Report on BCTCS 2012

The 28th British Colloquium for Theoretical Computer Science

2-5 April 2012, University of Manchester

Ian Pratt-Hartmann

The British Colloquium for Theoretical Computer Science (BCTCS) is an annual forum in which researchers in Theoretical Computer Science can meet, present research findings, and discuss developments in the field. It also provides an environment for PhD students to gain experience in presenting their work in a wider context, and to benefit from contact with established researchers.

BCTCS 2012 was hosted by the University of Manchester, and held from 2nd to 5th April, 2012. The event attracted over 50 participants, and featured an interesting and wide-ranging programme of six invited talks (two from the same speaker) and 33 contributed talks, covering virtually all areas of the subject. This year, BCTCS was collocated with the 19th Workshop for Automated Reasoning (ARW), which attracted over 30 participants; plenary sessions were shared between the two events. Abstracts for all of the talks are provided below.

The conference began with an invited talk by Mike Edmunds, of Cardiff University, entitled "The Antikythera Mechanism and the early history of mechanical computing." Other invited talks were given by Reiner Hähnle, of the Technische Universität, Darmstadt, ("Formal verification of software product families"), Nicole Schweikardt of the Goethe-Universität, Frankfurt am Main ("On the expressive power of logics with invariant uses of arithmetic predicates") and Daniel Kroening, of Oxford University ("SAT over an Abstract Domain"). As in previous years, the London Mathematical Society sponsored a keynote talk in Discrete Mathematics: for this, Rod Downey, of the Victoria University of Wellington, gave two lectures on "Fundamentals of Parametrized Complexity." The financial support of the London Mathematical Society (LMS) is gratefully acknowledged.

BCTCS 2013 will be hosted by the University of Bath from 25th to 28th March, 2013. Researchers and PhD students wishing to contribute talks concerning any aspect of Theoretical Computer Science are cordially invited to do so. Further details are available from the BCTCS website at `www.bctcs.ac.uk`.

Bulletin of the EATCS no 107, pp. 145–165, June 2012

Invited Talks at BCTCS 2012

Mike Edmunds, University of Cardiff
The Antikythera mechanism and the early history of mechanical computing

Perhaps the most extraordinary surviving relic from the ancient Greek world is a device containing over thirty gear wheels dating from the late 2nd century B.C., and now known as the Antikythera Mechanism. This device is an order of magnitude more complicated than any surviving mechanism from the following millennium, and there is no known precursor. It is clear from its structure and inscriptions that its purpose was astronomical, including eclipse prediction. In this illustrated talk, I will outline the results—including an assessment of the accuracy of the device—from our international research team, which has been using the most modern imaging methods to probe the device and its inscriptions. Our results show the extraordinary sophistication of the Mechanism's design. There are fundamental implications for the development of Greek astronomy, philosophy and technology. The subsequent history of mechanical computation will be briefly sketched, emphasising both triumphs and lost opportunities.

Reiner Hähnle, Technische Universität Darmstadt
Formal verification of software product families

Formal verification techniques for software product families not only analyse individual programs, but act on the artifacts and components which are reused to obtain multiple software products. As the number of products is exponential in the number of artifacts, it is essential to perform verification in a modular fashion instead of verifying each product separately: the goal is to reuse not merely software artifacts, but also their verification proofs. In our setting, we realize code reuse by delta-oriented programming, an approach where a core program is gradually transformed by code "deltas" each of which corresponds to a product feature. The delta-oriented paradigm is then extended to contract-based formal specifications and to verification proofs. As a next step towards modular verification we transpose Liskov's behavioural subtyping principle to the delta world. Finally, based on the resulting theory, we perform a syntactic analysis of contract deltas that permits us to automatically factor out those parts of a verification proof that stays valid after applying a code delta.

Nicole Schweikardt, Goethe-Universität
On the expressive power of logics with invariant uses of arithmetic predicates

In this talk I consider first-order formulas (FO, for short) where, apart from the symbols in the given vocabulary, also predicates for linear order and arithmetic may be used. For example, order-invariant formulas are formulas for which the

following is true: if a structure satisfies the formula with one particular linear order of the structure's universe, then it satisfies the formula with any linear order of the structure's universe. Arithmetic-invariant formulas are defined analogously, where, apart from the linear order, other arithmetic predicates may be used in an invariant way. When restricting attention to finite structures, it is known that order-invariant FO is strictly more expressive than plain FO, and arithmetic-invariant FO can express exactly the properties that belong to the circuit complexity class AC^0. On the other hand, by Trakthenbrot's theorem we know that order-invariance (on the class of finite structures) is undecidable. In this talk I want to give an overview of the state-of-the art concerning the expressive power of order-invariant FO and arithmetic-invariant FO.

Daniel Kroening, Oxford University
SAT over an abstract domain

We present a generalisation of the DPLL(T) framework to abstract domains. As an instance, we present a sound and complete analysis for determining the range of floating-point variables in embedded control software. Existing approaches to bounds analysis either use convex abstract domains and are efficient but imprecise, or use floating-point decision procedures, and are precise but do not scale. We present a new analysis that elevates the architecture of a modern SAT solver to operate over floating-point intervals. In experiments, our analyser is consistently more precise than a state-of-the-art static analyser and significantly outperforms floating-point decision procedures.

Rod Downey, Victoria University of Wellington
Fundamentals of parametrized complexity

Parameterized complexity is a multivariant view of complexity seeking to utilize the order present in natural problems to establish practical tractability, or to provide tools that such a methodology won't work. Since the original work in the early 1990's, there has been a clearly defined set of techniques tunes to this idea.

