Information Systems & Electronics Development Lab. ,
Mitsubishi Electric Corp. , 5-1-1 Ofuna Kamakura 247 Japan

Abstract: We present a new method for test sequence generation for communication systems. This method , which is called SW method, is based on the finite state machine model and can generate shorter length of sequence than PW method. And this method is garanteed the existency of the test sequence.

Report on the Fourth British Colloquium for Theoretical Computer Science (BCTCS4) University of Edinburgh, 28th–31st March 1988

BCTCS4 was held in Easter week amidst the granitic elegance of Edinburgh. Organised under the sure hand of Mark Jerrum of the University's Department of Computer Science, the event proved to be a happy and relaxed one. Due to his efforts and those of the organising committee, the local arrangements worked well.

Of the thirty technical presentations, the following were delivered by the distinguished group of invited speakers: *A Cook's tour of the finitary non-well-founded sets* (Samson Abramsky, Imperial College, London), *The Edinburgh LF: a framework for describing formal systems* (Furio Honsell, University of Edinburgh), *What use are process algebras?* (Robin Milner, University of Edinburgh), *Computationally feasible learning* (Leslie Valiant, Harvard University) and *Gröbner bases: complexity and applications* (Chee Yap, Courant Institute, New York). As can be judged from the list of abstracts of contributed talks that follows, there was also great variety and interest within the programme at large.

Amongst the majority of UK based theoretical computer scientists attending the colloquium, there was a detectable increase in the overseas visitors. Of course, this welcome trend served to enrich the proceedings and adds a general feeling that this is an annual event of growing importance. The Annual General Meeting, held within the colloquium timetable, confirmed that the venue for BCTCS5 would be Royal Holloway and Bedford New College, London. BCTCS5 will take place from 11th to 13th April 1989; further particulars can be obtained by writing to: Costas Iliopoulos, Department of Computer Science, Royal Holloway and Bedford New College, Egham, Surrey, TW20 0EX, England.

Alan Gibbons

BCTCS4 was sponsored by Hewlett-Packard Limited.

Invited Talks

Samson Abramsky, Imperial College, London, *A Cook's Tour of the Finitary Non-Well-Founded Sets.*

ABSTRACT: We present an example to illustrate a number of themes. The themes are:

- The alternative description of semantic models, and their systematic derivation from each other.
- The importance of the finitary (potential infinities arising as limits of the strictly finite) in the theory of computation, and of topological ideas as a framework for notions of finitariness.

The example comprises the finitary fragment of Peter Aczel's universe of non-well-founded sets; apart from connecting with a wealth of ideas in models for concurrency, this has interesting logical properties when conceived as a universe of sets in its own right (significantly different from those of Aczel's theory).

We call this universe $\mathcal{F}$, and give five descriptions of it:

1. As the metric completion of an ultrametric derived from a process tree representation of the hereditarily finite sets.
2. As the solution of a domain equation in a category of compact ultrametric spaces.
3. (Qua topological space) as the solution of a domain equation in the category of Stone spaces.
4. As the dual space of the solution of a domain equation in the category of Boolean algebras, hence as the free modal algebra.
5. As the subspace of maximal elements of the solution of a domain equation in the category of domains.

The solutions illustrate the relationships between the Hausdorff powerspace construction on metrics, the Vietoris construction on topological spaces, and on logical theories, and the Plotkin powerdomain.

We then describe some features of set theory over $\mathcal{F}$. Union, pairing, powerset, emptyset, and (a suitable version of) infinity are all satisfied; this falls easily out of the above descriptions. Foundation is *not* valid, and indeed we have the ultimate Anti-Foundation axiom: the universe is a set. *Continuous* versions of separation, replacement and choice (i.e., where the functions involved are continuous) are valid. Thus we get a computationally significant, topological approximation to set theory.

