is a set of tours covering all points from P, such that each tour starts and ends in the origin and covers at most k points from P. The objective of the k-tour cover problem is to find a k-tour cover which minimizes the total length of the tours.

This problem is known to be $\mathcal{NP}$-hard. It is also known to admit constant factor approximation algorithms for all values of k and even a polynomial-time approximation scheme (PTAS) for small values of k, i.e., $k = O(\log n / \log\log n)$.

I will present a new PTAS for all values of $k \leq 2^{\log^{\delta} n}$, where $\delta = \delta(\epsilon)$. The PTAS is based on a novel reduction of the k-tour cover problem with a set of n points to a small set of instances of the problem, each with $O((k/\epsilon)^{O(1)})$ points.

FORMAL SIMULATION OF SUPERVISED COMPONENTRY MODELS AND THEIR EXECUTION

Djihed Afifi

We consider the modelling of systems that can adapt their behaviour at runtime in response to external and internal stimuli. A logical framework for such evolvable systems is introduced in [1]. In this framework, an evolvable system is modelled as a tree of components. Each component is modelled as a first order logic theory with constraints, actions and a state modelled as ground atoms. A component may be standalone or may be formed by a special pairing of supervisor and supervisee components. In this pair, the supervisor has access to the supervisee's theory and so it can trigger an evolution by altering its sub-components, its constraints or its state.

A simulator tool for this framework is under development. The simulator accepts a logical specification and executes a sequence of instructions. During execution, the component's theories must remain internally consistent. Actions pre-conditions must be met before firing actions. New constraints or components must not render the component's state conflicting.

To ensure this, the component's theories are rewritten together with theorems that test the logical obligations. These theorems are then proved on the fly using automated theorem provers. The simulator currently invokes provers that support the TPTP [2] format, such as iProver, Vampire and Paradox, with plans to support other provers, such as PVS, in the future.

In this talk I will present some of the theoretical and practical issues arising from the simulation. Different theorem provers accept first order logics with different extensions while the logic used in this framework is a typed FOL. This necessitates theory translation. Other issues include ensuring state persistence and generating minimum models.

[1] H. Barringer, D. M. Gabbay, and D. E. Rydeheard. Logical modelling of evolvable component systems: Part I - a logical framework. To appear in *Logic Journal of the IGPL*. Available from: `http://www.cs.man.ac.uk/~david/evolution/evolution.html`

[2] G. Sutcliffe and C.B. Suttner. The TPTP Problem Library: CNF Release v1.2.1. *Journal of Automated Reasoning*, 21(2):177–203, 1998.

SPANNING CONNECTIVITY GAMES

Haris Aziz in joint work with Oded Lachish, Mike Paterson and Rahul Savani

The Banzhaf index, Shapley-Shubik and other voting power indices measure the importance of a player in a coalitional game. We consider a simple coalitional game called the spanning connectivity game (SCG) based on an undirected multigraph, where edges are players. We examine the computational complexity of computing the voting power indices of edges in the SCG. It is shown that computing Banzhaf values is #P-complete and computing Shapley-Shubik indices or values is NP-hard for SCGs. Interestingly, Holler indices and Deegan-Packel indices can be computed in polynomial time. Among other results, it is proved that Banzhaf indices can be computed in polynomial time for graphs with bounded tree-width. It is also shown that for any reasonable representation of a simple game, a polynomial time algorithm to compute the Shapley-Shubik indices implies a polynomial time algorithm to compute the Banzhaf indices. This answers (positively) an open question of whether computing Shapley-Shubik indices for a simple game represented by the set of minimal winning coalitions is NP-hard.

COMPUTATION OF THE INDEX OF A COMPONENT OF NASH EQUILIBRIA FOR BIMATRIX GAMES

Anne Balthasar

In game theory, the index of a component of Nash equilibria is an important topological notion which can be used to characterize certain properties of such a component. For example, an equilibrium component is hyperstable, i.e. does not vanish under certain manipulations of the game, if and only if its index is non-zero.

For non-degenerate bimatrix games, the calculation of the index of an equilibrium is straightforward via an explicit formula, which boils down to computing determinants of square matrices. However, in degenerate cases, this formula fails to make sense, and index computation then amounts to calculating the (relatively complex) topological degree of a function. To resolve this dificulty we present an algorithm for the computation of the index in degenerate bimatrix games.

MULTIPROCESSOR SPEED SCALING FOR JOBS WITH ARBITRARY SIZES AND DEADLINES

Paul C. Bell in joint work with Prudence Wong

Energy consumption has become an important concern in the design of modern processors, not only for battery-operated mobile devices with single processors but also for server farms or laptops with multi-core processors. A popular technology to reduce energy usage is *dynamic speed scaling* where the processor can vary its speed dynamically. The power consumption is modelled by s^α when the processor runs at speed s, where α is typically 3 in reality. Running a job slower saves energy, yet it takes longer to finish the job. The study of speed scaling was initiated by Yao et al., see [1]. They studied deadline scheduling on a single processor in which jobs with arbitrary sizes and deadlines arrive online and the aim is to finish all jobs by their deadlines using the minimum amount of energy.

Albers et al. have extended the study to the multiprocessor setting in the special cases of unit-size jobs or jobs with agreeable deadlines (jobs arriving earlier have earlier deadlines), and presented constant competitive algorithms for both cases [2]. In the multiprocessor setting, in addition to determining processor speed, a job dispatching algorithm is required to assign jobs to processors. We will present results concerning generalized problems where jobs have arbitrary sizes and arbitrary deadlines. We propose a non-migratory job dispatching algorithm, called DCRR, and show that DCRR is $O(\log^\alpha P)$-competitive, where P is the ratio between the maximum and minimum job size.

[1] - S. Albers, F. Müller, S. Schmeltzer, *Speed Scaling on Parallel Processors*, In Proc. 19th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA'07), 289-298, 2007.

[2] - F. Yao, A. Demers, S. Shenker, *A Scheduling Model for Reduced CPU Energy*, Proc. 36th Annual Symp. on Foundations of Computer Science, 374-382, 1995.

RECENT THEORETICAL AND PRACTICAL DEVELOPMENTS WITH BIGRAPHS

Clive Blackwell

Bigraphs [1] are a process calculus based on category theory. A bigraph is composed of a link and place graph with the same nodes, but different edges. The link graph represents logical communication (as in the pi calculus) and the place graph models physical locality (as in the ambient calculus).

Bigraphs can be used to model security with additional reaction rules to model cryptographic and other security mechanisms, which is analogous to the extension of the pi-calculus to the spi-calculus. A system and its users are modelled as bigraphs, where bigraph reaction or rewriting rules private to the defender model the possible security mechanisms, and inverse rules model the access rights of users. The defender's objectives can be defined by bigraph invariants such as preventing unauthorised access to critical nodes, preserving nodes and channels that represent security boundaries, and limiting access to protective reaction rules.

The advantages of bigraphs, compared to most other process calculi, are the explicit representation of location and system structure that are fundamental to modelling many aspects of systems realistically. Protection mechanisms cannot remove weaknesses; only transform them. Bigraphs model protection mechanisms that control access to protected resources using private communication channels from inaccessible locations. This allows a more general and realistic adversary model than many process calculi that abstract away location. An adversary can gain new powers by acquiring access to the locations of reaction rules situated within the bigraph.

We also illustrate some recent theoretical developments of bigraphs. Several graph algorithms such as transitive closure remain polynomial time when we consider bigraphs as special types of DAGs. We need to make some reasonable constraints on the reaction rules to ensure their finite application, and store system state using special attribute nodes to avoid the unnecessary application of reaction rules.

[1] Milner, R, The Space and Motion of Communicating Agents. Cambridge University Press, publication date March 2009.

THREE WAYS TO TEST IRREDUCIBILITY

Richard P. Brent in joint work with Paul Zimmermann

We consider several algorithms for testing irreducibility of sparse polynomials over finite fields of small characteristic. The algorithms that are fastest in theory turn out not to be best in practice. A hybrid algorithm that combines the classical approach with modular composition is suggested. As an application, we describe a search for irreducible trinomials (over GF(2)) whose degree is a large Mersenne exponent.