In this pair of tutorial lectures I will discuss the basic methods used for the construction of parameterized algorithms, and some of the methods for showing parameterized intractability and optimality of the computational classes.

This method is something that a modern person doing algorithms should know.

The first lecture will be about the positive techniques, and the second on limitations. The material will be accessible to a beginning graduate student.

Contributed Talks at BCTCS 2012

Martin Adamčík, University of Manchester
Collective reasoning under uncertainty and inconsistency

In practice probabilistic evidence is incomplete and often contradictory. To build an artificial expert system such as one recognizing diseases from list of symptoms, one classical approach is to define an inference process which picks the "most rational" probabilistic belief function which an agent should have, based solely on the given evidence. For a single incomplete but consistent probabilistic knowledge base satisfying certain reasonable topological criteria, the Maximum Entropy (ME) inference process championed by Jaynes and others on the basis of combinatorial arguments, has several different justifications, and was uniquely characterized by an elegant list of axioms developed by Paris and Vencovská. ME enables a single rational agent to choose an optimal probabilistic belief function on the basis of his incomplete but consistent evidence. If however probabilistic evidence is derived from more than one agent, where the evidence from each individual agent is consistent, but the evidence from all agents together is inconsistent, then the question as to how to merge the evidence in such a manner as to be able to choose a single "most rational" probabilistic belief function on the basis of the merged evidence from all agents, has been much less studied from a general theoretical viewpoint.

In this talk we briefly describe a "social" inference process extending ME to the multi-agent context, called the Social Entropy Process (SEP), based on Kullback-Leibler information distance, and first formulated by Wilmers. SEP turns out to be a generalisation of the well-known logarithmic pooling operator for pooling the known probabilistic belief functions of several agents. We show that SEP satisfies a natural variant of the important principle of Irrelevant Information which is known to be satisfied by ME. We also indicate how the merging process described by SEP satisfies a suitable interpretation of the set of merging axioms for knowledge bases formulated by Konieczny and Pino Pérez in.

Chris Banks, University of Edinburgh
Towards a logic of biochemical processes

The Continuous Pi-calculus, cπ, is a continuous time and continuous space process calculus. The prime motivation cπ is for the modelling of the evolution of biochemical processes where the state of process is the real concentration of its constituent species and these concentrations are evolving continuously. Our aim is to provide a logic suitable for expressing properties of such processes, an algorithm for model checking, and tools to support the analysis of these processes.

Our proposed logic is based on Linear Temporal Logic with real constraints.

The deterministic nature of cπ processes means that a linear time logic is sufficient for expressing their temporal properties. However, in the context of biochemical processes, it is desirable to allow the expression of contextual properties. One might like to express how the system changes in different contexts, for example, with the introduction of new species into the system. The proposed logic contains an operator similar to the guarantee from spatial logic which allows the expression of such properties.

We also aim to provide a model checking algorithm to verify assertions in the logic. The problem is how to model check over a continuous state space. One approach to the model checking problem is to take the numerical solutions of ODEs which describe the process; this gives a discrete, deterministic approximation of the process dynamics within a finite time interval. Model checking can then be done using a relatively simple algorithm. Using this technique, software tools for analysis of cπ processes are being developed as part of the project.

Richard Barraclough

A unifying theory of control dependence and its application to arbitrary program structures

There are several similar definitions of control dependence in the literature. These are given in terms of control flow graphs which have had extra restrictions imposed (for example, end-reachability). We define two new generalisations of nontermination insensitive and nontermination sensitive control dependence called *weak* and *strong control-closure*. These are defined for all finite directed graphs, not just control flow graphs, and are hence allow control dependence to be applied to a wider class of program structures than before.

We define an underlying semantics for control dependence by defining two relations between graphs: weak and strong projections. We prove that the graph induced by a set of vertices is a weak/strong projection of the original if and only if the set is weakly/strongly control-closed. Thus, all previous forms of control dependence also satisfy our semantics. Weak and strong projections, thus, precisely capture the essence of control dependence both in our generalisations and all the previous – more restricted – forms. More fundamentally, these semantics can be thought of as correctness criteria for future definitions of control dependence.

Brandon Bennett, University of Leeds

An 'almost analytic' sequent calculus for first-order S5 with constant domains

We present a cut-free sequent calculus for the first-order modal logic S5 with constant domains. The system has the advantage of simplicity, in that all the rules are straightforward and intuitive. The rule set is analytic apart from one rule for eliminating a $\Box$ operator in the succedent of a sequent. Although this rule is not strictly analytic, it does not introduce new non-logical symbols, and

hence provides for an 'almost analytic' proof mechanism. The system is close in form to Gentzen's original sequent calculus for first-order logic. It does not employ any generalisation or augmentation of the basic form of a sequent and only involves very simple side conditions restricting the applicability of two of the rules. Moreover these conditions can be checked locally by looking only at the syntactic form of the immediate conclusion of the rule application.

Adequacy of the system is demonstrated by an inductive cut-elimination proof, which shows equivalence to a well-established Hilbert system formulation of first-order S5. The current proof has the shortcoming that it requires that the antecedents and succedents of a sequent be multi-sets rather than ordinary sets, which seems to be an unnecessary complication. Further work is ongoing to determine whether the possibility of having duplicated formulae in sequents is essential to the completeness of the system or whether such formulae are redundant.

Mihai Burcea, University of Liverpool

Online multi-dimensional dynamic bin packing of unit fraction and power fraction items

We study 2D and 3D dynamic bin packing, in which items arrive and depart at arbitrary times. The 1D problem was first studied by Coffman, Garey, and Johnson motivated by the dynamic storage problem. Bar-Noy et al. have studied packing of unit fraction items (i.e., items with lengths $1/w$ for some integer $w \geq 1$), motivated by the window scheduling problem.