Furio Honsell, University of Turin, *The Edinburgh LF—a Framework for Describing Logical Systems.*

ABSTRACT: The need to provide machine assistance in the manipulation of logical systems is keenly felt in Computer Science, where the proliferation of logics is astonishing. The ability to generate proof editors and proof checkers depends however, on the existence of a logic-independent model of the process of deriving and checking proofs.

The Edinburgh Logical Framework (LF), developed at the LFCS by Arnon Avron, Robert Harper, Furio Honsell, Ian Mason and Gordon Plotkin is a first step towards determining such a model. It provides possible formal answers to fundamental questions like: what is a logic? what is a logical language? what is an assumption? what is logical dependence? what is a rule? what is a proof?

The LF framework is based on a predicative typed lambda-calculus with dependent types. Central to it is a "judgements/assertions as types" principle. This principle makes it possible to encode uniformly in LF logical systems in a natural deduction style. LF can thus be viewed as a specification standard for logics.

A few examples of how logics are represented in this framework will be given.

The prototype implementation of the LF which Timothy Griffin has constructed using the Cornell Synthesizer Generator will be discussed.

Robin Milner, University of Edinburgh, *What use are Process Algebras?*

ABSTRACT: It was argued that process algebras (PAs) like TCSP, ACP and CCS, taken alone, are not convenient and apparently not sufficiently expressive to specify parallel systems. ("Apparently", because if we consider a set of PA equations in a process variable X to be a specification of X, then no-one has yet measured the power of this specification method in comparison with a given specification logic). However, a PA accompanied by a process logic (PL) is a good candidate for the task. Here, a specification is of the form "P satisfies F", where $P \in$ PA and $F \in$ PL.

The relation between bisimulation equivalence in an appropriate PA on the one hand, and the formulae of a particular PL (Hennessy-Milner logic, HML) was reviewed—namely, that two processes $P, Q \in$ PA are bisimilar iff they satisfy exactly the same formulae of HML. This property of a PA/PL combination makes it a good substrate for a similar combination at a higher level, for example the combination of a parallel imperative programming language and an associated Hoare logic.

It was shown that one such combination rests firmly upon the CCS/HML substrate in a precise sense, namely

- The programming language is a derived algebra of CCS;
- The Hoare logic is a derived logic of HML;
- The rules over Hoare triples can be proved sound from the substrate.

It remains to prove similar results for other combinations, and for other substrates which are suitably primitive.

L. G. Valiant, Harvard University, *Computationally Feasible Learning.*

ABSTRACT: A distribution-free model of inductive learning is discussed with particular reference to the learning of Boolean expressions. Recent results concerning classes of representations that are learnable in polynomial time are reviewed. These include both positive and negative results and highlight the critical role of knowledge representation.

Learnability is not monotonic. There are natural examples of nonlearnable classes that become learnable when suitably extended.

Chee Yap, Courant Institute, N.Y.U., *Gröbner Bases: Complexity and Applications.*

ABSTRACT: Gröbner bases have generated considerable interest among constructive algebraic geometers and computer scientists. A Gröbner basis is a finite set of generators for a polynomial ideal with properties that makes it useful in a wide variety of applications. Until recently, very little was known about the complexity of constructing Gröbner bases. It is now known that a minimal Gröbner basis has degree $D \leq d^{2^n}$ where d is the maximum degree of the input polynomials on n variables [Dube, Guisti, Mora-Moller]. We show that $O(D^{4n+2})$ term-wise operations suffice for constructing such bases and describe data-structures useful in constructing Gröbner bases.

A number of applications will be described. In particular, a geometric editor system LINETOOL (jointwork with Lars Ericson) that we are constructing uses Gröbner bases. For this application, recent results of [Brownawell, Caniglia-Galligo-Heintz] yield the sharper bound $D \leq d^n$.