SAFETY DOES NOT CONSTRAIN EXPRESSIVITY FOR WORD-LANGUAGES

Christopher Broadbent

Higher-order recursion schemes are systems of rewrite rules that can generate infinite trees and word-languages. *Higher-order pushdown automata (HOPDA)*, which were first introduced by Maslov [5], are devices equipped with a stack that itself contains stacks of stacks... of stacks. If recursion schemes are restricted by a constraint known as *safety*, then they have the same expressive power as HOPDA [2, 4]. Arbitrary recursion schemes require *collapsible pushdown automata* [3].

For word-languages we have shown that for any order-n unsafe recursion scheme there is an equivalent order-$(n + 1)$ *safe* recursion scheme. The argument goes via automata and extends the idea of Aehlig *et al.* used at order-2 [1]. We extend recursion schemes with 'exception handling', which bears some connections with safety. Our argument shows that order-n recursion schemes with exceptions are equi-expressive with order-$(n + 1)$ recursion schemes without.

This talk focuses on introducing higher-order (collapsible) pushdown automata together with recursion schemes (with exceptions). We will briefly sketch the idea of Aehlig *et al.*'s proof at order-2 and the difficulties that must be surmounted to obtain our generalisation.

[1] K. Aehlig, J. G. de Miranda, and C.-H. L. Ong. Safety is not a restriction at level 2 for string languages. In *Proc. FoSSaCS*, 2005.

[2] W. Damm and A. Goerdt. An automata theoretic characterisation of the oi-hierarchy. *Information and Control*, 71:1–32, 1986.

[3] M. Hague, A. S. Murawski, C.-H L. Ong, and O. Serre. Collapsible pushdown automata and recursion schemes. In *Proc. LICS*, 2008.

[4] T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In *Proc. FoSSaCS*, 2002.

[5] A. N. Maslov. Multilevel stack automata. *Problems of Information Transmission*, 12:38–42, 1976.

LONGEST PREVIOUS REVERSE FACTOR

Supaporn Chairungsee

Data compression is useful in data communication over a low-bandwidth channel and for storing documents efficiently. Lempel-Ziv factorisations yield well-known powerful technique for data compression. The Longest Previous Factor (LPF) table of a string provides an efficient way to deal with the LZ77 factorisation. It is even more efficient if the Longest Previous Reverse Factor (LPRF) table is used as it captures more information on the string. We describe new algorithms for computating the LPRF of a string by using three data structures that are suffix trie, suffix tree and suffix automaton. The last two algorithms run in linear time on a fixed size alphabet.

ALGORITHMS FOR RANDOM k-SAT

Amin Coja-Oghlan

The k-SAT problem is well known to be NP-hard for any $k \geq 3$. Among the (empirically) most challenging benchmark instances for this problem are randomly generated k-SAT formulas. In this talk I will survey various algorithms for coping with these instances. These are either simple combinatorial heuristics, algorithms based on backtracking ("DPLL"), randomized algorithms, or message passing algorithms. The quality of these algorithms can be measured in terms of the constraint density (number of clauses divided by number of variables) up to which the algorithm typically succeeds. In addition, I'll present a new algorithm that succeeds up to a constraint density that has been suggested as a natural barrier for efficient algorithms to find satisfying assignments.

GENERATING AND COUNTING EULER TOURS OF RANDOM REGULAR DIGRAPHS
Páidí Creed

A graph (resp. digraph) is said to be Eulerian iff all vertices are of even degree (resp. all vertices have the same in-degree and out-degree). Every Eulerian graph G has a set of circuits in which each edge is used exactly once, known as the Euler tours of G. This talk is about the closely related problems of counting and sampling Euler tours of an Eulerian graph.

For any directed Eulerian graph G, we can count the number of Euler tours of G in polynomial time and there exist several polynomial time algo- rithms for sampling from the uniform distribution on the Euler tours of G. However, the complexity of counting Euler tours of an undirected Eulerian graph is #P-complete. Moreover, none of the sampling algorithms for the di- rected case can be applied to sampling Euler tours of an undirected Eulerian graph.

In this talk, I present a result characterising the asymptotic distribution of the number of Euler tours of a random r-in/r-out graph. This can be used to show that a simple algorithm can be used to sample or approximately count Euler tours of almost every r-in/r-out graph in expected polynomial time. This algorithm can also be used to sample or approximately count Euler tours of an undirected Eulerian graph, and I will briefly mention work underway towards generalising this result to the undirected case. In both cases, the approach used is the method of conditioning on small subgraph counts pioneered by Robinson and Wormald in their proof that almost every regular graph admits a Hamiltonian cycle. In particular, this can be seen as an analogue of Frieze et al's work on generating and counting Hamiltonian cycles of random regular graphs.

THE COMPLEXITY OF COUNTING INDEPENDENT SETS MODULO k, WITH APPLICATIONS TO CSP
John Faben

In 1979, Valiant introduced the complexity class $\bigoplus$P, the problem of counting the number of solutions to NP problems modulo two, and has since proved some completeness results. In this talk we define the notion of completeness for counting modulo integers other than 2, and we consider the complexity of counting the number of independent sets in a graph in this sense. In fact, we will prove that this problem is $\#_k$P-complete for all k even if the graphs are restricted to be bipartite.

This was a preliminary result in proving a dichotomy theorem for the complexity of counting the number of solutions to Boolean Constraint Satisfaction Problems modulo integers. That result builds on an earlier paper of Creignou and Hermann which gave a counting dichotomy for these types of problem, and the dichotomy itself is almost identical. Specifically, we have found that counting the number of solutions to a Boolean Constraint Satisfaction Problem can be done in polynomial time if all the relations are affine. Otherwise, except for one special case with $k = 2$, it is $\#_k$P-complete.

LCP ALGORITHMS FOR DISCOUNTED GAMES

John Fearnley in joint work with Marcin Jurdziński and Rahul Savani

We study the performance of Lemke's algorithm and the Cottle-Dantzig algorithm for P-Matrix LCPs is studied for the instances produced by the reduction from discounted games given by Jurdziński and Savani. Both algorithms are described purely in terms of the original discounted game, bypassing the reduction itself. A discounted game is given for which both the algorithms take an exponential number of steps, indicating that the algorithms perform no better for discounted games than they do for general P-matrix LCPs.

DESCRIPTIVE COMPLEXITY OF OPTIMISATION PROBLEMS

James Gate in joint work with Iain Stewart

The field of Descriptive Complexity is the bridge between finite model theory and algorithmic complexity theory. The majority of research in this field has focused on classes of decision problems and the logics that capture them. This talk looks at how to extend these logics to capture optimisation problems. Specifically, it shall examine the class of (deterministic) polynomial time optimisation problems (referred to as P_{opt}) and argue that a single logical framework, which does not discriminate between maximisation and minimisation problems, is the most appropriate way to capture this class. Such a framework, using fixed-point operators along with examples of their use, shall be presented.

EMBEDDING A FUNCTIONAL HYBRID MODELLING LANGUAGE IN HASKELL

George Giorgidze in joint work with Henrik Nilsson

Functional Hybrid Modelling (FHM) is a new approach to the design of non-causal modelling languages. The idea is to enrich a purely functional language with a few key abstractions for supporting hybrid, non-causal modelling. Our hypothesis is that the FHM approach will result in non-causal modelling languages that are relatively simple, have clear, purely declarative semantics, and, aided by this, advance the state of the art by supporting e.g. certain forms of meta-modelling and modelling and simulation of highly structurally dynamic systems. In this talk we present the first investigation into the implementation of an FHM language for non-causal modelling and simulation of physical systems. This is realised as a domain-specific language embedded in Haskell. The language facilitates construction and composition of model fragments given by systems of implicit differential algebraic equations. The method of embedding employs quasiquoting, thus demonstrating the effectiveness of this approach for languages that are not suitable for embedding in more traditional ways. Our implementation is available on-line, and thus the first publicly available prototype implementation of an FHM language.

A COMPLEXITY DICHOTOMY FOR HYPERGRAPH PARTITION FUNCTIONS

Leslie Ann Goldberg in joint work with Martin Dyer and Mark Jerrum

The talk will introduce *partition functions*, which arise in many computational contexts, and will discuss the complexity of computing them. It will explain what a *dichotomy theorem* is, and why we want such theorems (essentially, we want them so that we can better understand the boundary between the class of easy-to-compute functions and the class of functions that cannot be efficiently computed).

