New Connections between Mathematics and Computer Science

Jeremy Gunawardena
BRIMS, Hewlett-Packard Labs
Filton Road, Stoke Gifford
Bristol BS12 6QZ, UK.
jhcg@hplb.hpl.hp.com

January 22, 1996

Abstract

A workshop on *"New Connections between Mathematics and Computer Science"* was held at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England from 20-24 November 1995. The workshop was supported by the Engineering and Physical Science Research Council of the United Kingdom, the London Mathematical Society and Hewlett-Packard's Basic Research Institute in the Mathematical Sciences. This document contains a report on the workshop, the abstracts of the talks and the accompanying bibliography.

1 Report on the workshop

The interplay between mathematics and computer science has traditionally centered around areas in logic, category theory and discrete mathematics. In recent years new connections between mathematics and computer science have emerged from such unexpected quarters as algebraic topology, differential geometry, dynamical systems and operator algebras. These new developments hold the promise of bringing new insights and powerful mathematical tools to bear on problems in computing. At the same time, such problems have opened new avenues of exploration for the mathematician.

To investigate these developments and to introduce them to a wider audience, a workshop on *"New Connections between Mathematics and Computer Science"* was held at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England from 20-24 November 1995. It was timed to take advantage of parallel programmes in "Semantics of Computation" and "From Finite to Infinite Dimensional Dynamical Systems" at the Issac Newton Institute in the second half of 1995. The workshop was hosted jointly by the two programmes and by Hewlett-Packard's Basic Research Institute in the Mathematical Sciences (BRIMS) in Bristol, England. The workshop was organised by Jeremy Gunawardena of BRIMS, who was also a participant at the "Semantics of Concurrency" programme.

Each day of the workshop was devoted to a single theme. The five themes were, in chronological order:

- Differential geometry of algorithms,
- Developments in complexity theory,
- Algebraic topology and distributed computation,
- Dynamics of proof and computation,
- Geometry of images and computer vision.

The choice of these particular themes was influenced in part by the two programmes at the Newton Institute and in part by the prejudices of the organiser. The "Connections between Mathematics and Computer Science" cover a broad spectrum of areas and a number of workshops could have been held under the same title with little overlap among the participants. If there is a coherent thread among the various themes it is that of dynamics and geometry.

Four tutorial lectures of one hour each were given in the morning sessions by invited speakers and a small number of half hour contributed talks were given in the late afternoon. The programme was arranged to allow maximum time for discussion among the participants.

Particular emphasis was placed on attracting students and, due to generous financial support (see below), it was possible to fund 7 United Kingdom research students to attend the workshop. There were 62 registered participants, excluding some unregistered participants who were already present at the Isaac Newton Institute. By country of affiliation, these broke down as follows: 28 (UK), 10 (USA), 7 (France), 3 (Denmark), 2 (Canada), 2 (Germany), 2 (Holland), 2 (Italy), 1 (Belgium), 1 (Hong Kong), 1 (Japan), 1 (Mexico), 1 (Poland), 1 (Russia).

This workshop bought together a very disparate group of people who were not only unfamiliar with each other's work but often operating on the basis of quite different philosophies. In retrospect, and for the benefit of future workshops of a similar nature, it would have been helpful to have devoted some time to basic introductory lectures on dynamical systems and topology (from the mathematical side) and complexity theory and semantics (from the computer science side). Nevertheless, judging from the comments of a number of people, the workshop was stimulating and suggested the emergence of new and fruitful areas of study for both the mathematician and the computer scientist.

A independent review of the workshop by John Baez, one of the participants, appears in a weekly world wide web column on *"This week's finds in mathematical physics"* (accessible at URL:http://math.ucr.edu/home/baez/week70.html) and gives some idea of the scope of material that was presented.

Cambridge University Press have expressed interest in publishing a collection of articles based on the workshop. Further information regarding this may be obtained from Jeremy Gunawardena.

The next section of this report contains the abstracts of each talk (both invited and contributed) that was presented. The final section contains a bibliography of relevant papers refered to in the abstracts.

The workshop was financially supported by the Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom, the London Mathematical Society (LMS) and BRIMS. The LMS funding was obtained under the MathFit initiative—Mathematics for Information Technology—for which this was the introductory workshop. MathFit is currently being considered by EPSRC as a managed programme. Further information on the intitiative is available from Professor Ursula Martin, at the Department of Computer Science at the University of St-Andrews, or on the world wide web at URL:http://www.qmw.ac.uk/~lms/mathfit.html.

Further information about BRIMS, including a copy of this report, is available on the world wide web at URL:http://www-uk.hpl.hp.com/brims/.