Contributed Talks

Meurig Beynon, University of Warwick, *Combinatorial Models for Monotone Boolean Functions.*

ABSTRACT: The study of monotone boolean functions (MBG) can be viewed as the study of free distributive lattices. Computational equivalence and replacability have interpretations in this setting, leading to a novel kind of algebra: a distributive lattice with an auxiliary partial-order with respect to which is both a partially-ordered algebra, and an abstract simplicial complex. Via this theory, formulas and circuits for a MBF f can be interpreted combinatorially in terms of an array R_f whose rows and columns are respectively indexed by its prime implicants and prime clauses, of and whose ijth entry is the set of indices of inputs common to the ith implicant and the jth clause.

Combinatorial piecewise linear maps provide an alternative model for MBFs, motivated by computational geometry. The structure of such maps can be viewed in terms of systems of singular chains and cycles with the Cayley diagram of the symmetric group S_n in a standard presentation. The presence of such cycles can be interpreted in terms of configurations within R_f.

Paul E. Dunne, University of Liverpool, *Algorithms for Improving the Efficiency of Digital Simulation Methods* (joint work with **Paul H. Leng**).

ABSTRACT: Simulation is an important stage in designing and verifying digital logic circuits. Technological advances have resulted in circuits of great size and complexity becoming feasible and such developments have led to corresponding problems in performing simulation processes efficiently.

In this talk we briefly discuss some classical simulation approaches and their drawbacks. We then present a circuit model which seeks to exploit a property of certain gate operations—that their result may be determined without evaluating each input—in order to reduce simulation time. For this model, with certain simplifying assumptions, we present a simulation algorithm which is optimal in the sense that it provably minimises simulation time.

Martin Dyer, University of Leeds, *A Randomised Algorithm for Fixed-dimension Linear Programming.*

ABSTRACT: A Las Vegas randomised algorithm of complexity $\mathrm{O}(c^{d \log d} n)$, for some $c > 1$ is described. This improves the best deterministic result of $(3^{d^2} n)$ for fixed-dimension linear programming.

Alan Gibbons, University of Warwick, *Optimal Parallel Expression Evaluation on the P-RAM* (joint work with **Wojciech Rytter**).

ABSTRACT: We describe a deterministic parallel algorithm to evaluate algebraic expressions in $\mathrm{O}(\log n)$ time using $n/\log n$ processors on a parallel random access machine without write conflicts (P-RAM) and with no free preprocessing. The input to the algorithm is a string (of the symbols making up the expression) stored in an array. Such a form for the input enables a consecutive numbering of the operands in the expression in $\mathrm{O}(\log n)$ time with $n/\log n$ processors. This corresponds to a consecutive numbering of the leaves of the expression tree. This then further permits us to partition the leaves into small segments. We improve the result of Miller and Reif (1985), who described an optimal parallel randomized algorithm. (Strictly speaking, the input to their algorithm is different, being the parse tree of the expression. The input to the innovative part of our algorithm (step 2) is this parse tree which, in addition, has its leaves numbered consecutively from left to right. These two forms are equivalent if we note that such a numbering can be obtained by an optimal parallel algorithm which employs the Euler tour technique and optimal list ranking). Our algorithm can be used to construct optimal parallel algorithms for the recognition of two nontrivial subclasses of context-free languages: bracket and input-driven languages. These languages are the most complicated context-free languages known to be recognizable in deterministic logarithmic space. This strengthens the result of Matheyses and Fiduccia (1982) who constructed an almost optimal parallel algorithm for Dyck languages, since Dyck languages are a proper subclass of input-driven languages.

Our algorithm includes a new simple method for tree contraction which we call the leaves-cutting method. Its correctness is trivial (compared with the method of Miller and Reif) and it can be implemented on a P-RAM without write and without read conflicts.

Mike Holcombe, University of Sheffield, *Formalising the "Unformalisable"—from Saint to Psychopath.*

ABSTRACT: The problems of reasoning about user behaviour were discussed in the light of total system design. A formal model of the user's concept of a system was examined. The model consists of an experimental data base, a goal space and a logic processing unit. A denotational semantics for the experimental data base (based roughly on the headed record/file approach) was given. A semantic network type of goal space was considered. The logics considered as a basis for the system processing were the belief logics and society of minds ideas of Fagin and Halpern and Levesque, adapted to a fuzzy logic setting. The communication between user and system was considered in the light of an X-machine.