The particular technical problem which will be discussed, to some extent, is the complexity of counting homomorphisms from an r-uniform hypergraph G to a symmetric r-ary relation H. (But you don't need to know anything about that to understand the talk!) We give a dichotomy theorem for $r > 2$, showing for which H this problem is in FP and for which H it is $\#\mathsf{P}$-complete. Our dichotomy theorem extends to the case in which the relation H is weighted, and the partition function to be computed is the sum of the weights of the homomorphisms. This problem is motivated by statistical physics, where it arises as computing the partition function for particle models in which certain combinations of r sites interact symmetrically.

LOGICS AND BISIMULATION GAMES FOR CONCURRENCY, CAUSALITY AND CONFLICT

Julian Gutierrez

Based on a simple axiomatization of concurrent behaviour we define two ways of observing parallel computations and show that in each case they are dual to conflict and causality, respectively. We give a logical characterization to those dualities and show that natural fixpoint modal logics can be extracted from such a characterization. We also study the equivalences induced by such logics and prove that they are decidable and can be related with well-known bisimulations for interleaving and noninterleaving concurrency. Moreover, by giving a game-theoretical characterization to the equivalence induced by the main logic, which is called Separation Fixpoint Logic (SFL), we show that the equivalence SFL induces is strictly stronger than a history-preserving bisimulation (hpb) and strictly weaker than a hereditary history-preserving bisimulation (hhpb). Our study considers branching-time models of concurrency based on transition systems and petri net structures.

ATTACKING AES VIA SAT

Matthew Gwynne in joint work with Oliver Kullmann

We consider the translation of the AES (the "Advanced Encryption Standard", the successor of DES) into SAT.

In principle, many different questions regarding AES could be solved by a

SAT solver, such as the existence of weak keys, or questions regarding various uniqueness conditions, and therefore such a translation offers a perspective on the cryptanalysis of AES through the lens of satisfiability.

It seems likely, from similar work in the past with other ciphers (for example with DES), that one cannot expect too much at this time, and therefore we emphasise a modular open-source approach (as part of the OKlibrary - http://ok-sat-library.org - based on the Maxima/Lisp part of the library) which allows for small-scale variations, other experiments (for example replacing the S-box with other random permutations) and using general constraints as a target language (including polynomial equations and term based representations).

Such an approach will also allow us to integrate our methods with the approaches taken in the area using Groebner bases.

FLIPPING REGULAR GRAPHS AND A PEER TO PEER NETWORK

Andrew Handley in joint work with Martin Dyer and Colin Cooper

Mahlmann and Schindelhauer [1] defined a network that relies on random flip operations to keep its topology random regular, allowing it to repair damage and to embed new peers without over-complicated joining schema. It is then important to the protocol that the flip Markov chain mixes quickly enough. Work by Cooper, Dyer and Greenhill [2] gave a bound on the mixing time of the similar switch Markov chain, and this result was extended to acquire a loose polynomial bound on that of the flip Markov chain.

We dramatically tighten the mixing time bound using a two-stage direct canonical path construction. We go on to explore the behaviour of the protocol using simulations.

[1] P. Mahlmann and C. Schindelhauer. Peer-to-peer networks based on random transformations of connected regular undirected graphs. In *SPAA 05: Proceedings of the seventeenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 155-164, New York, NY, USA, 2005. ACM. `http://portal.acm.org/citation.cfm?id=1073992`

[2] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. *Combinatorics, Probability and Computing*, 16(4):557-593, 2007. `http://portal.acm.org/citation.cfm?id=1070574`

PARAMETERIZED COMPLEXITY CLASSES UNDER LOGICAL REDUCTIONS

Yuguo He in joint work with Anuj Dawar

In the theory of parameterized complexity, the W-hierarchy plays a role similar to NP in classical complexity theory in that many natural parameterized problems are shown intractable by being complete for some level $W[t]$ of the hierarchy. The classes $W[t]$ were originally defined as the sets of problems reducible to certain natural complete problems by means of fixed-parameter tractable (*fpt*) reductions. We investigate whether the classes can be characterised by means of weaker reductions, just like NP can. The latter is known to admit complete problems even under quantifier-free first-order projections.

We consider reductions defined in terms of first-order interpretations and introduce a number of parameterized versions of these. Our main result is that each class $W[t]$ has complete problems under *slicewise bounded-variable first-order* reductions. These are a natural weakening of the slicewise bounded-variable LFP reductions which, by a result of Flum and Grohe [1], are known to be equivalent to *fpt*-reductions. If we relax the restriction on having a bounded number of variables, we obtain *slicewise first-order* reductions, which are not necessarily *fpt*. Indeed, we are able to show that any problem in $W[t]$ is reducible to some problem in $W[1]$ under such reductions—a result which if it held for *fpt*-reductions would imply the collapse of the W-hierarchy. On the other hand, we show that if we consider *slicewise quantifier-free first-order* reductions, they are considerably weaker in that there are problems in $W[t+1]$ that cannot reduce to any problem in $W[t]$ under such reductions—a result which if it held for *fpt*-reductions would imply the strictness of the W-hierarchy and therefore the separation of P from NP.

[1] J. Flum and M. Grohe. Fixed-parameter tractability, definability, and model checking, in *SIAM Journal on Computing* 31, 2001.

EXPRESSIVE POWER OF RANK LOGICS

Bjarki Holm in joint work with Anuj Dawar, Bastian Laubner and Martin Grohe

Descriptive complexity theory studies the relationship between logic and computational complexity. One of the main open questions in this area is whether there exists a logic that can express exactly all the polynomial-time computable properties of finite structures. The work of Cai et al. [2], and later Gurevich and Shelah [3], established that fixed-point logic with counting (FP+C) is not expressive enough for this purpose. Atserias et al. [1] later showed that FP+C has the further limitation that it cannot determine the solvability of systems of linear equations, a natural polynomial-time problem.

We show that all the known shortcomings of the logic FP+C relate to its inability to determine the solvability of systems of linear equations; or more generally, its inability to find the row rank of a matrix. This leads us to consider extensions of first-order and fixed-point logics via operators for computing the row rank of definable matrices. We show that fixed-point logic with rank (FP+R) is strictly more expressive than FP+C and it is an open question whether FP+R can express all polynomial-time properties of finite structures. We also consider the expressive power of first-order logic with rank (FO+R) and show that in the presence of a linear ordering this logic captures the complexity class $\oplus$L, whose descriptive complexity had been previously unknown.

[1] A. Atserias, A. Bulatov, and A. Dawar. Affine systems of equations and counting infinitary logic. In *Proc. 34th International Colloquium on Automata, Languages and Programming*, volume 4596 of *Lecture Notes in Computer Science*, pages 558–570, 2007.

[2] J. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

[3] Y. Gurevich and S. Shelah. On finite rigid structures. *Journal of Symbolic Logic*, 61:549–562, 1996.

A HIGHER-ORDER OBSERVATIONAL EQUIVALENCE MODEL CHECKER

David Hopkins in joint work with Luke Ong

We have developed a tool, Homer, which can check for observational equivalence of programs from the third-order fragment of Idealized Algol augmented with iteration. *Idealized Algol* (IA) is a functional programming language with imperative features, such as sequencing and local variables. It is effectively a call-by-name variant of core ML. *Observational equivalence* is a powerful notion of program equivalence. Two programs are equivalent when one can be substituted for the other in every program context without causing any observable difference in the outcome of the computation. To our knowledge Homer is the first model checker for third-order programs.

Homer relies on the fully abstract *game semantics* of IA. In games semantics a type T is represented by a two-player game $[\![T]\!]$. A term in context, $\Gamma \vdash M : T$, is then interpreted as a strategy, $[\![M]\!]$, for the game $[\![\Gamma \vdash T]\!]$. A powerful result of the game semantics is that two terms are observationally equivalent if and only if their respective strategies contain the same set of complete plays (a play is complete when it has been played to termination).