2 Abstracts of talks

The combinatorics of abstract data types

Mike Atkinson, Department of Computer Science, University of St-Andrews
mda@dcs.st-and.ac.uk

An abstract data type may be regarded as a machine whose instruction set is the set of allowed operations which characterise the data type. The theory of machines has traditionally been concerned with the relationship between input sequences to the machine and output sequences. We shall discuss the combinatorial aspects of this relationship in the case of various data types regarded as machines. For references, see [2, 3, 58].

n-Categories in logic, topology and physics

John Baez, Department of Mathematics, University of California at Riverside
baez@ucrmath.ucr.edu

Loosely speaking, an 'n-category' is a structure with objects, morphisms between objects, morphisms between morphisms or '2-morphisms', and so on up to n-morphisms. While only well-understood for low n at present, n-categories appear to provide a framework for understanding 'processes between processes', thus making precise certain analogies between logic, topology, and physics. In categorical logic, the process of 'weakening' consists of replacing equational laws by specified natural isomorphisms, which then must satisfy certain equations of their own to be manipulated with some of the same facility as the original equations. (For example, the product of real numbers is strictly associative, while the tensor product of vector spaces is only associative up to natural isomorphism.) To iterate this process we need some notion of n-category. On the other hand, homotopy theory may be regarded as the study of the n-category associated to a space whose objects are points of the space, whose morphisms are paths, whose 2-morphisms are paths between paths, and so on. The n-categorical coherence laws arise naturally from the topology in this context. Similarly, in topological quantum field theory, n-dimensional cobordisms may be represented as n-morphisms, and important equations such as the Yang-Baxter and Zamolodchikov equations can also be seen as n-categorical coherence laws. Here we sketch some of the main ideas involved, illustrating them with many examples. For references, see [5, 6].

Complexity of trajectories in rectangular billiards

Yuliy Baryshnikov, School of Mathematics, University of Hull
y.baryshnikov@maths.hull.ac.uk

Take a cube as a billiard domain and a generic trajectory in it. Apparently, nothing more simple can be imagined in billiards - but even here some complexities hide. Write down the number i each time the trajectory hits a wall parellel to i-th coordinate hyperplane and count the number of different words of length n in the resulting infinite word. This number as function of n is called the complexity of the trajectory. A classical result by Morse and Hedlund, [55], is that the complexity in dimension 2 does not depend on trajectory (and equals to $n+1$). This was generalized recently, [1], to three dimensions. I prove the independence in all dimensions and find the function, [7]. The infinite words arise actually in great many areas like computer sciences (digitalized straight lines, finite languages), game theory and quasicrystalls. The proof hints at some possible underlying structures like combinatorial theory of infinite-dimensional polytopes or multidimensional generalisations of chain fractions.

Discrete computations and Hamiltonian and gradient flows

Anthony M. Bloch, Department of Mathematics, University of Michigan
abloch@math.lsa.umich.edu

In this talk I will discuss some of the connections between the theory of gradient flows and integrable Hamiltonian systems and various discrete computations. In particular I will discuss how problems of least squares fitting, the sorting of numbers and linear programming can be couched as discrete versions of smooth Hamiltonian and/or gradient flows.

I will explain a natural dualism that arises between gradient flows and integrable or solvable systems and the importance of the geometry of the moment map. The latter gives rise to a convex polytope which plays a crucial role in the computations and especially in understanding linear programming by interior point methods. I will discuss some of the theory of integrable systems with particular reference to the Toda lattice system which plays a key role in this work. I will also discuss some extensions of these ideas to an infinite dimensional setting. The talk describes mainly joint work with R. Brockett, H. Flaschka and T. Ratiu. See, for instance, [9, 10, 11].

From curve complexity to perceptual grouping

Benoit Dubuc and Steven W. Zucker, Center for Intelligent Machines, McGill University
benoit@athena.mcrcim.mcgill.edu

A classical problem in the design of a general purpose artificial vision system is the localization of image curves (edges or bars), commonly referred to as “edge detection”. Edge detection leads to a basic issue in perceptual grouping: once the local structure is established, the transition to global structure must be effected. To illustrate, consider an edge element inferred from an unknown image. Is it part of a curve, or perhaps part of a texture? If the former, which is the next element along the curve. If the pattern is a texture, is it a flow pattern, in which nearby elements are oriented similarly, or a turbulence pattern, in which they are not. These questions of perceptual grouping are linked to the relations existing between the different types of measures on curves. These might include the length, the area covered, or the number of components, and it is the relationship between these that is important both theoretically and practically.

In this presentation we propose a unifying theory of early visual structure from the more abstract perspective of geometric measure theory. The theory is built upon the notion of a tangent map, as this derives from edge detection, and results in a measure of structural complexity that requires separate normal and tangential components. Thus the descriptive repertoire is defined not just by the tangents, or the salient length property of curves, but by the full spatial context in which tangents are distributed. Natural extrema in the complexity map define semantically-meaningful structural categories, including dust, curves, turbulence, and flows. These categories capture the first stages of segmentation; but via complexity analysis not pixel grouping.

For a reference, see [18].

Dynamical systems, measures and fractals via domain theory

Abbas Edalat, Department of Computing, Imperial College London
ae@doc.ic.ac.uk

Domain theory investigates properties of “continuous posets”. It was introduced by Dana Scott in 1970 as a mathematical theory of computation in the semantics of programming languages, and since then, it has developed extensively in various areas of semantics.

In recent years, a new direction for applications of domain theory has emerged as it was uncovered that there are indeed natural domain-theoretic computational structures in dynamical systems, measure theory and fractals. In particular, a computational framework for measure theory was established, which then led to a generalisation of the Riemann theory of integration with diverse applications.

We will give a survey of this work and describe a number of results and algorithms in the periodic doubling route to chaos and in the theory of iterated function systems, with applications in the one-dimensional random field Ising model, forgetful neural nets and fractal image compression. We also explain the basics of a programming language with a real number data type for exact real number computation which includes computing integrals.

For references, see, [19, 20, 21].

