Report of 8th British Colloquium on Theoretical Computer Science

University of Newcastle upon Tyne, March 24-26th 1992.
(Sponsors Hewlett-Packhard, ICL and Logica)

BCTCS was again held close to the Easter festivities when the venue of Castle Leazes Halls of Residence of the University of Newcastle upon Tyne provided excellent amenities for the participants. From here excursions to the historic delights of Newcastle (the famous Tyne bridge with its great arching girders, the cathedral of St Nicholas, the 'New Castle', Eldon square and the museums amongst other delights within the city and then, for the more adventurous, the nearby numerous archeological remains of this cradle of English Chritianity such as those of the Island of Lindisfarne) were possible in conference breaks. John Tucker's speech of thanks at the conference dinner echoed everyone's appreciation of the excellent organisation of Iain Stewart and his local team. There was also opportunity for participants to thank John Tucker, the founding and now retiring Chairman of BCTCS, for his nurturing of the colloquium over the years to the extent that it is now an essentially successful and important part of the British conference calendar.

The colloquium was admirably supported this year by some forty technical presentations of wide diversity. In particular, the invited talks of seven distinguished guests added strength and balance to the programme. Abstracts of (contributed) talks may be found below.

BCTCS9 will be take place between the inclusive dates 29th-31st March 1993 at Wentworth College, University of York. Further information may be obtained from the local organiser, Hussein Zedan (Department of Computer Science, University of York, York YO1 5DD, UK; e-mail address: zedan@minster.york.ac.uk).

Alan Gibbons

Invited talks

The structural approach to complexity theory

J. L. Balcazar , Universitat Politechnica de Catalunya, Spain

Aspects of implementing P-RAM algorithms on distributed memory models of parallel computation

A. M. Gibbons , University of Warwick

Evolving Algebras

Y. Gurevich , University of Michigan, U.S.A.

The logical background to specification

W. Hodges , QMW, London University)

Design structures: configuring specifications

T. S. E. Maibaum , Imperial College, London University

VLSI analysis, synthesis and theory

J. Savage , Brown University, U.S.A.

Verification via model checking

C. P. Stirling , Edinburgh University

Abstracts of Contributed Talks

Formality in the analysis of timeliness requirements

T. Anderson, R. de Lemos and A. Saeed*, University Newcastle upon Tyne

We discuss the application of formal techniques to the analysis of timeliness requirements in safety-critical systems. To perform the analysis in terms of the different properties of interest, such as the physical process, sensors and actuators, and safety controller, different formalisms and underlying time structures are employed in distinct phases of analysis. We employ an event/action model for the analysis and specification of the timeliness requirements, and to establish a link between the timing constraints of the physical process and the safety controller. The event/action model is formalised in terms of a logic to facilitate the verification of specifications between two consecutive phases.

Universal properties of term graph rewriting

R. Banach , Manchester University

Term graphs are objects that locally look like terms, but globally resemble general directed graphs. Since their invention in the late 80's, they and their associated rewriting model have served both as a formalism for describing implementations of functional languages, and as a model of computation in its own right. The original semantics for rewriting term graphs is highly operational in character. This makes reasoning about the behaviour of term graph rewriting systems awkward and convoluted. Competing graph rewriting formalisms have tended to use categorical methods based on pushouts to express their semantics in a universal way. This approach does not work for all cases of generalised term graph rewriting. However, an alternative formalism, based on Groethendieck construction, can be used to give the semantics in a universal manner.

PRAM implementation on fine-grained MIMD multicomputers

F. Baude , Univ. Warwick

The more ideal tools to express data-parallelism are encompassed by the PRAM language mechanisms (mainly, global addressing space, implicit processes synchronization). In this paper, we evaluate the work needed to implement such mechanisms on the only class of parallel architectures that should prove to be really scalable as well as versatile, i.e. MIMD fine-grained multicomputers. The technique of PRAM programs implementation we will develop is valid for all possible MIMD fine-grained parallel machines and not only for one specific member (for example, not only for 3D-mesh based machines, but for all kinds of topologies). The point is that we consider an abstraction of these machines by reasoning in terms of the actor model of concurrent computation. The idea is to transform any PRAM program into an actor program such that it can be efficiently implemented on the target machine via mapping techniques such as for example graph embedding. This transformation is grounded on probabilistic theoretical simulations of PRAMs on parallel machines based on sparse communication networks.