Homer has two main components. The first maps a term M to a *Visibly Pushdown Automaton* (VPA) which represents the strategy denotation $[\![M]\!]$. VPA are a subclass of pushdown automata in which the stack action (push, pop or neither) is uniquely determined by the input symbol. Remarkably, VPA have closure properties almost as nice as the regular languages. The language accepted by the VPA produced is a precise representation of the set of complete plays in $[\![M]\!]$. A play is a (suitably constrained) sequence of moves, each equipped with a justification pointer to an earlier move. By ignoring the pointers, a set of plays naturally forms a language. Unfortunately, for third-order terms, doing so results in a loss of precision, so we have to add tags to certain moves to encode where the pointers should go.

The second main component of Homer is a VPA toolkit which we use to check for equivalence of VPA. Using complementation, intersection and an emptiness test we can check if two VPA accept the same language. If they are inequivalent, Homer produces as counter-examples both a game-semantic play and an operational-semantic separating context (a context which terminates given one of the terms, but diverges with the other).

THE COMPLEXITY OF WEIGHTED BOOLEAN #CSP WITH MIXED SIGNS

Markus Jalsenius in joint work with Andrei Bulatov, Martin Dyer, Leslie Ann Goldberg and David Richerby

We give a complexity dichotomy for the problem of computing the partition function of a weighted Boolean constraint satisfaction problem (denoted weighted #CSP). Such a problem is parameterized by a set of rational functions, each of which assigns a weight to any configuration. A configuration is an assignment of values from $\{0, 1\}$ to the variables in the instance, and the partition function is the sum of weights of all configurations. Our dichotomy extends previous work in which the weight functions were restricted to being non-negative. This extension is of particular interest because functions having mixed signs can cause cancellations in the partition function, and hence may make it easier to compute. We show that a weighted #CSP problem is either in FP (easy) or is $\mathsf{FP}^{\#\mathsf{P}}$-complete (hard). Not only do we have this dichotomy, it is also decidable.

In this talk we give an introduction to weighted #CSPs and and explain how many natural problems can be expressed as weighted #CSP problems with functions of mixed signs. We also give a characterisation of weighted #CSPs in order to determine whether the problem is in FP or is $\mathsf{FP}^{\#\mathsf{P}}$-complete.

VERIFYING TRAIN CONTROL SOFTWARE

Phillip James in joint work with Markus Roggenbach

Recently, in [1] K. Kanso developed and implemented a method to verify the control software of train stations. Given a program written in so-called ladder logic [2], general safety conditions would be broken down for the concrete layout of a train station, and—using SAT solving techniques [3]—it was verified if the program respected the conditions. We develop this approach further: first we turn the software from an application dedicated to a single train station into a generic product. Secondly, in the case the verification of a safety condition fails, we produce traces of counter examples, which shall help the engineer to locate the potential fault within the program.

[1] Karim Kanso. MRes Thesis, Formal Verification of Ladder Logic. Swansea University 2008.

[2] Programmable controllers – Part 3: Programming languages, IEC Standard 61131-3, IEC 2003.

[3] Oliver Kullmann: OKLibrary – A library for SAT solving. `http://www.ok-sat-library.org/`.

A COMPLEXITY DICHOTOMY FOR PARTITION FUNCTIONS WITH MIXED SIGNS

Mark Jerrum in joint work with Leslie Ann Goldberg, Martin Grohe and Marc Thurley

Partition functions, also known as homomorphism functions, form a rich family of graph invariants that contain combinatorial invariants such as the number of k-colourings or the number of independent sets of a graph and also the partition functions of certain "spin glass" models of statistical physics such as the Ising model.

Building on earlier work by Dyer and Greenhill, and Bulatov and Grohe, we completely classify the computational complexity of partition functions. Our main result is a dichotomy theorem stating that every partition function is either computable in polynomial time or #P-complete. Partition functions are described by symmetric matrices with real entries, and we prove that it is decidable in polynomial time in terms of the matrix whether a given partition function is in polynomial time or #P-complete.

While in general it is very complicated to give an explicit algebraic or combinatorial description of the tractable cases, for partition functions described by a Hadamard matrices — these turn out to be central in our proofs — we obtain a simple algebraic tractability criterion, which says that the tractable cases are those "representable" by a quadratic polynomial over the field GF(2).

PROPERTY VERIFICATION OF AN ELECTRONIC PAYMENT SYSTEM: EP2

Temesghen Kahsai

The EP2 system is an electronic payment system and it stands for 'EFT/POS 2000' short for 'Electronic Fund Transfer/ Point of Service 2000', is a joint project established by a number of (mainly Swiss) financial institutes in order to define the infrastructure for credit, debit and electronic purse terminals in Switzerland (`www.eftpos2000.ch`). The system consist of seven autonomous entities and they are centered around an *EP2 Terminal*. These entities communicate with the *Terminal* and, to a certain extent, with another via XML-messages in a fixed format. Each component is a reactive system defined by a number of use cases. The EP2 specification consists of 12 documents, each of which describe the different components or some aspect common to the components.

In this talk I will show the formalization of the EP2 specification in the formal specification language Csp-Casl [3]. Csp-Casl allows to formalize computational system in a combined algebraic / process algebraic notation. In [1] we introduced refinement notions for Csp-Casl. We verify the refinement of the different level of the EP2 specification and we prove some properties such as deadlock and livelock freedom using the interactive theorem prover Csp-Casl-Prover [2].

[1] Temesghen Kahsai and Markus Roggenbach. Refinement notions for CSP-CASL. In Andrea Corradini and Fabio Gadducci, editors, *WADT 2008 – Preliminary Proceedings*, Technical Report: TR-08-15, pages 15–16. 2008.

[2] L. O'Reilly, Y. Isobe, and M. Roggenbach. CSP-CASL-Prover – a generic tool for process and data refinement. In *AVOCS08*, 2008.

[3] Markus Roggenbach. CSP-Casl – A new integration of process algebra and algebraic specification. *Theoretical Computer Science*, 354:42–71, 2006.

AUTOMATED GENERATION OF VERIFIED RAILWAY INTERLOCKING SYSTEMS

Karim Kanso in joint work with Anton Setzer and Peter Mosses

In recent years the use of safety critical computer controlled industrial systems has increased. The railway domain is no exception. These large safety critical systems are very hard to produce requiring many man hours during development and testing. This project aims to automate the development and verification of railway interlocking systems, such that formal proofs are produced ensuring safety and liveliness properties. Briefly, this entails specifying and modelling the railway (Railway Domain Model) as done with domain engineering. Then, the next step is to create a view of this model for the interlocking system in question. Different views of this model can be taken for different interlocking systems or entirely different purposes.

AN ALGORITHM FOR FINDING k-VERTEX OUT-TREES AND ITS APPLICATION TO k-INTERNAL OUT-BRANCHING PROBLEM

EunJung Kim in joint work with Nathann Cohen, Fedor V. Fomin, Gregory Gutin, Saket Saurabh and Anders Yeo

An *out-tree* is an oriented tree with only one vertex of in-degree zero called the *root*. The k-Out-Tree problem is the problem of deciding for a given parameter k, whether an input digraph contains a given out-tree with $k \geq 2$ vertices. In their seminal work on Color Coding Alon, Yuster, and Zwick provided fixed-parameter tractable (FPT) randomized and deterministic algorithms for k-Out-Tree. While Alon, Yuster, and Zwick only stated that their algorithms are of runtime $O(2^{O(k)}n)$, however, it is easy to see that their randomized and deterministic algorithms are of complexity $O^*((4e)^k)$ and $O^*(c^k)$, where $c \geq 4e$.

The main results of Alon, Yuster, and Zwick, however, were a new algorithmic approach called Color Coding and a randomized $O^*((2e)^k)$ algorithm for deciding whether a digraph contains a path with k vertices (the k-Path problem). Recently Chen et al. and Kneis et al. developed an approach, called Divide-and-Color, that allowed them to design a randomized $O^*(4^k)$-time algorithm for k-Path. Divide-and-Color in Kneis et al. (and essentially in Chen et al.) is 'symmetric', i.e., both colors play similar role and the probability of coloring each vertex in one of the colors is 0.5. We further develop Divide-and-Color by making it asymmetric, i.e., the two colors play different roles and the probability of coloring each vertex in one of the colors depends on the color. As a result, we refine the result of Alon, Yuster, and Zwick by obtaining randomized and deterministic algorithms for k-Out-Tree of runtime $O^*(5.7^k)$ and $O^*(5.7^{k+o(k)})$ respectively.