Mike Holcombe, University of Sheffield, *Developing Practical Theoretical Skills.*

ABSTRACT: We ask questions like "Why do we teach theoretical computer science?", "How do students benefit?", etc. We discuss *aims* and *objectives* of such courses. Three classes of aims we considered—knowledge, applications and skills.

Investigational, modelling and analytical skills are stressed in the context of abstracting situations/processes in computer science.

Experiences in introducing innovative approaches into the curriculum, including investigational work, group work, software huts, etc. are discussed. The problems of integrating formal methods of design, based on theoretical foundations into the curriculum are explored.

Costas Iliopoulos, Royal Holloway & Bedford New College, London, *Coarsest Set Partition Problems and Multiprocessor Architectures.*

Maciej Koutny, University of Newcastle, *Synchronous and Asynchronous Communication in Dynamically Structured Systems* (Joint work with **R. P. Hopkins**).

ABSTRACT: The Dynamically Structured Communicating Systems (DSCS) model provides an interleaving model of concurrency using a communication tree representation of processes, similar to that in CCS but enhanced with some configuration information in the nodes. This allows the representation of a process which dynamically changes the connectivity of a system of processes of which it is a part. The model also accommodates both synchronous communication, where the send and receive of a message constitute a single event which synchronises the sending and receiving processes, and asynchronous communication, where a send can occur as one event with the matching receive occurring later and thus there may be an unbounded number of outstanding messages. These

two modes of communication are captured in two parallel composition operators on the same communication tree representations. The observable aspects of process behaviour are captured in the definition of observational equivalence which extends the concept introduced for CCS processes to a class of systems with dynamic structure.

Ralf Kneuper, University of Manchester, *A Formal Framework for Symbolic Execution.*

ABSTRACT: The aim of this talk is to formalise the notion of symbolic execution, handling specifications as well programs. As a first step, the notions of execution and specification are discussed. The meaning of a specification (or program) is considered to be a relation on states. Interpreting (or executing) a specification consists of a state transformation satisfying the meaning relation.

Similarly, symbolic execution is described as a transformation on symbolic states, where symbolic states map identifiers to symbolic values. Symbolic states denote a more general concept than sets of states, which allows describing the connection between actual input and output states. It can then be shown that symbolic execution has a number of properties that one would expect intuitively.

Marta Kwiatkowska, University of Leicester, *Towards a Formalisation of Fairness.*

ABSTRACT: The purpose of the research is to examine existing notions of fairness in uniform setting. Considering the multiplicity of models for parallelism and a large number of different definitions of fairness, often informal, subjective and model-dependent, this task seems to be of great importance. A model-oriented approach, rather than proof-oriented, has been taken here. The model chosen is (labelled) asynchronous transition systems [Shields] [Bednarcyzk] extended with ambiguity and process structure. Potential problems with straightforward translation of existing notions of fairness into the model are shown, in particular the fact that some fairness notions are not equivalence robust.

Colin McDiarmid, Oxford University, *Building Heaps Quickly.*

ABSTRACT: This talk is based on joint work with Bruce Reed (then at Bellcore). It presents a natural variant of Floyd's method for building heaps, which uses on average $\alpha + o(1)$ comparisons per element, where $\alpha \simeq 1.52$. This average complexity is the best known (and best possible?). Further, on the overwhelming proportion of inputs the algorithm uses close to the average number of comparisons.

Faron Moller, University of Edinburgh, *Non-finite-axiomatisability in Process Algebras.*

ABSTRACT: We consider the problem of equationally axiomatising congruences in process algebras. As our main result, we show that observational congruence over a simple subset of finite terms of Milner's language CCS cannot be finitely axiomatised. That is, the well-known Expansion Theorem of CCS cannot be replaced by any finite set of equational axioms.