Computer science contradicts mathematics

Peter Freyd, Department of Mathematics, University of Pennsylvania
pjf@saul.cis.upenn.edu

The first task in the semantics of programming is to make sense out of what it is that the designers of programming languages do. The situation is reminiscent of the interplay between quantum physics and mathematics: the applied researchers make progress using what appears to be mathematically unsound; the pure researchers are challenged to find mathematically sound settings to account for the progress of the applications; the pure and the applied proceed to transform each other—and with luck—transform each other to each other’s great benefit.

In a programming language that allows user-defined data-types one may interpret the rules of the language as the description a category, which category often looks quite impossible from a traditional mathematical perspective. Indeed, the “completeness” conditions in categories that arise, on one hand, from mathematics and on the other hand, from computer science are usually quite contradictory.

Many programming languages describe what have been called "algebraically complete"—and even more stringent—"algebraically compact" categories. There is, up to equivalence, only one example of the latter that is also a complete category in the usual mathematical sense, to wit, the category with just one morphism. The motivating property of algebraically compact categories in computer science is that for any endofunctor, T, (read "data-type constructor") on any number of covariant and any number of contravariant variables there is a "fixed-point", that is, an object, A, such that $T(A, A, \cdots, A)$ is canonically isomorphic to A. Moreover, among all such fixed-points there is one that appears as a retract of all the others. It surprised the writer that there are algebraically compact categories that arise in pure mathematics, the most easily described being the category whose objects are separable Hilbert spaces and whose maps are linear maps of norm at most one.

Recent work of Izumaya and Sano on affine differential geometry in the plane

Peter Giblin, Department of Mathematics, University of Liverpool
su15@liverpool.ac.uk

The Euclidean differential geometry of plane curves can be studied by means of functions called height functions and distance squared functions. Recently Izumaya and Sano have suggested a way of extending this method to affine differential geometry.

Geometry of proofs

Jean-Yves Girard, University of Marseille
girard@lmd.univ-mrs.fr

Linear logic reintroduced a *classical* symmetry in the constructive universe that was absent from intuitionistic logic. The symmetrization of natural deduction induced *proof-nets*, a kind of graph-like semantics, for which a geometrical correctness criterion was found.

From this it turns out that the syntactical manipulations coming from proof-theory are nothing but the symbolic solution of a geometrical problem: the inversion of an operator in a C^*-algebra.

The forthcoming book (Cambridge, May or June) "Advances in Linear Logic" contains several survey papers, on linear logic, geometry of interaction and proof-nets.

Scheduling problems and homotopy theory

Eric Goubault, ENS Paris
eric.goubault@ens.fr

I will introduce some typical scheduling problems that arise in many areas of computer science. For instance, the correctness of concurrent databases, or the robustness to failures of processors in distributed systems, [42], can all be expressed as scheduling properties of the actions of the different processes involved. Verifying that a parallel program can be implemented on a given architecture (shared resources or channels of communication) also relies on specific and sometimes intricate scheduling properties. I will develop a semantic framework based on Higher-Dimensional Automata (HDA), [27], for describing such concurrent and distributed systems. Then I will show that schedulers can be described, using a homotopy theory of these HDA, as deformation retracts, [26]. This homotopy theory is slightly different from the usual homotopy theory for say, CW-complexes, in that we are not allowed to reverse time. But still we can define homotopy groups for which Hurewicz-like and Seifer/Van Kampen-like properties hold. Connections with word problems, [45, 29, 70], and parallelization of programs will also be sketched. See also [33, 39].

Complexity in groups

Mikhail Gromov, IHES Paris
gromov@ihes.fr

Many standard complexity problem can be adequately translated to the group theoretic language. The group theoretic language is easy to make geometric which suggest a similar geometrization of more traditional problems in the complexity theory. See [56], particularly the Introduction, for some related observations.

Digital circuits and nonexpansive maps

Jeremy Gunawardena, BRIMS, HP Labs Bristol
jhcg@hplb.hpl.hp.com

In this talk I will discuss certain discrete dynamical systems, $F : R^n \to R^n$, in which the maps, F, are nonexpansive in the supremum norm, [34]. Such systems arise, for instance, in modelling the timing behaviour of certain digital circuits, [30]. The geometry of R^n with the supremum norm constrains the dynamics in ways that are still quite mysterious. Of particular interest for the applications is the asymptotic average, or cycle time vector, $\lim_{k\to\infty} F^k(x)/k$ which, when it exists, appears to give information on the existence of fixed points of F, [31]. I will mention some new results that shed light on this asymptotic average, [36]. I will then describe the main conjecture regarding the cycle time vector and explain some of the special cases in which it has been proved, [31, 32]. This work suggests that some aspects of the classical Perron-Frobenius theory can be extended to nonlinear operators.

Part of this is joint work with Mike Keane and Colin Sparrow.

Theoretical aspects of morphological image processing

Henk Heijmans, CWI, Amsterdam
henkh@cwi.nl

Mathematical morphology is a quantitative approach in image analysis which, on the practical side, has been applied successfully in various disciplines such as mineralogy, medical diagnostics and histology, and, on the theoretical side, has been provided with a solid mathematical basis leaning on concepts from algebra, topology, integral geometry and stochastic geometry.

The central idea of mathematical morphology is to examine the geometrical structure of an image by matching it with small patterns at various locations in the image. By varying the size and the shape of the matching patterns, called structuring elements, one can extract useful information about the shape of the different parts of the image and their interrelations. In general the procedure results in nonlinear image operators which are well-suited for the analysis of the geometrical and topological structure of an image.