Speeding up two string matching algorithms

*A. Czumaj** (University of Warwick), *M. Crochmore, T. Lecroq* (University of Paris 7), *L. Gasieniec, S. Jarominek, W. Plandowski and W. Rytter* (Warsaw University)

The talk deals with two string matching algorithms - the Boyar-Moore algorithm and its version called the reversed-subword algorithm. The main feature of both algorithms is that they scan the text from left-to-right from the supposed right position of the pattern. The Boyar-Moore algorithm goes as far as the scanned segment is a suffix of the pattern while the reversed-subword algorithm is scanning while it is a subword of the pattern. We show how to improve the Boyar-Moore algorithm such that it makes at most 2.42n comparison. We also speed up the

reversed-subword algorithm to make at most 2n comparisons. It will be shown that the average number of comparisons in the former algorithm is O((n/m)log m). The talk demonstrates the techniques to transform algorithms and shows new interesting applications of suffix trees.

A structured genetic algorithm

D. Dasgupta and D.R. McGregor* , University of Strathclyde

Genetic Algorithms (GAs) are iterative adaptive general-purpose search/optimization strategies, based on the principles of population genetics and natural selection. They simulate the mechanics of population genetics by maintaining a population of knowledge structures, analogous to the gene pool of a species, which is made to evolve. This paper investigates a new genetic model called the Structured Genetic Algorithm (sGA). The novelty of this new genetic approach lies primarily in its use of apparently redundant genetic material and a gene activation mechanism which in combination give a multi-layered structure to the chromosome. The additional genetic material serves to retain alternate candidate solutions in function optimisation. This paper presents important aspects of sGA and reports preliminary results of experiments performed to evaluate its performance in non-stationary function optimisation. In adapting to non-stationary environments of repeated nature genes (specifically building blocks) long-term utility can be retained for rapid future deployment when favourable environments recur.

An interpreter for a subset of FITL with concurrent operators

Z. Duan , University of Newcastle upon Tyne

Interval Temporal Logic (ITL) is a useful formalism for specifying and verifying the hardware and software systems. An executable subset of ITL has been developed as a programming language - Tempura. Further research works have been done to augment the ITL with frame and concurrent operators recently. It is suitable for specifying and verifying the concurrent programs. To put them in practice, we have been developing an interpreter for an executable subset of FITL with concurrent operators using Prolog language. In this paper, the outline of implementation of the interpreter is presented. Some new operators including Frame, Projection, Parallel, and Await etc. are introduced in detail. Some examples are also provided to explain how to apply the interpreter for programming and verifying of concurrent computations. Finally, we investigate some techniques for synchronization between parallel processes in our underlying logic.

Analysis of speed-up ratios in demand driven multiprocessor simulation algorithms

P.E. Dunne, C. Gittings and P. Læng* , University of Liverpool

Simulation is an important tool in the design and verification of digital logic systems. Demand-driven simulation methods exploit the fact that certain logic functions may be determined by inspecting only a few of their inputs, and thus offer a means of speeding up simulation time. Earlier work of the authors has concentrated on approaches to optimising signal selection methods in order to gain maximal benefit from this capability for lazy evaluation. While sequential simulation has proved amenable to the theoretical analyses required, problems arise when similar signal selection problems are considered in an environment where several processors are available to perform simulation. In a recent paper the authors showed how the analysis of the concurrent processor problem could be simplified by assuming that the ratio of 1-processor to 2-processor simulation time was constant and, in the same paper, values for these ratios for circuits over different logic bases were obtained experimentally. In this talk we present an analytic justification for the speed-up ratios used which confirms the experimental results obtained earlier.

Formal design of synchronous concurrent algorithms

M.V.H. Fairtlough , University of Edinburgh

I describe research on the formal synthesis/verification of a parameterized family $\{OE_n : n \in \mathbb{N}\}$ of synchronous concurrent sorting algorithms (the odd-even transposition sorters of size 2n+2), give the top-level specification in LEGO as a (parameterized) primitive recursion over vectors, show how to verify that a sorting algorithm has indeed been specified, and how to turn the specification into an algorithm that operates as a vector-stream transformer. I also show how to give a corresponding specification in HUL using lists rather than vectors and sketch how an implementation of the expanded OE algorithm could be defined in terms of a network of modules connected by wires.