We extend the study of 2D and 3D dynamic bin packing to unit fraction and power fraction items (i.e., items with lengths $1/2^k$ for some integer $k \geq 0$). The objective is to pack the items into unit-sized bins such that the maximum number of bins ever used over all time is minimized. We give a scheme that divides the items into classes and show that applying the first-fit algorithm to each class is 6.7850- and 21.6108-competitive for 2D and 3D, respectively, for unit fraction items. Similarly, we provide a scheme dividing power fraction items into classes for which the first-fit algorithm is 6.2455- and 20.0783-competitive for 2D and 3D, respectively. These are in contrast to the 7.788 and 22.788 competitive ratios for 2D and 3D general sized items.

This is joint work with Prudence W.H. Wong and Fencol C.C. Yung.

Evelyn-Denham Coates, Logic Code Generator Ltd, London, UK

Optimum sort algorithms with o(N) moves

We show implementation of an unstable algorithm that uses $o(N)$ additional memory to do no more than $N\lceil\log_2(N)\rceil - \lceil\log_2(N)\rceil$ comparisons, and no more than $3N$ data moves to sort an array of N values. We show modification to the algorithm so that it does unstable in-place sort with about the same number of comparisons and $O(N)$ data moves. We use our main algorithm to implement a stable merge-sort

with $o(N)$ pointers and $N + C - S$ data moves, where S = number of single cycle permutations in the set to be sorted and C = number of permutation cycles. The operation and operational complexity is as with natural merge-sort except for the additional amount of memory. We show an implementation of the main algorithm that uses $\log_2(N)$ parallel steps with $\frac{1}{2}N(N+1)$ interconnected processors to sort N input values. We present tabulated data from experimental test results on our main algorithm.

The algorithm does no more than $O(N(\log_2(N))^3)$ bit level operations. Without mathematical fanfare or much theoretical exposition, we contend that our algorithm demonstrate an instance solution to the comparing sort problem where a decoupling of comparisons from data moves improves the operational complexity on the number of comparisons and the number of data moves. We pose a theoretical challenge for additional research and development with this approach. We consider our result to be the best so far and possible the final solution to the sort/merge problem from what we have seen in the literature.

Laurence Day, University of Nottingham
The Silence of the Lambdas

At last year's BCTCS, I presented the preliminary results of implementing a modular compiler for a language supporting arithmetic and exceptions which has been constructed as the least fixpoint of functors, where functions over said language are defined as catamorphisms. In this talk, I will recap the necessary ideas before going on to discuss the implementation of the de-Bruijn indexed lambda calculus in this system and the need to switch from catamorphisms to explicit recursion when dealing with term substitution. I will conclude by discussing the potential impact that such a shift may have on notions such as modular proofs.

Michael Gabbay, King's College London
A very simple, explicit construction of some models of beta-equality and beta-eta-equality

We discuss a (relatively) new method of providing models of λ-reduction by which we are able to interpret λ-terms compositionally on "possible world" structures with a ternary accessibility relation. The simplicity of the structures is striking, moreover, they provide us with a surprising richness of interpretations of function abstraction and application.

We show how the models can differentiate between 'extensional' λ-reduction, which supports β-contraction and η-expansion, and 'intensional' reduction which supports only β-contraction. We state semantic characterisation (i.e. completeness) theorems for both. We then show how to extend the method to provide a sound and complete class of models for reduction relations that additionally support β-expansion and η-contraction (i.e. β-equality and η-equality). In this respect

the models we present differ from the familiar models of the λ-calculus as they can distinguish, semantically, between intensional and extensional λ-equality.

For the main result of the paper, we outline an explicit construction of a model of untyped λ-calculus. Again, the simplicity of the construction is striking. Furthermore the construction is sufficiently general that it can be modified to construct models either of $\beta\eta$-equality or simply β-equality.

Finally, we draw some speculative connections between the models constructed and neural nets, abstractly construed. This opens up the possibility that we can view a neural network as computing a λ-term in some sense.

Murdoch Gabbay, Heriot-Watt University
Game semantics using nominal techniques

Game semantics gives denotation to logic and computation using as metaphor a dialogue between *Proponent* and *Opponent*. This can be modelled as a labelled acyclic graph called a *pointer sequence*: nodes are labelled with Proponent / Opponent moves; edges represent the move's *justification*. We propose a model of pointer sequences based on nominal sets, using *atoms* to model edges. Atoms are just a countably infinite set of distinct symbols $a, b, c, \ldots$. Questions and answers are q and a. Pointers are rendered as a pair of atoms. The tip of an arrow is represented as *coabstraction* $[a]$ or $[b]$. Coabstractions bind 'into the future', and are a new idea to nominal techniques. The tail of an arrow is an atom occurence like a or b. *Dangling pointers* are just *free names* (in the sequence above c is free).

Nominal sequences have the following good properties: (1) Closure under subsequences. A subgraph of a pointer sequence is not a pointer sequence, because it might have 'dangling pointers'. (2) Closure under concatenation. Names link up and there are no reindexing isomorphisms. It is less obvious how pointer sequences concatenate. (3) Nominal sequences are an inductive datatype and can be manipulated with standard tools. There is also a specific *nominal* advantage: it enables efficient management of renaming pointers. This is why we use *names* and not e.g. numbers, which are permutatively asymmetric. Taking names and permutations as *primitive* gives good meta-theoretic properties since 'obvious' symmetry properties up to 'reindexing' become obvious. This style of name management and has proven effective in other applications. We shall see that it is also effective here, and we speculate that mechanisation of game semantics using our nominal model will be significantly easier than with pointer sequences.