Originally, mathematical morphology has been developed for binary images which can be represented mathematically as sets. The corresponding morphological operators use essentially three ingredients from set theory, namely set intersection, union, complementation and also translation. But the urgent need for a more general theory including other object spaces has finally led to the observation that a general theory on morphological image analysis should be based upon the assumption that the underlying image space is a complete lattice.

In our talk we will try to give the audience a basic understanding of the mathematics behind morphology. For references, see [4, 37, 38].

Quantum computation and Shor's factoring algorithm

Richard Jozsa, University of Plymouth
p08205@prime-a.plymouth.ac.uk

Quantum phenomena lead to qualitatively new modes of computation which are not in the repertoire of any classical computing device such as a general purpose digital computer. A quantum computer cannot compute any Turing-non-computable function but it may be able to perform computations more efficiently than any classical device. Indeed Peter Shor has shown how integers can be *efficiently* factorised on a quantum computer—a problem for which there is no known efficient (i.e. polynomial-time) classical (even randomised) algorithm.

In this talk we describe the essential principles of quantum computation and discuss Shor's quantum factoring algorithm, including a brief review of the (elementary) quantum theory required.

For references, see, [8, 17, 22, 24, 65].

Homological methods and word problems

Yves Lafont, University of Marseille
lafont@lmd.univ-mrs.fr

Complete (or convergent) rewrite systems have been introduced as a tool for solving word problems. Anick and Squier discovered that they can also be used for calculating homological invariants. This allows for instance to construct counterexamples in the theory of complete rewrite systems. I will present some of the main ideas and results in this new area.

Distributed computation and the twisted isomorphism

Michael Manthey, Department of Mathematics and Computer Science, Aalborg University
manthey@iesd.auc.dk

This work is the result of pursuing the idea of representing all aspects of computation in terms of patterns of synchronization. Computationally, the context is that of truly distributed, self- organizing, growing systems embedded in an environmental surround, where the system/environment boundary is constituted by primitive binary sensors and effectors. Mathematically, the context is discrete and finite (but unbounded).

The starting point is to view the succession of individual discrete sensor states (= primitive process states) as distinctions - that is, exclusions - in the time domain. The contrary distinction to exclusion is then 'co-occurrence', which, following Leibniz, is a space-like distinction. It then follows that 'action' can be expressed as co-occurrences that exclude each other, which we call 'co-exclusion'. Co-exclusion expresses the computational concept called 'mutual exclusion' between processes, and the two constituent co-occurrences of an action defined by co-exclusion designate that action's pre- and post-conditions, although which is 'pre' which is 'post' depends on the current state of the environment. We have im-plemented a system called Topsy on this basis, and are currently carrying out various experiments, [23]; [71] is interesting in this connection.

Taking binary sensors over the field $(-1, 0, 1)$ and imposing a vector space structure on the set of sensors, it can be shown that they satisfy the requirement of a Clifford algebra, whose $+$ is then in-terpreted as 'co-occurrence' and whose product as mutual exclusion or action. Hence, for sensors A and B, the expression $A + -A = 0$ means that a given value of sensor A cannot co-occur with its opposite value, and $AB(A + B)BA = -A + -B$ expresses the effect of carrying out action AB on the state $A + B$, [44, 48, 49].

Since products such as AB are themselves oriented, they can be viewed as meta-sensors, and co-exclusion can then be applied recursively to them, resulting in a hierarchical structure, in fact, a cohomology. The goal-oriented run-time structure of Topsy can for its part be interpreted as an homology. The relationship between these can in turn be viewed as an instance of the "twisted isomorphism" between homology and cohomology, yielding a rich and detailed model of distributed computation as well as a wealth of powerful and suggestive interpretations.

Termination and invariants

Ursula Martin, Department of Computer Science, University of St Andrews
um@dcs.st-and.ac.uk

Many different techniques have been developed for proving the termination of programs or functions. Some are purely ad-hoc arguments developed solely to solve an immediate problem. However a typical termination proof involves finding a quantity which decreases at each step of a computation, or in other words finding a well-founded ordering which is respected by the process we are considering. An extraordinary diversity of well-founded division orderings on structures such as vectors, strings, partitions or terms have been devised to prove termination. Such orderings can also be used to constrain the search space in the search for solutions or normal forms.

Recently Martin, Scott and others, [50, 52] have developed a classification of such orderings in terms of a family of numerical invariants which characterise an ordering by a sequence of real functions. The characterisation can be refined by counting occurrences of certain substructures, essentially using the theory of Lie elements in a free algebra.

This classification has practical implications, since we may attempt to decide if a certain system is terminating in such an ordering by solving certain linear or polynomial constraints. The ordering associates numeric and logical invariants to the underlying algorithms and data structures. The link between these invariants, the mathematical structure of the search space, and the underlying algorithms is as yet largely unexplored. See also, [16, 51].

(Editor's note: Professor Martin's talk was changed at the last minute. She actually spoke on *"The princess and the plumber: the role of mathematics in computer science"*. No abstract is available for this.)

Dynamics of computational processes

Gianfranco Mascari, University of Rome
mascari@vaxiac.iac.rm.cnr.it

A challenging problem in modelling computational processes is to take into account the ”modularity of parallelism”. We present an approach to modularity of parallelism in which the space of computations has a topological nature:

- modularity of computations is analysed by the methods of proof theory,
- parallelism of computations is analysed by the methods of quantum computation.