From Dijkstra's Fusc to Lucas Fractions

D. Gault , Queen's University of Belfast

Dijkstra, in EWD570 Selected Writings on Computing: A Personal Perspective, discusses a recursively defined function, fusc, and presents an algorithm for its efficient evaluation. The algorithm is not a direct implementation of the specification of the function. One of Dijkstra's aims in presenting fusc and its implementation was to deliver a challenge to advocates of the transformational development of programs who might find difficulty deriving his program using their techniques. In the nineteenth century, Lucas published "Theorie des Nombres", in which he descibes various mathematical sequences. One of the sequences differs from the others in that it consists of rational numbers rather than integers. Lucas discussed briefly how it might be generated but did not supply any proof that this method of generation would always retain the essential properties of the values of the sequence. In this paper it is shown that the Lucas Fractions may be defined in terms of Dijkstra's function fusc. Since Dijkstra gives no indication in his paper that fusc is anything other than an interesting example to be used in support of a particular claim, it is gratifying to discover that the function has an application in quite a different context.

Confluent CCS and interaction nets

S.J. Gay , Imperial College, University of London

Two models of communicating processes are considered. Confluent CCS is a deterministic fragment of Milner's calculus of communication systems, a model based on processes which change state via communications with other processes. Lafont's interaction nets is a graph rewriting system based on the proof nets of Girard's linear logic; a graph rewrite can be viewed as a collaboration between two processes which replace themselves with another network. A translation from confluent CCS to interaction nets is described, with the property that every tau transition in confluent CCS corresponds to one or two graph reductions in interaction nets and every intermediate graph corresponds (up to observation congruence) to a confluent CCS process.

Listing first order graph properties

L.A. Goldberg , University of Edinburgh

For every sentence θ in the first order language of graphs let $G_\theta(n)$ denote the set of n-vertex graphs that satisfy θ and let G_θ denote the family of sets $\{G_\theta(1), G_\theta(2), ...\}$. In this work, we consider the following problem. Given a sentence θ in the first order language of graphs, design a fast algorithm that takes as input a positive integer n and lists the members of $G_\theta(n)$. The criterion for "fast" we use is polynomial delay. We say that a listing algorithm for G_θ has polynomial delay if and only if there is a polynomial p such that when the algorithm is run with input n it executes at most p(n) machine instructions before each successive member of $G_\theta(n)$ is output. The results that we obtain are related to a theorem of Fagin. Fagin showed that for every sentence θ in the first order language of graphs

the proportion of n-vertex graphs that satisfy θ is either 1-o(1) or o(1). In the former case we call G_θ a first order one property and in the latter case we call it a first order zero property. In this talk we will describe a general method that can be used to obtain a polynomial space polynomial delay listing algorithm for any first order one property.

Visualizing mathematical structures

C. Holt, University of Newcastle upon Tyne

Many of the structures that arise in mathematics can be represented as directed graphs. Lattices have values as nodes and immediate partial order as arcs. Groups have values as nodes and arcs labelled by values, such that an arc y with source x has target yx. Proofs have axioms, rules of inference, and theorems as nodes; arcs link these together as deductions. Commutative diagrams are clearly directed graphs. The ubiquity of such structures, and the problems inherent in equational representations, make it worth looking to find an alternative approach for their description. Recent work in visualizing program structures (which are also directed graphs) can be adapted to relatively pure mathematics. This paper examines the clarity of the resulting structures.

PARTY : A realistic real-time process algebra

C. Ho-Stuart, M. Fang and H. Zedan*, University of York

Many real-time process algebras have been proposed in recent years, but they invariably make restrictive assumptions about the kinds of behaviour which can be described, or else do not provide a sound and complete proof theory for a sufficiently large class of behaviours. We introduce PARTY, the Process Algebra with Real-Time from York, which uses a simple yet powerful and general model of real-time behaviour. Operational semantics give a tree structure for a process. A path through the tree corresponds to a sequence of time stamped events. A dense time domain can be used. Unlike TCSP, there is no lower limit on the time between events. Unlike many real-time extensions to CCS, internal activity can be completely hidden from the environment. A hierarchy of equivalence relations can be defined, corresponding to bisimulations, failure equivalence (CSP), and refusal equivalence (TCSP). All these equivalences are congruences for the PARTY language.