This is joint work with Dan Ghica

Thomas Gorry, University of Liverpool
Communication-less agent location discovery

We study a randomised distributed communication-less coordination mechanism for uniform anonymous agents located on a circle. The agents perform their ac-

tions in synchronised rounds. At the beginning of each round an agent chooses the direction of its movement from *clockwise* and *anticlockwise*, as well as its speed $0 \leq v \leq 1$ during this round. We assume that the agents are not allowed to overpass, i.e., when an agent collides with another it instantly starts moving with the same speed in the opposite direction. The agents cannot leave marks on the ring, they have zero vision and they cannot exchange messages. However, on the conclusion of each round each agent has access to a detailed trajectory of its movement during this round. This information can be processed and stored by the agent for further analysis.

We assume that n mobile agents are initially located on a circle with circumference one at arbitrary but distinct positions unknown to other agents. The main *location discovery task* to be performed by each agent is to determine the initial position of every other agent and eventually to stop at its initial position, or proceed to another task, in a fully synchronised manner. Our main result is a fully distributed randomised (Las Vegas type) algorithm, solving the *location discovery problem w.h.p* in $O(n \log^2 n)$ rounds. We also show how this mechanism can be adopted to distribute the agents evenly, at equidistant positions, and how to coordinate their joint effort in patrolling the circle. Note that our result also holds if initially the agents do not know the value of n and they have no coherent sense of direction.

Tom Grant, University of Leicester
Maximising lifetime for fault-tolerant target coverage in sensor networks

We present the problem of maximising the lifetime of a sensor network for fault-tolerant target coverage in a setting with composite events. Here, a composite event is the simultaneous occurence of a combination of atomic events, such as the detection of smoke and high temperature. We are given sensor nodes that have an initial battery level and can monitor certain event types, and a set of points at which composite events need to be detected. The point and sensor nodes are located in the Euclidean plane, and all nodes have uniform sensing radius. The goal is to compute a longest activity schedule with the property that at any point in time, each event point is monitored by at least two active sensor nodes.

We present a $(6 + \varepsilon)$-approximation algorithm by devising an approximation algorithm with the same ratio for the dual problem of minimising the weight of a fault-tolerant sensor cover. This generalises previous approximation algorithms for geometric set cover with weighted unit disks and is obtained by enumerating properties of the optimal solution that guide a dynamic programming approach.

Paolo Guagliardo, Free University of Bozen-Bolzano
On the relationship between view updates and logical definability

Given a set of views defined over a database, the *view update problem* consists

in finding suitable ways of propagating an update performed on the views to the underlying database in a consistent and unique way. In this talk, we highlight the strong connection between the view update problem and the notion of definability in logic, and we revisit the abstract functional framework by Bancilhon and Spyratos in a setting where views and updates are exactly given by functions that are expressible in first-order logic. We give a characterisation of views and their inverses based on the notion of definability, and we introduce a general method for checking whether a view update can be uniquely translated as an update of the underlying database.

Christopher Hampson, King's College London
Modal Products with the difference operator

The modal logic **Diff** of the difference operator is known to be Kripke complete with respect to the class of symmetric, pseudo-transitive frames. These frames closely resemble **S5**-relations (i.e. equivalence relations) and it is little surprise that the validity problems for **Diff** and **S5** have the same co-NP complexity, and both logics enjoy the finite model property.

Here we turn our attention to two-dimensional product logics $L_1 \times L_2$, by which we mean the multimodal logic of all product frames where the first component is a frame for L_1 and the second a frame for L_2. It is well-known that product logics of the form $L \times \mathbf{S5}$ are usually decidable whenever L is a decidable (multi)modal logic. We even have that $\mathbf{S5} \times \mathbf{S5}$ enjoys the exponential finite model property. However, it is little understood how product logics of the form $L \times \mathbf{Diff}$ behave.

Here we present some cases where the transition from $L \times \mathbf{S5}$ to $L \times \mathbf{Diff}$ not only increases the complexity of the validity problem, but in fact introduces undecidability. The logics we consider are (i) $\mathbf{K}_u \times \mathbf{Diff}$, which is shown to be undecidable in contrast to $\mathbf{K}_u \times \mathbf{S5}$, which lies in co-N2ExpTime, (ii) $\mathbf{PTL}_{\bigcirc\Box} \times \mathbf{Diff}$, which is shown to be non-r.e. in contrast to the ExpSpace-completeness of $\mathbf{PTL}_{\bigcirc\Box} \times \mathbf{S5}$, and (iii) $\mathbf{Diff} \times \mathbf{Diff}$, which is shown to lack the finite model property, in contrast to $\mathbf{S5} \times \mathbf{S5}$ as mentioned above. Our undecidability and non-r.e.ness results are obtained by reductions of halting-type problems for Minsky machines on two registers, which are known to be Turing-complete.

This is joint work with Agi Kurucz.

Tie Hou, Swansea University
Modeling a language of realizers using domain-theoretic semantics

How to synthesize efficient programs from proofs obeying their formal specifications has been a long sought after goal. One method of program extraction is to employ a realizability interpretation. Kleene first introduced the concept of realizability with the idea of defining a relation between natural numbers and logical sentences. Later many other notions on realizability were introduced, e.g.

the "modified realizability" of Kreisel and the "function realizability" of Kleene-Vesley. The possibility of effectively obtaining a program and its verification proof is based on a sound realizability interpretation.