The role of recent work in low dimensional topology and operators algebras is essential in studying such spaces of computations.

For references, see [53, 54].

Differential invariants in computer vision

Peter Olver, Department of Mathematics, University of Minnesota
olver@ima.umn.edu

Recent advances in computer vision, based on invariant geometric diffusion processes including Euclidean and affine invariant curve shortening, have underscored the importance of the classical theory of symmetry groups in the p.d.e. approach to image processing. In this talk, I will survey the basic theory of differential invariants, and show how they are practically employed to construct invariant differential equations. Recent work applying the theory of joint invariants to construct invariant finite difference numerical approximations to differential invariants will be outlined. Finally, I will discuss how such invariant diffusion equations are being used for multi-scale resolution, denoising, edge detection, and segmentation of real images, with particular emphasis on medical image processing.

Proof nets as formal Feynman diagrams

Prakash Panangaden, Department of Computer Science, McGill University
prakash@trichur.cs.mcgill.ca

Proof Nets, introduced by Girard when he invented Linear Logic, have a subtle combinatorial theory. The introduction of linear logic has revolutionized many semantical investigations. For example, the search for fully-abstract models of PCF and the analysis of optimal reduction strategies have been guided by linear logic. In my talk I will show how proof nets can be interpreted as operators in a simple calculus. This calculus was inspired by Feynman diagrams in quantum field theory. The ingredients are formal integrals, formal power series, a few special constants, addition and multiplication. A derivative operator, behaving like the familiar variational derivative, is the basic operator acting on such expressions. The calculus of these operators captures the combinatorics of proof nets. Most of the manipulations closely resemble what one does in a beginning calculus course. In particular the "box" construct is just exponentiation and the nesting of boxes phenomenon is the analogue of an obvious differential calculus formula. The theorem that we ahve established is that we have a linear realizability algebra (LRA). In origin and form this LRA is strikingly different from any other in the extant literature.

This is joint work with Richard Blute of the University of Ottawa. See also [13].

Computational and dynamical interpretations of mathematical structures

Vaughan Pratt, Department of Computer Science, Stanford University
pratt@cs.stanford.edu

A Chu space (A, X) over a set K is a general-purpose concrete object having a dual object (X, A) and admitting both mathematical and computational interpretations.

Organizing K as a quantale equips (A, X) with the structure of a metric space whose points represent the events of a schedule and whose distances represent sharp or fuzzy delays between events. Dually (X, A) becomes a metric space whose points represent the states of an automaton and whose distances represent sharp or fuzzy correlations between states. Different quantale structures for K can lead to strikingly different interpretations of (A, X) as a metric space.

We apply chain-folding to enumerate the 2^{n-2} chain quantales of n elements. The three nontrivial such of cardinality up to three find valuable employment in the process interpretation of Chu spaces, for respectively untimed automata making discrete transitions, higher dimensional automata making continuous transitions, and causal automata making circumspective transitions. For references, see, [59, 60, 61].

On the decidability of a distributed decision task

Sergio Rajsbaum, UNAM, Mexico
rajsbaum@concha.matem.unam.mx

A *task* is a distributed coordination problem in which each process starts with a private input value taken from some finite set, communicates with the other processes by applying operations to shared objects and eventually halts with a private output value, also taken from a finite set. A *protocol* is a program that solves a task. We consdier protocols in which processes communicate by shared read/write memory possibly augmented by other synchronization primitives. A protocol is *t-resilient* if it tolerates failures by t or fewer primitives.

We say that tasks are *decidable* in a given model of computation if therte exists an effective procedure for deciding whether a task has a t-resilient protocol in the model. We show that in most models, tasks are undecidable by a reduction to the contractibility problem of topology. In the models where tasks are decidable we present algorithms.

This is joint work with Maurice Herlihy. For references, see [39, 40, 41, 42]

On the writing of numbers

Jacques Sakarovitch, Institut Blaise Pascal, Paris
jacques.sakarovitch@ibp.fr

Numbers do exist by themselves, in abstracto. Some have even been given names, as 0, 1, π, e, etc. However, on many occasions, particularly in everyday life and in computer science, they are dealt with by means of writing them, using a representation in some numeration system or other. They are thus turned into sequences (finite or infinite) of letters, usually called digits.

We shall survey two main questions related to the treatment of such sequences by finite automata.

The first one is the characterization, due to A. Cobham, of those sets of numbers that are definable by finite automata in two different bases. If the statement goes back to the early seventies, its proof has been reworked, clarified and simplified quite recently, by means of different techniquess: the use of logic and a generalization to higher dimensions.

The second one is the treatment of representations of numbers in non classical systems (such as the Fibonacci numeration system or the golden mean base system). In such systems the writing of every number is not necessarily unique. The characterization of those systems for which the normalization, ie: the transformation of any representation of a number into a suitably prescribed one, may be realized by a finite automaton will bring to light a close relationship between Pisot numbers and finite automata (and sofic dynamical systems, of course). For references, see, [14, 25].

Geometric flows: theory and applications in image analysis

Guillermo Sapiro, HP Labs Palo Alto
guille@hpl.hp.com

The explicit use of partial differential equations (PDE's) in image processing and computer vision became an important research topic in the last years. A number of results related to the theory of geometric PDE's and their applications in image processing and computer vision are presented in this talk.