Periodic schemes for disseminating information in processor networks

Y. Igarashi, Gunma University, Japan

Data broadcasting is a very fundamental operation in parallel and distributed computer systems. It can be accomplished by the data disseminating process without physical broadcast in such a way that each processor in a network repeatedly receives and forwards messages. In this paper we describe a class of information disseminating schemes called periodic schemes in a general form. The fault tolerance of some schemes in the class is discussed. We show sufficient numbers of rounds for broadcasting in hyper-cube networks, binary jumping networks and some variations as functions of the numbers of processors in the networks and the numbers of faulty processors and/or links. Some of our results are shown to be tight (i.e., they are also necessary numbers of rounds in the worst case).

Simple translation-invariant concepts are hard to learn

M.R. Jerrum, University of Edinburgh

The concept class TCM of 'translation-closed monomials' was proposed by Maragos and Valiant as a natural starting point for the investigation of the computational complexity of learning translation-invariant concepts. Concepts in TCM are (satisfying assignments to) DNF formulas such as :

$$(x_0 \wedge x_1 \wedge x_3) \vee (x_1 \wedge x_2 \wedge x_4) \vee \ldots \vee (x_4 \wedge x_0 \wedge x_2)$$

(over the variables $x_0, x_1, \ldots, x_4$) which are generated from a single monomial (conjunction of variables) by

cyclically permuting indices and forming a disjunction. Note that concepts in TCM are invariant under cyclic permutation of the variable set. The computational complexity of learning TCM concepts is investigated within the Valiant (PAC) model. Despite their obvious simplicity, TCM concepts are apparently difficult to learn

On stepwise explicit substitution

F. Kamareddine and R. Nederpelt* , University of Eindhoven, The Netherlands

This paper starts by setting the ground for a lambda calculus notation that strongly mirrors the two fundamental operations of term construction, namely abstraction and application. In particular, we single out those parts of a term, called items in the paper, that are added during abstraction and application. This item notation proves to be a powerful device for the representation of basic substitution steps, giving rise to different versions of β-reduction including local and global β-reduction. In other words substitution, thanks to the new notation, can be easily formalized as an object language notion rather than remaining a meta language one. Such formalization will have advantages with respect to various areas including functional application, lazy evaluation and the partial unfolding of definitions. Moreover our substitution is, we believe, the most general up to date. This is shown by the fact that our framework can accommodate most of the known reduction strategies.

Automatic Verification of Interval Temporal Logic

S. Kono , University of Newcastle upon Tyne

Interval Temporal Logic is a simple and powerful temporal logic, and it has decidable propositional subset. Here we show efficient verification algorithms. First method is based on tree based subterm classification and determinization of tableau method. The other one features binary decision diagram and its fixpoint calcuration. These are written in Prolog and it works faster than automaton based method written in C.

Partial Metrics

S.G. Matthews , University of Warwick

Scott models are topological models of complete partial orders used for Tarskian fixed point semantics of the lambda calculus. As of yet there are no methods for deriving Scott models from specifications of the"complete" objects beyond an arbitrary choice. This talk introduces "partial metrics" for generalising a theory of complete objects into a Scott model including partial objects.

Consistent closed sets and logical inference

N. Measor , University of Leicester

In the case of classical logic, the set of deductively closed consistent sets of formulas forms a consistently complete algebraic cpo. Hence, by Zorn's Lemma we are assured of the existence of maximal consistent sets which serve as valuations for the purposes of completeness proofs. Furthermore the deductively closed sets form the elements of an information system in the sense of Scott. These observations, although not novel, provide a good framework to compare the properties of various logical systems. We consider the extent to which the same properties hold for various non-standard logics such as three-valued logic, autoepistemic logic, and linear logic.

Fast string matching on the average

K. Park , Kings College, Univ. London

The string matching problem is: Given a pattern string of length m and a text string of length n, find all occurrences of the pattern in the text. Based on various criteria, more than one hundred algorithms have been developed for string.

matching. In this paper we present a new algorithm for the string matching problem which is simple and elegant, but that satisfies many criteria simultaneously; that is,

(1) O(n) search time in the worst case,

(2) O(nlog m/m) expected time when the strings are random,

(3) small number of text characters examined in experiments, and,

(4) fast in actual running time. Our algorithm preprocesses the pattern in O(m) time, which is common in almost all algorithms.