We study the domain-theoretic semantics of a Church-style typed λ-calculus with constructors, pattern matching and recursion, and show that it is closely related to the semantics of its untyped counterpart. The motivation for this study comes from program extraction from proofs via realizability where one has the choice of extracting typed or untyped terms from proofs. Our result shows that if the extracted type is regular, the choice does not matter.

The proof uses hybrid logical relations. Logical relations have been used successfully to prove properties of typed systems. Famous examples are the strong normalization proofs by Tait and Girard using logical relations called computability predicates or reducibility candidates. The crucial feature of a logical relation is that it is a family of relations indexed by types and defined by induction on types such that all type constructors are interpreted by their logical interpretations.

The reason for studying this domain-theoretic semantics is that it allows for very simple and elegant proofs of computational adequacy, and hence for the correctness of program extraction. Since domain theory combines the computational features of functions with the mathematical definition of function as a mapping from one domain to another, from the perspective point of view, a functional language is basically a shorthand notation for domain-theoretic concepts.

Phillip James, Swansea University
Domain-specific languages and automatic verification

In this talk, we explore the support of automatic verification via careful design of a domain specific language (DSL). For verification, such a specialized language has two effects: (i) Only specific proof goals can be expressed in the language. (ii) The language semantics includes axioms expressing domain knowledge. We illustrate these ideas within the Railway Domain. The semantics of our DSL is a specification in the algebraic specification language CASL. To provide proof support, we use various automated theorem provers which are accessible via the Heterogeneous tool-set (Hets). Finally, we provide concrete verification results illustrating that careful design of a DSL and systematic use of domain knowledge is useful for supporting automatic verification. The result is a step towards a platform for creating domain specific languages with effective automatic verification tools for domain engineers.

Sam Jones, University of Leicester
Groups, formal language theory and decidability

One natural way to describe a group G is by means of a “presentation" consisting of a set X of generators for G and a set R of relations between words over X. If X

is finite then G is finitely generated, and if R is finite then G is finitely presented.

The word problem for a finitely generated group G is an algorithmic question which asks: given two words α and β over some (finite) generating set for G are the elements of G represented by the words α and β the same? An equivalent formulation of this question is: given two words α and β over some generating set for G is the element of G represented by $\alpha\beta^{-1}$ the identity element of G? In this way we can think of the word problem for G as the problem of determining membership of the set of all words which represent the identity element of G.

In this talk I give a brief overview of some of the interactions between group theory and formal language theory, in particular, I will focus on the word problem for groups and the study of the word problem as a formal language. I will explain how groups can be classified in terms of the type of automata which accept their word problem. I will then talk about some decidability questions and results in formal language theory on which I have been working which were motivated by the study of the word problem for groups as a formal language.

Stanislaw Kikot, Birkbeck College, London

The length of query rewriting for OWL 2 QL

Let Σ be a signature consisting of a finite number of constants, unary and binary predicates (A_i and R_i, respectively) and equality. We consider a class of first-order *theories* $\mathcal{T}$ with formulas of the form

- $\forall x(C_1(x) \to C_2(x))$, where $C_1(x)$ and $C_2(x)$ are $A_i(x)$ or $\exists y R_i(x, y)$,
- $\forall x \forall y(R_i(x, y) \to R_j(x, y))$ and $\forall x \forall y(R_i(x, y) \to R_j(y, x))$,

and *conjunctive queries* $\mathbf{q}(\vec{x}) = \exists \vec{y} \varphi(x, y)$, where $\varphi(x, y)$ is a conjunction of atoms $A_i(t_1)$ and $R_i(t_1, t_2)$ and t_1 and t_2 are either constants or variables from $\vec{x}, \vec{y}$. The *query rewriting problem* is, given a theory $\mathcal{T}$, a query $\mathbf{q}(\vec{x})$ and a first-order language $\mathcal{L}$, construct a first-order formula $\mathbf{q}^{\mathcal{T}}(\vec{x})$ in the language $\mathcal{L}$ (we call it "the $\mathcal{L}$-rewriting of a query $\mathbf{q}(\vec{x})$ with respect to a theory $\mathcal{T}$") such that for all sets $\mathcal{A}$ of *data* (ground atoms of the form $A(a), R(a, b)$), we have $\mathcal{A} \cup \mathcal{T} \models \mathbf{q}(\vec{a})$ if and only if $\mathcal{A} \models \mathbf{q}^{\mathcal{T}}(\vec{a})$. For example, the rewriting of the query $\mathbf{q}(x) = \exists y R(x, y)$ with respect to the theory $\{\forall x(A(x) \to \exists y R(x, y))\}$ is $\mathbf{q}^{\mathcal{T}}(x) = A(x) \vee \exists y R(x, y)$.

We prove that for "natural" languages $\mathcal{L}$ the minimal size of $\mathbf{q}^{\mathcal{T}}(\vec{x})$ may be exponential in the combined size of $\mathbf{q}(\vec{x})$ and $\mathcal{T}$. On the other hand, we supply an algorithm which gives short rewritings for all reasonable practical cases.

We believe that our research is relevant to the next generation search engine design, where individuals range, say, over web pages, people and products, and examples of unary predicates are "contains the word *dextrose*" and "contains some information relevant to biology".

Andrew Lawrence, Swansea University
Extracting a DPLL Algorithm

In order for verification tools to be used in an industrial context they have to be trusted to a high degree and in some cases need to be certified. We have come up with a new application of program extraction to develop correct certifiable decision procedures. SAT-solvers are one such decision procedure which are common in verification tools. The majority of SAT-solvers used in an industrial context are based on the DPLL proof system. We have performed a correctness proof of the DPLL proof system in the Minlog theorem prover. Using the program extraction facilities of Minlog we have been able to obtain a formally verified SAT-solving algorithm. When run on a CNF formula this algorithm produces a model satisfying the formula or a DPLL derivation showing its unsatisfiability. Computational redundancy was then removed from the algorithm by labelling certain universal quantifiers in the proof as non-computational. The performance of the resulting program was tested with a number of pigeonhole formulae.