The topics we will cover in this tutorial presentation will be:

a) Classical mathematical morphology - We show how classical operations of mathematical morphology can be defined via geometric PDE's, based on the theory of curve evolution.

b) Affine invariant scale-space - We briefly describe how to extend classical results on Euclidean curve evolution to the affine case. Extensions to other groups will be given in other lectures at the workshop.

c) Geometric segmentation - We show the relation between energy snakes and geodesic computations for image segmentation. We prove that the solution to the active contours problem is given by a geodesic curve in a Riemannian space. A novel object detection algorithm is then proposed, based on a geometric PDE computing the geodesic curve. Results for 2D, 3D, color, and texture images are shown.

d) Histogram modification. We show how to improve the image contrast based on PDE's.

e) Anisotropic diffusion of color images. We present a PDE to perform anisotropic diffusion in general Riemannian spaces. Examples for color data are provided. For references, see [15, 57, 63].

Bezout's Theorem and Complexity

Mike Shub, IBM Yorktown Heights
shub@watson.ibm.com

We discuss the problem of locating the zeros of systems of algebraic equations from the point of view of complexity theory. The main theorem asserts that with respect to a reasonable measure on the space of n polynomial equations of degrees $d_1, \cdots, d_n$ in n unknowns, an approximate zero may be found with the number of arithmetic operations a low degree polynomial in the input size. This is joint work with Steve Smale. For a reference see, [67].

Invariance, termination, and time refinement in structured dynamical systems

M. Sintzoff, Department of Computing Science and Engineering, University of Louvain
ms@info.ucl.ac.be

Non-deterministic iterative programs can be viewed as discrete-time dynamical systems where transitions are relational, viz. multivalued. The analysis of programs by set functions (viz. predicate transformers) entails a set-based viewpoint of the dynamics, instead of the more usual point-based viewpoint. This facilitates the qualitative characterization of control modes, [46, 62], the definition of invariance, attraction and topological transitivity in the case of relational discrete-time dynamical systems, [68], and the compositional analysis of complex dynamics from simple ones, [69].

Here, we pursue this approach in order to tackle continuous-time systems together with discrete-time ones, and to ensure termination (viz. finite-time attraction) in addition to invariance and attraction. In hybrid systems, which also support the composition of continuous state-transitions and discrete ones, the latter ones are always assumed to be instantaneous. Here, we view discrete state-changes as abstractions of continuous ones: indeed, a discrete approximation, as any approximation, amounts to an abstraction. In this light, discrete-time phases are assumed to take non-zero time; this allows to keep a uniform underlying model of the dynamics. Discrete vector fields are so to speak plunged into continuous ones; this is achieved by a time refinement which yields a linear interpolation of discrete flows into continuous flows. The common underlying model is that of piecewise-smooth flows. Since the discrete steps can be refined into continuous ones, continuous and discrete phases in dynamics are interchangeable.

The verification laws for the invariance and the termination of continuous-time systems and discrete-time ones can then be restated in quite similar ways. Invariance is based on a local invariance defined in terms of continuous- or discrete-time transitions. Termination, i.e. finite-time attraction, is ensured by Lyapunov stability functions with a negative slope; in the case of discrete time, these functions amount to Floyd termination functions.

In this context, it is straightforward to compose dynamical systems using both kinds of time. This reminds of the freedom to compose program components using different data types. As a first step in this direction, we here consider classical means such as sequential composition and iteration; flows are concatenated as in the case of motion composition.

The proposed framework leaves the structure of reasoning invariant under change of time types. In consequence, the composition and design methods known for continuous-time systems can be reused for discrete-time ones, and conversely.

Filter automata and their particles

Pawel Siwak, Poznan University of Technology
psiw@agat.sk-kari.poz.edu.pl

This lecture is focussed on my idea of filter automata (FA) and on their particles. The filter automaton, F, is a sequential transducer. We use it in iterative processing of sequences of symbols. This computation may be equivalent to a pipeline of copies of F, or to a one-way linear cellular automaton (OLCA). The sites in such an OLCA follow rather Mealy model properties which is in contrast to Moore type model typically employed as cells. Such an approach is crucial to support moving periodic substrings, known widely as particles. These particles seem to mimic the phenomenon of solitary waves, or solitons. On the space time diagrams of the automata, one can observe a variety of soliton behaviours: higher order solitons, soliton collisions with non-demolition interaction, mutual interaction and jumps. We also observed annihilating solitons, fusion of solitons, decay of solitons and other fascinating phenomena. So far as I know, not all of these have been observed experimentally. The lecture will present a sketch of the theory of filter automata.

Blum-Shub-Smale theory

Steve Smale, City University Hong Kong
masmale@sobolev.cityu.edu.hk

We will transform the big problem of computer science “does P=NP?” into an algebraic setting and then give some recent related mathematical developments. For references, see [12, 66].

Factorization of linear ordinary differential operators: old (mathematical) results and their modern computer echo

Sergei Tsarev, Krasnoyarsk State Pedagogical University
tsarev@class.mian.su

This is in fact an exposition of some results about effectivization of the factorization technique for solution of linear ODE which was paid so much attention in the last decade on the basis of results obtained more than 90 years ago. See, for instance, [28, 47, 64].

References

[1] P. Arnoux, C. Mauduit, I. Shiokawa, and J. Tamura. Complexity of sequences defined by billiards in the cube. *Bull. Soc. Math. France*, 122:1–12, 1994.