We apply the above deterministic algorithm to obtain a deterministic algorithm of runtime $O^*(c^k)$, where c is a constant, for deciding whether an input digraph contains a spanning out-tree with at least k internal vertices. This answers in affirmative a question of Gutin, Razgon and Kim (Proc. AAIM'08).

ON COMPUTATIONAL EXPERIMENTS WITH ARITHMETIC PROGRESSIONS

Oliver Kullmann

How large needs the natural number n be, so that, however $\{1, \dots, n\}$ is partitioned into two parts, at least one part will contain an arithmetic progression of length 3? $n = 9$ is sufficient, while $n = 8$ won't do (an instructive exercise for the reader).

This is an example for the smallest non-trivial *van der Waerden number*, and these numbers have been studied extensively since van der Waerden proved their existence in 1927. Computing precise values is a daunting task, and we give an overview on what has been achieved (and how).

Special emphasise is put on the considerations put forward by additive number theory, which in this context replaces the set $\{1, \ldots, n\}$ by the set of the first n prime numbers and otherwise asks the same questions, based on the Green-Tao theorem (2004), and I will give first experimental results on these *Green-Tao numbers*.

The main computational methods are based on SAT solving, in the context of the `OKlibrary` (`http://www.ok-sat-library.org/`). Depending on the parameter values, very different approaches perform best, which is an interesting topic in itself.

CONSISTENCY STATEMENTS IN THE FRAGMENTS OF EQUATIONAL THEORY PV

Ebrahim A Larijani

Bounded Arithmetic is a subsystem of Peano Arithmetic which is strongly related to the computational complexity classes. Functions which are computable by polynomial Turing Machine are exactly the functions which are definable in the specific theories of Bounded Arithmetic. Consequently by separating theories in the hierarchy of Bounded Arithmetic we could understand the situation in the separation of Polynomial Hierarchy namely $P \neq NP$ which is the main open problem in the computational complexity theory.

In this talk I will explain one approach to the separation problem of Bounded Arithmetic based on examining consistency statements in the weak theories of arithmetic and I will discuss possible ways of improving results concerning consistency of fragments of equational theory PV.

UML SPECIFICATION AND CORRECTION OF OBJECT-ORIENTED ANTIPATTERNS

Maria Teresa Llano in joint work with Rob Pooley

Nowadays, the detection and correction of software defects has become a very hard task for software engineers. Due to the constant evolution of the industry, the technology and systems that support its operation are required to fit into a constantly changing environment. Most importantly, the lack of standard specifications of these software defects alongside with the lack of tools for their detection, correction and verification enforces developers to perform manual modifications, incurring not only in mistakes, but also in costs of time and resources.

The work presented here is aimed at the study of the specification and correction of a particular type of software defects: *Object-Oriented Antipatterns*. More specifically, we have defined a UML specification of antipatterns and established guidelines for their correction process through the use of rewrite rules. With this specification we expect to open up the possibility to auto- mate the detection and correction of this kind of software defects.

FPT ALGORITHMS FOR THE MAXIMUM INDEPENDENT SET AND MAXIMUM CLIQUE PROBLEMS

Vadim V. Lozin

We study the MAXIMUM INDEPENDENT SET and MAXIMUM CLIQUE problems parameterized by the solution size k. A parameterized problem is *fixed-parameter tractable* (fpt for short) if it can be solved in $f(k)n^{O(1)}$ time, where $f(k)$ is a computable function depending on the value of the parameter only. In general, both problems are W[1]-hard, which means they are not fixed-parameter tractable unless $P = NP$. On the other hand, fpt-algorithms have been developed for the MAXIMUM INDEPENDENT SET problem in the classes of triangle-free graphs, graphs of bounded vertex degree, segment intersection graphs with bounded number of directions, planar graphs, and more generally, graphs excluding a single-crossing graph as a minor [2]. Therefore, the MAXIMUM CLIQUE problem admits fpt-algorithms in the complements of these graphs.

A common feature of all these classes is that all of them are hereditary (i.e., closed under vertex deletion) and all of them are small in the following sense. It is known [1] that for every hereditary class X, the number X_n of n-vertex graphs in X (also known as the speed of X) satisfies $\lim_{n\to\infty} \frac{\log_2 X_n}{\binom{n}{2}} = 1 - \frac{1}{k(X)}$, where $k(X)$ is a natural number called the *index* of the class. The triangle-free graphs have index 2 and the index of all other classes mentioned above is 1.

We focus on hereditary classes of index $k > 1$. Each class in this range can be approximated by a minimal class of the same index. Our main result is that both problems are fixed-parameter tractable in *all* minimal classes of index k for *all* values of k.

[1] J. Balogh, B. Bollobás, D. Weinreich, The speed of hereditary properties of graphs, *J. Combin. Theory*, Ser. B 79 (2000) 131–156.

[2] E.D. Demaine, M. Hajiaghayi, D.M. Thilikos, Exponential speedup of fixed-parameter algorithms for classes of graphs excluding single-crossing graphs as minors, *Algorithmica*, 41 (2005) 245–267.

KEEPING PARTNERS TOGETHER: ALGORITHMIC RESULTS FOR THE HOSPITALS/ RESIDENTS PROBLEM WITH COUPLES

David Manlove in joint work with Eric McDermid

The classical Hospitals / Residents problem (HR) has many practical applications: in particular it models the assignment of junior doctors to hospitals, which is carried out by large-scale centralised matching schemes in many countries. The Hospitals / Residents problem with Couples (HRC) is a generalisation of HR that models the important case where couples submit joint preference lists over pairs of hospitals (h_i, h_j) that are typically geographically close.

We consider a natural restriction of HRC in which the members of a couple wish to be placed at the *same* hospital, i.e., $h_i = h_j$ for every such pair. We show that, in this context, the problem of deciding whether a stable matching exists is NP-complete, even if each resident's preference list is of length at most 3 and each hospital has capacity at most 2. However we show that if each hospital's preference list is of length at most 2, then a stable matching always exists and can be found in linear time (for arbitrary hospital capacities).

We also consider a more general restriction of HRC in which the members of a couple have individual preference lists over hospitals, and the joint preference list of the couple is *consistent* with these individual lists in a precise sense. In this context, with respect to classical (Gale-Shapley) stability, we give a linear-time algorithm to find a stable matching or report that none exists, regardless of the preference list lengths or the hospital capacities.

NEW PARAMETERIZED COMPLEXITY RESULTS FOR PROBLEMS ON DEGENERATE GRAPHS

Luke Mathieson in joint work with Stefan Szeider

We present some new parameterized results on the theme of degree constrained graphs, in particular degenerate graphs. We show that restricting inputs of problems to degenerate graphs often results in fixed-parameter tractable algorithms. Previously we had demonstrated the fixed-parameter tractability of a family of problems involving editing to achieve a regular graph, in the case of degenerate graphs we show that such editing problems are W[P]-complete, even when the desired degeneracy is a constant, and are thus unlikely to be fixed-parameter tractable. Furthermore we show that there also exist simple problems that remain hard when restricted to degenerate graphs, even when the degeneracy is a constant.

A $\frac{3}{2}$-APPROXIMATION ALGORITHM FOR GENERAL STABLE MARRIAGE

Eric J. McDermid

An instance of the *stable marriage problem with ties and incomplete lists* (SMTI) involves a set of n men and n women, each of whom provides a ranking of a subset of the members of the opposite sex in the form of a preference list. A man or a woman's preference list may contain *ties*, which are sets of agents all having the same rank. A matching M of men to women is *stable* if there is no pair $(m, w) \notin M$ such that m and w prefer each other to their situation in M. It is known that every SMTI instance admits at least one stable matching, however stable matchings can have different cardinalities. It is APX-hard to compute a maximum cardinality stable matching, but there have recently been proposed polynomial-time approximation algorithms, with constant performance guarantees for both the general version of this problem, and for several special cases. Our contribution is to describe a $\frac{3}{2}$-approximation algorithm for the general version of this problem, improving upon the recent $\frac{5}{3}$-approximation algorithm of

Király. Interest in such algorithms arises because of the problem's application to centralized matching schemes, the best known of which involve the assignment of graduating medical students to hospitals in various countries.