David Love, Sheffield Hallam University
Why Don't Cantor's Sets Compute?

In this talk we explore the rejection of the *structural equivalence* used in Hilbet's formal theory (and by extension those of computation). For Hilbert's formal theories, we assume that the structure *of* the theory is congruent with the structures described *by* the theory. Such a congruence is necessary for the self-referential properties of Hilbert's formal theories (which are in turn responsible for the power and scope of those theories).

Even rejecting the assumption of structural equivalence we show that we can build coherent mathematical theories: even if these are not actually formal theories. In this talk we do so using the example of two 'halting like' machines, creating two non-formal analogues of the more well known formal halting machines. By doing so we show that we can explore the space of mathematical theories beyond Hilbert formal theories – without rejecting the vast body of work that has been undertaken on those theories since Hilbert's pronouncement of 1904.

Yavor Nenov, University of Oxford
Computability of topological logics over Euclidean spaces

In the last decades, formalisms for representing and reasoning with spatial knowledge have been of a significant interest to the AI community. Such formalisms are usually referred to as *spatial logics* and consist of a logical language whose variables are interpreted as subsets of a topological space, called *regions*, and whose non-logical symbols have a fixed geometric interpretation. Spatial logics whose non-logical symbols represent topological relations and operations are

called *topological logics*.

We consider quantifier-free languages that feature symbols for Boolean operations and relations (e.g. *union*, *complementation*, etc.) and a predicate symbol for one of two notions of *connectedness* – the property of being topologically connected or the property of having a connected interior. We take the variables of each language to range over different collections of regions in a Euclidean space of dimension higher than one ($\mathbb{R}^n$, $n \geq 2$). We investigate the computability and the computational complexity of the resulting topological logics and show that, despite being based on a very simple logical syntax, they generally exhibit a very high computational complexity, and with few exceptions are all undecidable. The considered logics stand in stark contrast to other studied quantifier-free topological logics, which are all decidable and of relatively low computation complexity.

Jude-Thaddeus Ojiaku, University of Liverpool

Online makespan scheduling of linear deteriorating jobs on parallel machines

Traditional scheduling assumes that the processing time of a job is fixed. Yet there are numerous situations that the processing time increases (deteriorates) as the start time increases. Examples include scheduling cleaning or maintenance, fire fighting, steel production and financial management. Scheduling of deteriorating jobs was first introduced on a single machine by Browne and Yechiali, and Gupta and Gupta independently. In particular, lots of work has been devoted to jobs with linear deterioration. The processing time p_j of job J_j is a linear function of its start time s_j, precisely, $p_j = a_j + b_j s_j$, where a_j is the normal or basic processing time and b_j is the deteriorating rate. The objective is to minimize the makespan of the schedule.

The problem has been mainly studied in the context of offline setting, with optimal offline solutions on single machine and FPTAS on parallel machines. We first consider simple linear deterioration, i.e., $p_j = b_j s_j$. It has been shown that on m parallel machines, when jobs are given one by one and the algorithm has to schedule a job before knowing the next one (online-list model), LS (List Scheduling) is $(1 + b_{\max})^{1-\frac{1}{m}}$-competitive. We extend the study to the online-time model where each job is also associated with a release time. We show that for two machines, no deterministic online algorithm is better than $(1 + b_{\max})$-competitive, implying that the problem is more difficult in the online-time model than in the online-list model. We also show that LS is $(1 + b_{\max})^{2(1-\frac{1}{m})}$-competitive, meaning that it is optimal when $m = 2$. We further consider the case when one of the two machines is unavailable for a fixed time period. For the online-list model, we give an optimal semi-online-list algorithm when $b_{\max}$ is known in advance.

We also study another linear deterioration function, namely, $p_j = a_j + b\, s_j$. In the online-list model, on m machines, we show that RR (Round Robin) is α-competitive and LS is α-competitive in a special case, where α is the ratio of

maximum and minimum normal processing times.

Arnoud Pastink, University of Liverpool
Approximate Nash Equilibria in an uncoupled setup with limited communication

Since it was shown that finding a Nash equilibrium is PPAD-complete, attention has been given to other equilibrium concepts that give approximately a Nash equilibrium in polynomial time. The two most common concepts are (additive) ϵ-approximate Nash equilibria and well-supported Nash equilibria (ϵ-SuppNE); the first requires a strategy of every player such that deviating can give a gain of at most ϵ, and the latter requires that every pure strategy that is played with positive probability is an ϵ-approximate best-response.

Most algorithms for computing ϵ-approximate Nash equilibria assume that all payoffs are known by everybody. There are very few results about approximations in an uncoupled setup where players only know their own payoff matrix. In this uncoupled setup we allow the players to communicate a limited amount of communication. The best ϵ-approximate Nash equilibrium procedure with limited communication in an uncoupled setup at the moment is a simple algortihm which achieves a 0.5–approximate Nash equilibrium by looking at strategies with a support size of 2 and a communication complexity of $O(\log n)$. For ϵ-SuppNE, no non-trivial approximations are known with limited communication.

I will present algorithms that achieve a 0.438–approximate Nash equilibrium and a 0.732-SuppNE, both with a polylogarithmic communication complexity. These results are achieved by cleverly using the properties of the zero-sum game of the player's own payoff matrix.