[2] M. D. Atkinson. Transforming binary sequences with priority queues. *Order*, 10:31–36, 1993.

[3] M. D. Atkinson and R. Beals. Priority queues and permutations. *SIAM Jour. Computing*, 23:1225–1230, 1994.

[4] A. J. Baddeley and H. J. A. M. Heijmans. Incidence and lattice calculus with applications to stochastic geometry and image analysis. *Applicable Algebra in Engineering, Communication and Computing*, 6:129–146, 1995.

[5] J. Baez and J. Dolan. Higher-dimensional algebra and topological quantum field theory. To appear in Jour. Math. Phys. November 1995.

[6] J. Baez and M. Neuchl. Higher-dimensional algebra I: braided monoidal 2-categories. Available as ftp://math.ucr.edu/pub/baez/bm2cat.ps.Z.

[7] Yu. Baryshnikov. Complexity of trajectories in rectangular billiards. *Comm. Math. Phys.*, 1995.

[8] E. Bernstein and U. Vazirani. Quantum complexity theory. In *Proc. STOC*, pages 11–20. ACM Press, 1993.

[9] A. M. Bloch. Steepest descent, linear programming and Hamiltonian flows. *Contemporary Mathematics*, 114:77–88, 1990.

[10] A. M. Bloch, R. W. Brockett, and T. S. Ratiu. Completely integrable gradient flows. *Communications in Mathematical Physics*, 147:57–74, 1992.

[11] A. M. Bloch, H. Flaschka, and T. S. Ratiu. A Schur-Horn-Kostant convexity theorem for the diffeomorphism group of the annulus. *Inventiones Mathematicae*, 113:511–529, 1992.

[12] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bull. Amer. Math. Soc*, 21:1–46, 1989.

[13] R. Blute, P. Panangaden, and R. Seely. Old foundations for linear logic. In *Proc. MFPS.* Springer LNCS 802, 1993.

[14] V. Bruyère, G. Hansel, Ch. Michaux, and R. Villemaire. Logic and p-recognizable sets of integers. *Bull. Belg. Math. Soc.*, pages 191–238, 1994.

[15] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *Proc. Int. Conf. Comp. Vision*, Cambridge, Mass., 1995.

[16] E. A. Cichon. Bounds on derivation lengths from termination proofs. International Journal of Foundations of Computer Science, to appear.

[17] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc. Roy. Soc. London*, 439:553–558, 1992.

[18] B. Dubuc and S. W. Zucker. Indexing visual represenations through the complexity map. Technical Report TR-CIM-95-02, Centre for Intelligent Machines, McGill University, 1995.

[19] A. Edalat. Domain theory and integration. *Theoretical Computer Science*, to appear.

[20] A. Edalat. Dynamical systems, fractals and measures via domain theory. *Information and Computation*, to appear.

[21] A. Edalat. Power domains and iterated function systems. *Information and Computation*, submitted.

[22] A. Ekert and R. Jozsa. Shor's quantum algorithm for factorising numbers. *Reviews of Modern Physics*, to appear 1995.

[23] M. Manthey *et al.* A simple vision experiment using Topsy. Technical Report R-95-2016, Aalborg University, Department of Mathematical and Computer Science, 1993.

[24] R. Feynman. Simulating physics with computers. *Int. Jour. Theor. Phys.*, 21:467–488, 1982.

[25] Ch. Frougny. Representation of numbers and finite automata. *Math. Systems Theory*, 25:37–60, 1992.

[26] E. Goubault. Schedulers as abstract interpretations of HDA. In *Proc. of PEPM'95*. ACM Press, 1995.

[27] E. Goubault. Higher-dimensional automata, Part I. Technical report, Ecole Normale Supérieure, to appear 1995.

[28] D. Yu. Grigoriev. Complexity of factoring and calculating the GCD of linear ordinary differential operators. *Jour. Symbolic Computation*, 10:7–37, 1990.

[29] J. R. J. Groves. Rewriting systems and homology of groups. In L. G. Kovacs, editor, *Groups—Canberra 1989*, pages 114–141. Springer LNM 1456, 1991.

[30] J. Gunawardena. Timing analysis of digital circuits and the theory of min-max functions. In *TAU'93, ACM International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, September 1993.

[31] J. Gunawardena. Cycle times and fixed points of min-max functions. In G. Cohen and J.-P. Quadrat, editors, *11th International Conference on Analysis and Optimization of Systems*, pages 266–272. Springer LNCIS 199, 1994.

[32] J. Gunawardena. A dynamic approach to timed behaviour. In B. Jonsson and J. Parrow, editors, *CONCUR'94: Concurrency Theory*, pages 178–193. Springer LNCS 836, 1994.

[33] J. Gunawardena. Homotopy and concurrency. *Bulletin of the EATCS*, 54:184–193, 1994.

[34] J. Gunawardena. Min-max functions. *Discrete Event Dynamic Systems*, 4:377–406, 1994.

[35] J. Gunawardena, editor. *Idempotency.* Proceedings of the Isaac Newton Institute. Cambridge University Press, 1996, to appear.

[36] J. Gunawardena and M. Keane. On the existence of the cycle time for some nonexpansive maps. Technical Report HPL-BRIMS-95-003, Hewlett-Packard Labs, 1995.

[37] H. J. A. M. Heijmans. *Morphological Image Operators.* Academic Press, 1994.