SOLVING SIMPLE STOCHASTIC GAMES WITH INTERIOR POINT METHODS
Julian Merschen

We introduce simple stochastic games (SSG) and give reasons why it is of interest to find the optimal strategy when considering a one-player SSG. In this setup, the optimal strategy can be found by solving a linear program (LP). We show that the constraint matrix of the corresponding linear program is an integer matrix. We introduce Vavasis' and Ye's layered interior point algorithm, the running time of which is polynomial and only depends on the encoding of the constraint matrix of the corresponding LP. As the encoding of the corresponding linear problem is polynomially bounded in the dimension of the SSG this known algorithm solves the LP in strongly polynomial time. Next we turn to solving SSG when both players are present. It is known that this problem can be rewritten as a linear complementarity problem with a P-matrix (LCPP). Under certain conditions LCCP can be solved in polynomial time using interior point methods. We analyze the feasibility of these conditions when solving SSG with two players.

THE ITERATED PRISONER'S DILEMMA ON A CYCLE
Velumailum Mohanaraj in joint work with Martin Dyer

Prisoner's dilemma is a two-person game widely used as a metaphor for the evolution of cooperation among selfish agents. Many strategies have been studied in this context. A particular strategy, called Pavlov, has been shown to have some advantages. However, the Pavlov strategy leaves room to be exploited. We modify this strategy by introducing some stochasticity, thereby reducing the possibility of exploitation. We call the resulting strategy Rational Pavlov. This has a parameter p which measures the "degree of forgiveness" of the players. We study the evolution of cooperation in the Iterated Prisoner's Dilemma game, when n players are arranged in a cycle, and all play this strategy. We examine the effect of varying p on the time taken for complete cooperation to emerge. We prove that the convergence rate is fast, $O(n \log n)$ time, for high values of p, but exponentially slow in n for low values of p. Our analysis leaves a gap in the range of p, but simulations suggest that there is, in fact, a sharp phase transition.

MODULAR TYPE SYSTEMS
Mark New

Type systems for programming languages are commonly specified using a big-step variant of the structural operational semantics (SOS) framework. There are two ways in which a type system can be specified: declaratively, or algorithmically – the latter is 'syntax directed', with the rules corresponding directly to a particular type checking algorithm.

We are investigating a component-based approach to language description, which involves analysing the constructs of each language in terms of abstract, language-independent constructs. For each abstract construct there should be a simple and unique typing rule, which should not require reformulation when constructs are combined. It appears that this requires declarative typing rules. However, it has been argued that certain aspects of programming languages such as Java can only be specified algorithmically.

After outlining the differences between algorithmic and declarative typing, this talk will discuss potential solutions to the problem outlined above.

A NEW APPROACH FOR AUTOMATA COORDINATION ON $\mathbb{Z}^2$

Thomas Nickson in joint work with Igor Potapov and Russell Martin

Swarm robotics is generally categorised by simple robots, usually of unknown quantity, with access only to local information, limited resources and each with the same algorithm. The aims are to produce a collected response from robots that are in essence oblivious to the global aims which they are a part of. In this talk a method for such coordination is described. Automata are modelled on an integer grid as a cluster of unknown size and shape. Inspiration is taken from harmonic resonance phenomena which exemplifies how complex and consistent global results may be formed from only local interactions. With an initial case of two robots producing a regular pattern of waves, showing a variety of relatively complex patterns, through superposition of these waves, can be produced on a plane of very simple robots with the ability to communicate with others in their immediate neighbourhood. These patterns create points of reference and a breach of symmetry throughout the system which can be exploited for a number of organisational duties including orientation, leader election and many other situations in which a break in symmetry can be exploited. Essentially this algorithm can be seen as a form of preprocessing easing the difficulties of other operations required to manipulate the robots. Extensions to cellular automata may also be explored.

LUCIAN : DATAFLOW AND OBJECT-ORIENTATION

Dominic Orchard

There are a multitude of programming languages in existence. Why? Because no single language or paradigm can be all things to all people. Whilst one class of programs may be succinctly written, easily reasoned about, and efficiently compiled within one language another class of programs may be impenetrably intractable for the human and the compiler. Language interoperation and multi-paradigm languages are the means by which programmers can get the best of both worlds.

This talk introduces the Lucian programming language, a cross-paradigm derivative of the Lucid dataflow language, that interoperates declarative dataflow and object-orientation. Programs that are dynamic or reactive can be succinctly expressed within Lucid. However, not all parts of such a program are necessarily easy to express in dataflow form. Lucian provides an escape to an imperative object-oriented language where subparts of a program may be more easily written and compiled. Conversely Lucian provides a way to write programs using objects in Lucid's dataflow equation style.

This talk introduces Lucid for the uninitiated viewer and proceeds to introduce the central constructs of Lucian. A comparison of the underlying computational models of Lucid and object-orientation is given to explain the appropriateness of this interoperation.

STRUCTURED THEOREM PROVING FOR Csp-Casl

Liam O'Reilly in joint work with Markus Roggenbach

At the last WADT T. Mossakowski and M. Roggenbach [1] suggested Csp-Casl[2] as an institution, this construction gives rise to the possibility of structure Csp-Caslspecifications.

In our talk we will discuss how to implement the structuring mechanisms for Csp-Casland demonstrate how to use these structuring mechanisms for the compositional verification of systems specified in Csp-Casl.

[1] Till Mossakowski and Markus Roggenbach. An institution for processes and data. In Andrea Corradini and Fabio Gadducci, editors, *WADT 2008 – Preliminary Proceedings*, Technical Report: TR-08-15, pages 13–14. Universita Di Pisa, Dipartimento Di Informatica, 2008.

[2] Markus Roggenbach. CSP-CASL - a new integration of process algebra and algebraic specification. *Theoretical Computer Science*, 354(1):42–71, 2006.

COUNTING INTERVAL SIZES VS. COUNTING NONDETERMINISTIC COMPUTATION PATHS

Aris Pagourtzis in joint work with E. Bampas, A. Göbel, and A. Tentes

We investigate the complexity of hard counting problems that belong to the class #P but have easy existence version; several well-known problems such as #Perfect Matchings, #DNFSat share this property. We focus on classes of such problems which emerged through two disparate approaches. In the first one, taken by Hemaspaandra *et al.* [1], they define classes of functions that count the size of intervals of ordered strings, where the underlying order is assumed to be polynomial-time decidable. They consider both partial and total orders, and they also investigate the case where the adjacency query on the underlying order is decidable in polynomial-time. They characterize #P as the class of functions that count the size of intervals of polynomial-time decidable total orders. They also characterize the class of #P functions with easy existence version, as the class of functions that count the size of intervals of polynomial-time decidable partial orders with efficient adjacency checks.

In the second approach, by Kiayias *et al.* [2], they define the class TotP, consisting of functions that count the total number of paths of NP computations.

Pagourtzis and Zachos [3] show that the Karp-closure of TotP coincides with the set of self-reducible #PE functions, under a natural notion of self-reducibility, thus containing many natural #P functions with easy existence version such as the ones mentioned above.

In this work, we define interval size counting classes on orders with increasingly strong feasibility constraints. We provide inclusion and separation relations between TotP and these classes. Among others, we are able to give suitable feasibility constraints that characterize TotP and FP as interval size counting classes. Our results imply that many known #P-complete problems with easy decision are contained in the classes defined in [1]—but are unlikely to be complete for these classes under certain types of reductions. We also define a new class of interval size functions which lies strictly between FP and TotP under reasonable complexity-theoretic assumptions, and we show that it contains some hard counting problems.

[1] Hemaspaandra, L.A., Homan, C.M., Kosub, S., Wagner, K.W.: The complexity of computing the size of an interval. SIAM J. Comput. **36**(5) (2007) 1264–1300

[2] Kiayias, A., Pagourtzis, A., Sharma, K., Zachos, S.: The complexity of determining the order of solutions. In: Procedings of the First Southern Symposium on Computing. (Hattiesburg, Mississippi, 4-5 December 1998)

[3] Pagourtzis, A., Zachos, S.:The Complexity of Counting Functions with Easy Decision Version. In: Proceedings of MFCS 2006, Lecture Notes in Computer Science Vol. 4162, 741–752.