Robert Piro, University of Oxford
Model-theoretic characterisation of TBoxes and the TBox rewritability Problem

With Knowledge Representation in AI, knowledge is represented in logic, thus giving the representation a clearly defined semantics and making it machine processible. Description Logics (DL), which are essentially decidable fragments of First Order Logics, have been introduced to facilitate this. The decidability allows for automated reasoning and the developing of tool support. The properties and statements a specific DL allows to express must be carefully chosen, as the reasoning complexity rises with the expressiveness of a logic. To cater for the different needs and requirements, a whole zoo of DLs have been introduced and classified w.r.t. their complexity.

An interesting problem therefore is to determine their expressivity. Traditionally, the expressivity of a logic is determined by a characterisation theorem, in which model theoretic properties are determined, such that every first order for-

mula with these properties is expressible as formula of the logic in hand and vice versa. The famous theorem of van Benthem, which characterises the DL $\mathcal{ALC}$ on concept level, is of this kind as well as the work of Kurtonina and de Rijke, who gave characterisations for a whole zoo of description logics.

TBoxes however, which contain sentences describing concept hierarchies and play an important role in ontologies, have not been investigated. Thus, the talk will concentrate on characterisation theorems of TBoxes of the $\mathcal{ALC}$-family as well as $\mathcal{EL}$-TBoxes. Additionally we shall present the rewritability problem for TBoxes, which asks whether a TBox of a certain DL is expressible as TBox of another DL with lower complexity.

Giles Reger, University of Manchester
Quantified event automata: towards expressive and efficient runtime monitors

Runtime verification techniques have recently focused on parametric specifications where events take data values as parameters. These techniques exist on a spectrum inhabited by both efficient and expressive techniques. These characteristics are usually shown to be conflicting - in state-of-the-art solutions, efficiency is obtained at the cost of loss of expressiveness and vice-versa. To seek a solution to this conflict we explore a new point on the spectrum by defining an alternative runtime verification approach. We introduce a new formalism for concisely capturing expressive specifications with parameters. Our technique is more expressive than the currently most efficient techniques while at the same time allowing for optimizations.

In this talk we present event automata and show how they can be extended with quantification to achieve an expressive formalism for monitoring the behaviour of programs at runtime. Using a range of examples, we will demonstrate how these quantified event automata can be interpreted and used within the context of runtime monitoring.

Yanti Rusmawati, University of Manchester
Dynamic networks as concurrent systems and supervised evolution

Highly dynamic and complex computing systems often need to adapt to changing external and internal environments. One approach to this is to build in evolvability as a feature of such systems. Dynamic networks provide examples of such systems, in which a collection of nodes are linked through edges with the number of nodes and edges varying over time. Applications include internet, mobile networks, and unreliable networks. The dynamic behaviours impact on message-passing mechanisms and computations attempting to reach 'consensus' or compute global properties. In order to undertake formal reasoning about such systems, abstract models are essential.

We consider abstract descriptions of dynamic networks by developing a formal

framework. We view dynamic networks as concurrent systems, in which there are at least two kinds of processes: a disrupter (which disrupts the connectivity of dynamic networks) and an organizer (which attempts to run the normal execution). Various notions of fairness enable us to reason about message-passing and routing. We also consider how dynamic networks may be modelled via supervised evolution, where components consist of computational systems which are 'monitored' by a supervisory system which may evolve the computation if necessary.

Hugh Steele, University of Manchester
Double glueing and MLL full completeness

Linear Logic (LL) is a deductive system that has garnered considerable attention over the past two decades. Its correspondence to a polymorphic lambda calculus has made it a topic particularly of interest to theoretical computer scientists. The logic's original description takes the form of a sequent calculus, making it ungainly at times; but there has been success expressing the proof theory of LL and of its smaller logical fragments in other ways. Derivations can be described graphically (with propositional atoms and connectives being represented by vertices) or as arrows in an appropriate category.

Naturally it is important for a category describing any logic to be as accurate a model as possible in order for it to be considered useful. The formalisation of this concept is known as 'full completeness': a categorical model of a logic is fully complete if all of its arrows correspond directly to a derivation. In this talk we demonstrate how it is possible to create fully complete models of MLL, the strongly normalising multiplicative fragment of LL, from certain degenerative models. The approach taken makes use of a 'double glueing' construction placed on top of tensor-generated compact closed categories with biproducts; and the new arguments which we employ to show the resulting categories have the properties desired are based around considering the combinatorics behind this construction using standard linear algebra.

Alistair Stewart, University of Edinburgh
Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars

We show that one can approximate the least fixed point solution for a multivariate system of monotone probabilistic polynomial equations in time polynomial in both the encoding size of the system of equations and in $\log(1/\epsilon)$, where $\epsilon > 0$ is the desired additive error bound of the solution.

We use this result to resolve several open problems regarding the computational complexity of computing key quantities associated with some classic and heavily studied stochastic processes, including multi-type branching processes and stochastic context-free grammars.

Martin Sticht, University of Bamberg
A Game-Theoretic Decision Procedure for the constructive Description Logic cALC

In the last years, several languages of Description Logic have been introduced to model knowledge and perform inference on it. There have been several propositions for different application scenarios. The constructive Description Logic *cALC* deals with uncertain or dynamic knowledge.

We make use of a game-theoretic dialogue-based proof technique that has its roots in philosophy and introduce rules so that we can perform reasoning in *cALC* and the modal-logical counterpart CK. The game-theoretic presentation can be considered as an alternative technique to tableau-based proofs, emphasising interaction semantics. As we will see, showing validity is more complex but in return we have a philosophical approach that might make it possible to find out more about related constructive theories and that provides a rich playground of possibilities to extend or alter the underlying semantics.