[38] H. J. A. M. Heijmans. Mathematical morphology: a modern approach in image processing based on algebra and geometry. *SIAM Review*, 37:1–36, 1995.

[39] M. Herlihy. A tutorial on algebraic topology and distributed computation. Technical report, UCLA, 1994.

[40] M. Herlihy and S. Rajsbaum. Set consensus using arbitrary objects. In *Proc. of PODC'94.* ACM Press, 1994.

[41] M. Herlihy and S. Rajsbaum. Algebraic spans. In *Proc. of PODC'95.* ACM Press, 1995.

[42] M. Herlihy and N. Shavit. A simple constructive computability theorem for wait-free computation. In *Proc. of STOC'94.* ACM Press, 1994.

[43] M. W. Hirsch, J. E. Marsden, and M. Shub, editors. *From Topology to Computation: Proceedings of the SMALEFEST.* Springer Verlag, 1993.

[44] L. H. Kaufmann. Space and time in computation, topology and discrete physics. In *2nd IEEE Workshop on Physics and Computation.* IEEE Press, 1994.

[45] Y. Kobayashi. Complete rewriting systems and homology of monoid algebras. *Journal of Pure and Applied Algebra*, 65:263–275, 1990.

[46] R. Kumar, V. Garg, and S.I. Marcus. Predicates and predicate transformers for supervisory control of discrete event dynamical systems. *IEEE Trans. Automatic Control*, 38:232–247, 1993.

[47] A. Loewy. Uber vollstandig reduzible lineare homogene differentialgleichungen. *Math. Annalen*, 62:89–117, 1906.

[48] M. Manthey. Synchronization - the mechanism of conservation laws. *Physics Essays*, 5, 1992.

[49] M. Manthey. Toward an information mechanics. In *2nd IEEE Workshop on Physics and Computation.* IEEE Press, 1994.

[50] U. Martin. A geometrical approach to multiset orderings. *Theoretical Computer Science*, 67:37–54, 1989.

[51] U. Martin. Linear interpretations by counting patterns. In *Proc. 5th Int. Conf. on Rewriting Techniques and Applications.* Springer LNCS 690, 1993.

[52] U. Martin and E. Scott. The order types of termination orderings on terms, strings and multisets. Journal of Symbolic Logic, to appear.

[53] G. F. Mascari. Dynamics of computational processes. In preparation.

[54] G. F. Mascari and M. Pedicini. Dynamics of algorithms in partially additive categories. Appears in [35].

[55] M. Morse and G. Hedlund. Symbolic dynamics II: Sturmian trajectories. *Amer. Jour. Math.*, 62, 1940.

[56] G. A. Niblo and M. A. Roller, editors. *Geometric group theory, Volume 2: asymptotic invariants of finite groups, M. Gromov.* Number 182 in London Mathematical Society Lecture Note Series. Cambridge University Press, 1993.

[57] P. Olver, G. Sapiro, and A. Tannenbaum. Differential invariant signatures and flows in computer vision: a symmetry group approach. In B. Romeny, editor, *Geometry Driven Diffusion in Computer Vision.* Kluwer, 1994.

[58] V. R. Pratt. Computing permutations with double-ended queues, parallel stacks and parallel queues. In *Proc. ACM Symp. Theory of Computing*, pages 268–277. ACM Press, 1973.

[59] V. R. Pratt. Time and information in sequential and concurrent computation. In *Proc. Theory and Practice of Parallel Programming*, Sendai, Japan, 1994.

[60] V. R. Pratt. Rational mechanics and natural mathematics. In *TAPSOFT'95*, pages 108–122. Springer LNCS 915, 1995.

[61] V. R. Pratt. The Stone gamut: a coordinatization of mathematics. In *Logic in Computer Science.* IEEE Computer Society, 1995.

[62] P. J. Ramadge and W.M. Wonham. Modular feedback logic for discrete-event systems. *SIAM Jour. Control and Optimization*, 25:1202–1218, 1987.

[63] G. Sapiro and A. Tannenbaum. On affine plane curve evolution. *Journal of Functional Analysis*, 119:79–120, 1994.

[64] F. Schwarz. A factorization algorithm for linear ordinary differential equations. In G. Gonnet, editor, *Proceedings of ISSAC'89*, pages 17–25. ACM Press, 1989.

[65] P. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. FOCS*, pages 116–123. IEEE Press, 1994.

[66] M. Shub. On the work of Steve Smale on the theory of computation. Appears in [43].

[67] M. Shub. Some remarks on Bezout's Theorem and complexity theory. Appears in [43].

[68] M. Sintzoff. Invariance and contraction by infinite iterations of relations. In J.P. Banâtre and D. Le Métayer, editors, *Research Directions in High-Level Parallel Programming Languages*, pages 349–373. Springer LNCS 574, 1992.

[69] M. Sintzoff and F. Geurts. Analysis of dynamical systems using predicate transformers: attraction and composition. In S. I. Andersson, editor, *Analysis of Dynamical and Cognitive Systems*, pages 227–260. Springer LNCS 888, 1995.

[70] C. C. Squier, F. Otto, and Y. Kobayashi. A finiteness condition for rewriting systems. *Theoretical Computer Science*, 131:271–294, 1994.

[71] G. Theraulax and E. Bonabeau. Coordination in distributed building. *Science*, 289:686–688, 1995.