We then extend our result to show that no reasonable notion of congruence contained in observational congruence can be finitely axiomatised. Thus in the absence of an elegant axiom schema such as the Expansion Theorem, it is necessary to go outside the signature of the language in order to axiomatise the congruence. Such is the case, for instance, with the introduction of the left merge operator in the work of Castellani and Hennessy on axiomatising their distributed bisimulation congruence.

Karl Meinke, University of Leeds, *Specification and Representation of Synchronous Concurrent Algorithms.*

ABSTRACT: We introduce a graph-theoretic representation language for synchronous concurrent algorithms together with five specification languages for the classes of:

(i) computable functions;

(ii) computable stream transformers;

(iii) primitive recursive functions;

(iv) primitive recursive stream transformers;

(v) polynomial functions;

over an abstract structure A. For each of these classes we identify a class of synchronous algorithms which implement precisely the functions of that class. The proofs are constructive and define compilers from simple register machine languages to synchronous concurrent algorithms.

Maurice Naftalin, University of Stirling, *Correctness for Beginners.*

ABSTRACT: Stepwise refinement can be formalised in a natural way by regarding specifications as unimplemented program components. We present a graphical notation for specifications and their refinement rules which supports this approach. The resulting development style is proposed as an appropriate model for introductory programming instruction.

Luke Ong, Imperial College, London, *The Full Abstraction Problem in Lazy λ-calculus.*

K. V. S. Prasad, Chalmers University of Technology, Göteborg, *Combinators and Bisimulation Proofs for Restartable Systems.*

ABSTRACT: We are interested in carrying out bisimulation proofs by hand, rather than by machine. Other researchers, Larsen and Milner for example, have noted that it is important to develop new notations and formulations to suit the problem at hand, rather than expect a single calculus to work adequately for all examples.

We report a similar experience, working with restartable systems as an application area. We develop a set of combinators to reflect the behaviour of such systems, and discover that a separation of normal and exception behaviours across all components of the system is possible under a natural condition which is best described by synchronisation. Since our system is primarily asynchronous, we are lead to develop an asynchronous calculus that permits synchronisation. With this calculus, and the new operators, we can finally capture concisely and naturally the behaviour of a simple restartable system.

Two theoretical issues arise naturally from this work. One is the relation between our new calculus, MCCS, and Milner's older calculi SCCS, ASCCS and CCS. We show that MCCS fits between the last two, and that it clarifies some issues previously left unresolved. Finally, we ask if our new operators are derivable from the standard operators in SCCS and CCS. This leads to general study of derivability of operators. We develop a technique to show non-derivability, and show that many of the standard operators are primitive, i.e., they cannot be derived from the others.

A. W. Roscoe, Oxford University, *Here be Dragons! An Expedition into the Land of Unbounded Nondeterminism.*

ABSTRACT: Unbounded nondeterminism leads to mathematical difficulties arising from the need to reason about infinitary behaviours explicitly. We introduce a model for unboundedly nondeterministic CSP where each process is identified with a triple $\langle F, D, I\rangle$—

failures, divergences and infinite traces. All the usual CSP operators plus infinite hiding and general nondeterministic composition have natural interpretations.

Unfortunately the model is an incomplete partial order and several operators are noncontinuous (though monotone). However the existence of fixed points for all CSP recursions and the correctness of the semantics is provable via a congruence theorem with a Plotkin-style operational semantic.

J. S. Shawe-Taylor, Royal Holloway and Bedford New College (University of London), *Computing Inverse Images of Domain Open Sets.*

ABSTRACT: The uses of domain open sets in studying programs are discussed. After indicating how higher order domain open sets might be denoted, Dybjer's example of open set expressions for a data domain is considered in detail. The problems that can arise when Dybjer's method of computing inverse images is used are enumerated. One solution to these problems with pleasing theoretical properties is presented. It is argued, however, that the solution is not of practical significance because it fails to make use of the structure of the recursion. An algorithm is presented which computes all possible inverse computations of the body of a recursive program. It is indicated how the results of this algorithm can be combined heuristically to give approximate inverse image computation for recursive programs.