Fast multipliers and wise investments

*M.S. Paterson** (Univ. Warwick) and *U. Zwick* (Tel Aviv University, Israel)

Networks of carry-save-adders are often used to sum n binary numbers in O(log n) time. Such multiple addition is crucial for fast multipliers. A new extension of this classic design is presented which results in multiplication circuits with the smallest known depth. The construction involves the same skills as in a prudent portfolio selection from diverse investment schemes in each of which payments and dividends may be in various currencies.

New results on lines in 3-Space

M. Pellegrini , Kings College, University of London

This talk surveys recent results for problems in 3-dimensional geometry involving polyhedra, lines, point, and spheres. Particular attention is devoted to the following problems: given a set of polyhedra in $\mathbb{R}^3$ solve efficiently ray shooting queries; detect efficiently if a simplex is collision-free among obstacles; find the sphere of minimum radius meeting a set of lines.

Modal Logics with Past for True Concurrency

S. Pinchinat and P. Schnoebelen*, Lifia-Imag, France

We investigate modal logics characterizing behavioral equivalences for "true" concurrency. Past-time operators are used for this purpose. We define HML_bfp, a simple modal logic over labelled events structures, based on backward and forward pomset observations, and prove it characterizes history preserving (h.p.) bisimulation. This result relies on a new characterization of h.p. bisimulation as back and forth pomset bisimulation. We also prove that the L_P logic of [De Nicola et al., 10th FST&TCS, 1990] (a modal logic combining linear and branching formulae) also characterizes h.p. bisimulation. This result relies on a two-way translation between HML_bfp and L_P.

Graphical methods for information commodity assessment

D. Preston , South Bank Polytechnic

We are concerned in estimating the value of production of an Information Commodity. Here value is that normally associated with supply and demand. We consider the production of Information Commodities: how they acquire value. This leads to an introduction of the graphical tool used. We report on other methods used and suggest why our method appears more practicable. We discuss how our software tool exists in prototype and report on its use. It would appear further research is needed in this important area and offer suggestions as to how this may proceed.

Database theory: sets versus categories

B.N. Rossiter and M.A. Heather*, University of Newcastle upon Tyne

The subject of databases is both theoretical and very practical but theory has hardly kept pace with the demand-driven momentum of practical applications. The power of databases is to operate across many levels coherently from the

conceptual schema right across to the bit address on the physical storage medium, but at the time database theory was developing, it was necessary to customise notation from set theory to represent transformations of common data structures at an abstract level. Standard database texts indicate a need for the use of higher level abstract formalism. An example of an enhanced set theoretical notation which today could be better represented in a subject like category theory is Ullman's need to define functional dependencies by the set F related to F^+ (the closure of F) in the following way: $F^+ = (X \rightarrow Y : F \models X \rightarrow Y)$ This paper compares the set theoretic and Z language form of representation for database modelling to the use of category theory.

Refinement of real-time systems in TAM

D. Scholefield and H. Zedan* , University of York

The "temporal agent model" (TAM) is a wide-spectrum development language in which descriptions of real-time systems may be expressed at any level of abstraction - from requirements specification to concrete design. The TAM language is given a weakest pre-condition semantics (extensions of Dijkstra's semantics to cover time and concurrency). A refinement relation is then defined which models our intuition of what it means to derive a concrete design from an abstract specification. Refinement laws - sound with respect to this relation - are provided in order to enable the user to develop real-time programs from initial requirements specifications.

Simulated annealing on fractal energy landscapes

G.B. Sorkin , University of Edinburgh

Existing analyses of simulated annealing do not adequately explain the effectiveness of the algorithm, in particular failing to find why annealing is better than application of the Metropolis procedure at some fixed temperature. We show that for a class of deterministic fractals, annealing with a geometric cooling schedule produces a solution of expected cost ξ in time which is a power of $1/\xi$. The power depends on parameters describing the problem instance, but is not on the number of dimensions. Annealing is much more efficient than Metropolising at any fixed temperature, and, for high-dimensional problem instances, than hillclimbing, repeated hillclimbing, or random sampling. The fractal model is unrealistic, although motivated by properties observed in real-world problem instances. More natural problems are currently under investigation. The analysis relies on a general theorem regarding annealing with monotonically nonincreasing temperatures. This, and a surprising example of what can go wrong with oscillating temperatures, are of independent interest.