ON THE (SEMI)LATTICES INDUCED BY CERTAIN REDUCIBILITIES

Arno Pauly

A natural approach to computability on uncountable sets such as infinite words or real numbers involves approximation by elements of a countable set. Thus approximability, formalized as continuity, of functions is of great relevance to theoretical computer science. In particular, in computable analysis [2], continuity appears to be a straightforward generalization of computability.

In this setting, we study a continuous equivalent to bounded Turing reducibility, limited to a single oracle query. For two functions f, g, $f \leq_2 g$ holds, iff continuous functions F, G exist with $f(x) = F(x, g(G(x)))$ for all x in the domain of f. We will show that the partial ordered class of equivalence classes regarding $\leq_2$ is a complete join-semilattice, that is all suprema exist.

Another interesting reducibility corresponds to many-one reducibility, where $f \leq_0 g$ holds, iff there is a continuous function G with $f = g \circ G$. As will be demonstrated, the equivalence classes for $\leq_0$ even form a complete lattice, so all suprema and infima exist.

The ability to construct suprema and infima facilitates the study of different degrees of discontinuity and incomputability of functions defined on uncountable sets.

The results presented here are a special case of those given in [1], where proofs can also be found.

[1] Arno Pauly. On the (semi)lattices induced by continuous reducibilities. arXiv:0903.2177v1, March 2009.

[2] Klaus Weihrauch. *Computable Analysis*. Springer-Verlag, 2000.

GENERALIZED MATCHING

Alexandru Popa in joint work with Raphael Clifford, Aram Harrow and Benjamin Sach

Given a pattern p over an alphabet Σ_p and a text t over an alphabet Σ_t, we consider the problem of determining a mapping f from Σ_p to Σ_t^+ such that $t = f(p_1)f(p_2)\dots f(p_m)$. This class of problems, which was first introduced by Amir and Nor in 2004, is defined by different constraints on the mapping f. We give NP-Completeness results for a wide range of conditions. These include both when f is function or a bijection, when Σ_t is binary and when the range of f is limited to strings of constant length. We then introduce a related problem we term *pattern matching under string classes* which we show to be solvable efficiently. Finally, we discuss an optimization variant of generalized matching and give a polynomial time $\sqrt{\text{OPT}/k}$-approximation algorithm for fixed k.

FUNCTIONS DEFINABLE BY ARITHMETIC CIRCUITS

Ian Pratt-Hartmann in joint work with Ivo Düntsch

An *arithmetic circuit* is a labelled, directed graph specifying a cascade of arithmetic and logical operations to be performed on sets of non-negative integers (henceforth: *numbers*). Each node in this graph evaluates to a set of numbers, representing a stage of the computation performed by the circuit. Nodes without predecessors in the graph are called *input nodes*, and are labelled with any of the symbols $\{1\}$, $\{0\}$, $\emptyset$ or $\mathbb{N}$, denoting a set of numbers in the conventional way, or, alternatively, with a variable ranging over sets of numbers. Nodes with predecessors in the graph are called *arithmetic gates*, and are labelled with any of the symbols $+$, $\bullet$, $\bar{}$, $\cap$ or $\cup$, denoting an operation on sets. The symbols $\bar{}$, $\cap$, $\cup$ have the obvious Boolean interpretations (with $\bar{}$ denoting complementation in $\mathbb{N}$), while $+$ and $\bullet$ denote the result of lifting addition and multiplication to the algebra of sets.

In this talk, we consider the expressive power of arithmetic circuits. In particular, we ask: which functions (from tuples of sets of numbers to sets of numbers) are definable by arithmetic circuits? We obtain two negative results: the first shows, roughly, that a function is not circuit-definable if it has an infinite range and sub-linear growth; the second shows, roughly, that a function is not circuit-definable if it has a finite range and fails to converge on certain 'sparse' chains under inclusion. We observe that various functions of interest fall under these descriptions. In particular, arithmetic circuits can compute remainders (on division by a constant) but not quotients; they can determine whether a set is a singleton, but not whether it is finite; they can compute the minimum of a set, but not the maximum of a finite set, or its cardinality, or its sum.

This work is supported by the EPSRC, grant number EP/F069154/1.

FLEXIBLE BUSINESS PROCESSES USING STPOWLA

Stephan Reiff-Marganiec

Service Oriented Computing is a paradigm for developing software systems as the composition of a number of services. Services are loosely coupled entities, that can be dynamically published, discovered and invoked over a network. The engineering of such systems presents novel challenges, mostly due to the dynamicity and distributed nature of service-based applications. In this paper, we focus on the modelling of service orchestrations. We discuss the relationship between two languages developed under the Sensoria project: SRML as a high level modelling language for Service Oriented Architectures, and StPowla as a process-oriented orchestration approach that separates core business processes from system variability at the end-user's level, where the focus is towards achieving business goals. A fundamental challenge of software engineering is to correctly align business goals with IT strategy, and as such we present an encoding of StPowla to SRML. This provides a formal framework for StPowla and also a separated view of policies representing system variability that is not present in SRML.

AN INTRODUCTION TO THE COMPLEXITY OF CONSTRAINT SATISFACTION PROBLEMS

David Richerby

I will give a brief survey of the computational complexity of the constraint satisfaction problem (CSP). We are given a set of variables, to which we can assign values from some finite domain, along with a set of constraints, expressed using relations over the domain. Three versions of CSP will be considered.

- Decision CSP: can we assign values to the variables so that all the constraints are satisfied simultaneously?
- Counting CSP: how many distinct variable assignments satisfy all the constraints?
- Weighted CSP: generalizes constraints to functions expressing the probability or desirability of variable configurations.

ONLINE APPROXIMATE MATCHING WITH NON-LOCAL DISTANCES

Benjamin Sach in joint work with Raphaël Clifford

A black box method was recently given that solves the problem of online approximate matching for a class of problems whose distance functions can be classified as being local. A distance function is said to be local if for a pattern P of length m and any substring $T[i, i + m1]$ of a text T, the distance between P and $T[i, i + m1]$ is equal to $\Sigma_j \Delta(P[j], T[i + j1])$, where Δ is any distance function between individual characters. We extend this line of work by showing how to tackle online approximate matching when the distance function is non-local.

In our model we assume that we are given a pattern in advance and the text to which it is to be matched arrives one character at a time. The overall task is to report matches between the pattern and text as soon as they occur and to bound the worst case time *per input character*. It is an important feature of both our approach and the previous work that the running time of the resulting algorithms is not amortised.

In the talk, we will present an overview of this work and briefly discuss the methods used. Our solutions are applicable to a wide variety of matching problems including function and parameterised matching, swap matching, swap-mismatch, k-difference, k-difference with transpositions, overlap matching, edit distance/LCS, flipped bit, faulty bit and L_1 and L_2 rearrangement distances. The resulting algorithms bound the worst case running time to within a log factor of their comparable offline counterpart.

HOW TO AVOID WASTING PARALLEL PERFORMANCE

András Z. Salamon

Parallel processing aims to reduce the amount of time (makespan) that a computation requires, by using additional processors. When a parallel computation is expressed using environments such as Khronos OpenCL or Mathematica, one is generally forced to sequence activities one after the other, or to break parts of the computation into independent activities. This imposes a series-parallel structure on the activity network that underlies the computation. How does this affect the performance that is achievable?

I consider a simple activity network model of parallel computation, consisting of a partial order with node weights, representing activity durations. Edges mean that one activity must complete before another may start, and are called precedence constraints. When expressing a computation in a parallel programming environment, one implicitly chooses which precedence constraints are added to the activity network to make it series-parallel. Adding precedence constraints cannot decrease the makespan, but can one always series-parallelise without increasing the makespan too much?

I conjecture that a 4/3 increase is always achievable, but that achieving this bound is NP-complete. The 4/3 bound is tight. On the other hand, a linear algorithm achieves an increase that is bounded by the ratio between the largest and smallest activity durations. The activity network model suggests that it is possible to avoid wasting one-third of the potential performance gains of parallel processing by using more expressive language constructs, by carefully choosing what is an activity, and by introducing some redundancy.