Mike Shields, University of Kent, *Solving Context Equations.*

Harold Simmonds, University of Aberdeen, *Logic for IT.*

Alistair Sinclair, University of Edinburgh, *Approximating the Permanent.*

ABSTRACT: Computing the permanent of a square 0-1 matrix, or equivalently counting perfect matchings in a corresponding bipartite graph, is a well-known #P-complete enumeration problem and as such almost certainly intractable. In this talk, we describe a randomised approximation algorithm which with arbitrarily high probability produces an output which estimates the number of perfect matchings in a given (bipartite) graph with arbitrarily small relative error. The basic approach, which was first proposed by Andrei Broder, involves generating matchings from an almost uniform distribution by simulating a suitable ergodic Markov chain. Using a novel method for analysing the rate of convergence of Markov chains, we are able to show for the first time that this algorithm is efficient, provided the minimum degree of the graph is large. (The problem remains #P-complete under this restriction.)

The method also works for a. e. random bipartite graph with $2n$ vertices and density at least $c \log n/n$ for some constant c, and can be extended to approximately count *all* matchings in an arbitrary graph.

Iain A. Stewart, University of Newcastle upon Tyne, *Colouring Perfect Planar Graphs in Parallel.*

ABSTRACT: We present a parallel NC algorithm that colours a perfect planar graph using at most 4 colours. The algorithm uses $O(n^3)$ processors and takes time $O(\log^4 n)$ when implemented on an EREW PRAM. We present a sequential algorithm, which we develop into a randomized parallel algorithm. This randomized parallel algorithm is then converted into a deterministic parallel algorithm using well-known techniques. It is

unknown whether the problem of colouring an arbitrary planar graph using at most 4 colours is in NC.

Bent Thomsen, Imperial College, London, *Calculus of Higher Order Communicating Systems.*

ABSTRACT: According to R. Milner's book: *A Calculus of Communicating Systems* from 1980, one of the original intentions of CCS was that it should serve as the λ-calculus of concurrent systems. Subsequent research shows that it serves well as such for a large range of applications. But, as already pointed out in R. Milner's book, it has limitations when one wants to describe unboundedly unstructured expanding systems as, e.g., an unbounded number of procedure invocations in an imperative concurrent programming language.

We believe that this deficiency comes from the first order nature of CCS and this has led us to consider higher order constructions such as sending and receiving processes. To elaborate on this deficiency we have devised a "minimal" extension of CCS, called CHOCS, to take processes as values into account, preserving as much of the original CCS as possible.

CCS was intended to describe concurrent systems both in software as well as in hardware. Current trends in hardware research, towards dynamic reconfigurable systems, show that our theoretical developments have strong possibilities of being implementable in the future, as well as being used in the specification of such systems.

As an example of a specification in CHOCS we define a fault tolerant editor. This example easy generalises to an operating systems setting.

Clearly CCS with processes as first class objects is a powerful metalanguage and we show that it is possible to simulate the untyped λ-calculus in CHOCS. The relationship between CHOCS and the untyped λ-calculus is further strengthened by a result showing that the recursion operator is unnecessary in the sense that recursion can be simulated by means of process passing and communication. We do therefore not need an explicit recursion operator to obtain infinite behaviours.

J. V. Tucker, University of Leeds, *Generalisations of the Church-Turing Thesis.*

ABSTRACT: The talk concerned the generalisation of computability theory to many-sorted algebras and classes of many-sorted algebras. Two motivations were discussed: the need for a theory of *synchronous concurrent computation on streams* (that applies to systolic computation and other clocked hardware computation); and the need for a theory of verifiable computation (that applies to proving program properties using data type specifications).

The functions defined by *simultaneous course-of-values recursion* and the *least number operator* were introduced and related to many other definitions. A *Church-Turing Thesis* concerning the limits of parallel deterministic computation on abstract data types was stated.