Match-free permutations of multistrings: an algorithmic analysis of derangements

B.R. Stonebridge , University of Bristol

This paper describes ongoing work relating to the enumeration of derangements of the symbols of multistrings. The problem of deranging symbols which belong to, possibly discontiguous, internally ordered, blocks in a multistring is that of rearranging them so that none is in its own original position or that of an identical symbol. It might be supposed that by the simple expedient of dividing through by the number of permutations within the individual blocks the number of derangements would result. However, as we prove for the case of a string of n symbols with a substring of r symbols, this is only true for the limiting case when n is large compared with r. For intermediate cases there is interference between the symbols of a block, reducing the number of derangements and making the analysis more complex, as is shown by the resulting formulae. Having considered an algorithm for a single block of identical symbols, we extend the analysis to the case of two distinct blocks and from this we then generalize to the formula for multiple blocks. We also give the asymptotic form for n large compared with the block size.

Axiomatization of Calculus of Constructions without infinite type hierarchy

Y. Sun , University of York

This paper gives an axiomatization of Calculus of Constructions (CC) [Coquand and Huet 1988] without infinite type hierarchy in a Framework for Binding Operators (FBO) [Sun 1991]. Specifically, we introduce an extra universe kind above the original universe type in CC. But we allow neither quantification over kind nor introduction of y : kind. Another operator → is added for typing π types in FBO, since we are only considering a single-sorted FBO. Also, every CC term is typed in FBO. For example, πx : M.N in CC is translated as π y.u : t → v in FBO, where x, M and N correspond to y, t and u respectively under the translation. Hence, as a result of our axiomatization of CC in FBO, we know that a countably infinite hierarchy of type universes for CC may be not necessary.

One-counter groups

*R.M. Thomas** (University of Leicester) and
T. Herbst (Standard Elektrik Lorenz AG, Germany)

Given a presentation for a group G, we define the word problem of G (with respect to that presentation) to be the set of all words, in the generators and their inverses, that represent the identity element of G. If G has word problems W1 and W2 with respect to presentations P and Q, respectively, and if F is a family of languages closed under inverse homomorphism, then W1 ∈ F if and only if W2 ∈ F; so in this case, we may say that the word problem belongs to F without reference to the presentation. In this talk, we give several characterizations of groups whose word problem is a one-counter language (i.e. is accepted by a one-counter pushdown automaton).

Algebraic specifications and computable functions

J.V. Tucker , University College Swansea

First, we consider the functions computable over any many-sorted algebra A. Secondly, we consider the question: if f is computable on A and A has an algebraic specification (under initial algebra semantics) then does the algebra (A,f) have an algebraic specification (under initial algebra semantics)? We give theorems that provide strong affirmative answers to the question, and discuss their applications. This work is part of the author's programme with J.I. Zucker (McMaster) on generalising computability theory to classes of many-sorted algebras.

Modelling concurrent processes in logical circuits

A.V.Yakovlev , University of Newcastle upon Tyne

The talk contributes to the theory of asynchronous logical circuits (ALCs) and applications and theory of concurrency. ALCs are free from any explicit timing mechanism and therefore can exhibit highly concurrent switching behaviour. It is often important to model and analyse their behaviour in terms of causality and ordering relations between individual signal transitions. Signal transition graphs (STGs), a model based on interpreted Petri nets (PNs), have been identified as one of the most convenient formal tools for specification, verification and synthesis of ALCs. We pose some recent advances and open problems related to developing efficient algorithms for checking a well-known Unique State Assignment property of STGs. We demonstrate limitations and extensions of STGs to accommodate some important behavioral paradigms of ALCs. Using syntax and semantics of extended STG, we characterize the behaviour of a crucial class of ALCs, semi-modular circuits, which are free from hazards in their operation. We prove semantical links with other related models such as change diagrams and labelled causal automata. We also examine the general effect of net transition labelling upon the semantics of interpreted PNs.