AN ALGORITHMIC AND GRAPH THEORETIC VIEWPOINT OF SECURITY

Paul Sant in joint work with Tim French and Nik Bessis

As we move into a world in which we are increasingly dependent upon technology, and in which ubiquitous systems are on the increase issues related to Security and Trust are becoming increasingly important.

Many models of trust have been proposed, but there is still a need to merge methods for calculating trust values with ideas from, for example, ideas from the social sciences (e.g. semiotics).

In this talk I will discuss some initial ideas and results for models of trust that approach the problem from an algorithmic perspective. There will also be a discussion about the benefits of using the framework of semiotics to add value to models of trust.

The talk will conclude with a discussion of open issues in the area.

SAFE FUNCTIONAL REACTIVE PROGRAMMING THROUGH DEPENDENT TYPES

Neil Sculthorpe in joint work with Henrik Nilsson

Functional Reactive Programming (FRP) is an approach to reactive programming where systems are structured as networks of functions operating on signals. FRP is based on the synchronous data-flow paradigm and supports both continuous-time and discrete-time signals (hybrid systems). What sets FRP apart from most other languages for similar applications is its support for systems with dynamic structure and for higher-order reactive constructs.

Statically guaranteeing correctness properties of programs is an attractive proposition. This is true in particular for typical application domains for reactive programming such as embedded systems. To that end, many existing reactive languages have type systems or other static checks that guarantee domain-specific properties, such as feedback loops always being well-formed. However, they are limited in their capabilities to support dynamism and higher-order data-flow compared with FRP. Thus, the onus of ensuring such properties of FRP programs has so far been on the programmer as established static techniques do not suffice.

Here, we show how dependent types allow this concern to be addressed. By embedding an implementation of FRP in the the dependently-typed language Agda, we can use the type system of the host language to craft a domain-specific (dependent) type system for FRP. As the implementation passes the Agda type, coverage, and termination checks, we know our type system is safe.

APPROXIMATING NODE-WEIGHTED MULTICAST TREES IN WIRELESS AD-HOC NETWORKS

Ambreen Shahnaz in joint work with Thomas Erlebach

Multicast communication in a wireless ad-hoc network can be established using a tree that spans the multicast sender and receivers as well as other intermediate nodes. If the network is modelled as a graph, the multicast tree is a Steiner tree, the multicast sender and receivers correspond to terminals, and other nodes participating in the tree are Steiner nodes. As Steiner nodes are nodes that participate in the multicast tree by forwarding packets but do not benefit from the multicast, it is a natural objective to compute a tree that minimizes the total cost of the Steiner nodes. We therefore consider the problem of computing, for a given node-weighted graph and a set of terminals, a Steiner tree with Steiner nodes of minimum total weight. The problem is defined as follows: Given an undirected graph $G = (V, E)$ with nonnegative weights w_v for $v \in V$ and a subset of nodes $K \subseteq V$ called *terminals*, compute a *Steiner tree* for G and K, i.e., a subgraph T of G that is a tree and contains all the nodes in K. The objective is to minimize the total weight of the vertices of T. We can assume without loss of generality that the terminals have weight 0 (they are present in any solution and their weight increases the objective value of any solution by the same amount), so our goal is to minimize the total weight of the Steiner nodes of T. For graph classes that admit spanning trees of maximum degree at most d, we obtain a $0.775d$-approximation algorithm.

We show that this result implies a 3.875-approximation algorithm for unit disk graphs, an $O(1/\alpha^2)$-approximation algorithm for α-unit disk graphs, and an $O(\lambda)$-approximation algorithm for $(\lambda + 1)$-claw-free graphs. Unit disk graph is a simplified and idealistic model, whereas α-unit disk graph is a more general graph model for wireless ad-hoc networks. $(\lambda + 1)$-claw-free graphs include bounded independence graphs, which also better reflect realistic wireless ad-hoc networks.

FA-PRESENTABLE STRUCTURES

Richard M Thomas in joint work with Alan Cain, Graham Oliver and Nik Ruškuc

We are interested in the notion of computing in structures (where a structure consists of a set together with a collection of relations). The natural approach would be to take some general model of computation (such as a Turing machine). A structure would then be said to be "computable" if its domain can represented by a set which is accepted by a Turing machine and if there are decision-making Turing machines for each of its relations. However, there have been various ideas put forward to restrict the model of computation used; whilst the range of possible structures decreases, the computation can become more efficient and certain properties of the structures may become decidable.

One interesting approach was introduced by Khoussainov and Nerode who considered structures whose domain and relations can be checked by finite automata (as opposed to Turing machines); such a structure is said to be "FA-presentable". This was inspired, in part, by the theory of "automatic groups" introduced by Epstein et al; however, the definitions are somewhat different.

We will survey some of what is known about FA-presentable structures, contrasting it with the theory of automatic groups and posing some open questions. The talk is intended to be self-contained, in that no prior knowledge of these topics is assumed. We will be concentrating on some recent results on FA-presentable semigroups (joint work with Alan Cain, Graham Oliver and Nik Ruškuc).

PARTITIONING GRAPHS INTO CONNECTED PARTS

Pim van 't Hof in joint work with Daniël Paulusma and Gerhard J. Woeginger

The 2-Disjoint Connected Subgraphs problem asks if a given graph has two vertex-disjoint connected subgraphs containing prespecified sets of vertices. We show that this problem is already NP-complete if one of the sets has cardinality 2. The Longest Path Contractibility problem asks for the largest integer ℓ for which an input graph can be contracted to the path P_ℓ on ℓ vertices. We show that the computational complexity of the Longest Path Contractibility problem restricted to P_ℓ-free graphs jumps from being polynomially solvable to being NP-hard at $\ell = 6$, while this jump occurs at $\ell = 5$ for the 2-Disjoint Connected Subgraphs problem. We also present an exact algorithm that solves the 2-Disjoint Connected Subgraphs problem faster than $O^*(2^n)$ time for any P_ℓ-free graph. For $\ell = 6$, its running time is $O^*(1.5790^n)$. We modify this algorithm to solve the Longest Path Contractibility problem for P_6-free graphs in $O^*(1.5790^n)$ time.

PROPERTY SPECIFICATIONS FOR WORKFLOW MODELLING

Peter Y. H. Wong in joint work with Jeremy Gibbons

Formal developments in workflow languages allow developers to describe their work flow systems precisely, and permit the application of model checking to automatically verify models of their systems against formal specications. One of these workflow languages is the Business Process Modelling Notation (BPMN), for which we previously provided two formal semantic models in the language of Communicating Sequential Processes. Both models leverage the refinement orderings that underlie CSP's denotational semantics, allowing BPMN to be used for specification as well as modelling of workflow processes. However, the expressiveness of BPMN is strictly less than that of CSP, and as a result some behavioural properties, against which developers might be interested to verify their workflow processes, might not be easy or even possible at all to capture in BPMN.

In this talk we will consider a pattern-based approach to expressing behavioural properties. We will describe a property specification language *PL* for capturing a generalisation of Dwyer et al.'s Property Specication Patterns, and present a translation from *PL* into a bounded, positive fragment of linear temporal logic, which can then be automatically translated into CSP for simple renement checking. We demonstrate its application via a simple example.

This work is supported by a grant from Microsoft Research.

THE EXPRESSIVE POWER OF BINARY SUBMODULAR FUNCTIONS

Stanislav Živný in joint work with David Cohen and Peter Jeavons

It has previously been an open problem whether all Boolean submodular functions can be decomposed into a sum of binary submodular functions over a possibly larger set of variables. This problem has been considered within several different contexts in computer science, including computer vision, artificial intelligence, and pseudo-Boolean optimisation. Using a connection between the expressive power of valued constraints and certain algebraic properties of functions, we answer this question negatively. Our results have several corollaries. First, we characterise precisely which submodular functions of arity 4 can be expressed by binary submodular functions. Next, we identify a novel class of submodular functions of arbitrary arities which can be expressed by binary submodular functions, and therefore minimised efficiently using a so-called expressibility reduction to the Min-Cut problem. More importantly, our results imply limitations on this kind of reduction and establish for the first time that it cannot be used in general to minimise arbitrary submodular functions. Finally, we refute a conjecture of Promislow and Young on the structure of the extreme rays of the cone of Boolean submodular functions.