Dirk Sudholt, University of Sheffield
The analysis of evolutionary algorithms: why evolution is faster with crossover

Evolutionary algorithms use search operators like mutation, crossover and selection to 'evolve' good solutions for optimisation problems. In the past decades there has been a long and controversial debate about when and why the crossover operator is useful. The 'building-block hypothesis' assumes that crossover is particularly helpful if it can recombine good 'building blocks', i. e. short parts of the genome that lead to high fitness. However, attempts at proving this rigorously have been inconclusive; there have been no rigorous and intuitive explanation for the usefulness of crossover. In this talk we provide such an explanation. For functions where 'building blocks' need to be assembled, we prove that a simple evolutionary algorithm with crossover is twice as fast as the fastest evolutionary algorithm using only mutation. The reason is that crossover effectively turns fitness-neutral mutations into improvements by combining the right building blocks at a later stage. This leads to surprising conclusions about the optimal mutation rate.

Christopher D. Thompson-Walsh, University of Cambridge
Extending a Rule-Based Biological Modelling Language Semantics with Containment

Rule-based modelling of biochemical systems has emerged as an important approach in the development of computational techniques for the analysis of these systems. Many biological systems of chemical reactions exhibit a combinatorial explosion in the number of possible species and reactions. Rule-based techniques

rely on using rules which specify *patterns*, rather than explicit species, to succinctly describe these reactions.

One such rule-based modelling language is Kappa, a calculus which defines how a graph, representing a system of linked agents, can be modified by rules that specify which changes may occur at places that match specific local patterns. It has a clean graph-rewriting based semantics; and though the calculus has a wide degree of applicability, it has emerged as a natural description of well-mixed, protein-protein interaction systems and pathways in molecular biology. However, tdoes not presently model the partition of space by membranes, resulting in multiple well-mixed and interacting compartments.

In this talk, we describe work expanding on this graph-based semantics to add containment structure. This containment structure allows us to begin to model the various ways in which biological mixtures are partitioned and enclosed by membranes, which have important effects in real biological systems.

This is joint work with Jonathan Hayman and Glynn Winskel.

Patrick Totzke, University of Edinburgh
Weak bisimulation approximants for BPP processes

In automated verification we want to algorithmically check if a system satisfies a given property. The two main approaches are model checking and equivalence checking. In model checking, the properties are given as formulae of a temporal logic and one checks if a system satisfies these formulae. In equivalence checking, one checks if two given systems are in some semantical sense equal and thereby verify if an implementation is equivalent to a specification that encodes the desired properties. The question arises for which kinds of systems and equivalences this is decidable and if so, what are the complexity bounds.

We consider *weak bisimulation*, an equivalence that has a very intuitive characterization in terms of two-person games and look at systems called *Basic Parallel Processes*, which were introduced as derivations of commutative context-free grammars and are equi-expressible with communication-free Petri nets. The decidability of checking *weak bisimilarity* for *BPP* is a long standing open problem.

In this talk we explore the well known approach of weak bisimulation *approximants* and see how far this takes us for different notions of approximation. We successfully apply the approximation method to restricted classes of BPP processes and establish a few rather surprising lower bounds for the convergence levels of approximants considered before. Lastly, we define new subclass which demands a lot of additional structure. Surprisingly, all "hard" systems that we use to show lower bounds are contained in this very restricted subclass.

Chiara Del Vescovo, University of Manchester

The modular structure of an ontology: atomic decomposition

Ontologies are special logical theories: they are finite sets of axioms in a language that belongs to the family of Description Logics, which are decidable fragments of First Order Logic. They aim to describe knowledge about a domain of interest; and in general they are complex systems, unstructured and large. Decomposing ontologies into *modules* is widely accepted as a fruitful mechanism to ease processing, modifying, analyzing, and reusing parts of an ontology. However, modularisation is a difficult task to achieve for ontologies, because we want to preserve logical properties.

There exist several notions of logically coherent modules for logical theories. Each kind of module determines in the ontology a different *modular structure*, i.e. a set of logically coherent chunks of the ontology and interesting relations between these chunks. However, these suffer from being based on a loose conceptualization of logical connection, and in some notable examples the ontology cannot be decomposed into smaller bits, even if it seems to be well structured.

An important family of modules of ontologies is based on the notion of deductive Conservative Extensions (d-CEs): such modules encapsulate all the ontology's knowledge about a set of terms Σ, called *signature*. We focus on *locality-based modules*, that are computable in polynomial time, whose the most important notions are $\bot$, $\top$, and $\top\bot^*$. Locality-based modules are currently used in many scenarios, e.g. the reuse of part of an ontology.

In our talk, we are going to present Atomic Decomposition (AD), which is the modular structure induced by locality-based modules. We show how ADs are efficiently computed, and describe some of their properties. Finally, we discuss a consequence of ADs on building a model of an ontology.

Domagoj Vrgoc, University of Edinburgh

Regular expressions for data words

In data words, each position carries not only a letter form a finite alphabet, but also a data value coming from an infinite domain. There has been a renewed interest in these due to applications in querying and reasoning about data models with complex structural properties, notably XML, and more recently, graph databases. Logical formalisms designed for querying such data often require concise and easily understandable presentations of regular languages over data words.

Our goal, therefore, is to define and study regular expressions for data words. As the automaton model, we take register automata, which are a natural analog of NFAs for data words. We first equip standard regular expressions with limited memory, and show that they capture the class of data words defined by register automata. The complexity of the main decision problems for these expressions

(nonemptiness, membership) also turns out to be the same as for register automata. We then look at a subclass of these regular expressions that can define many properties of interest in applications of data words, and show that the main decision problems can be solved efficiently for it.