conduit!

Volume 5, Number 1 **Department of Computer Science** **Spring, 1996**
Brown University

PARIS C. KANELLAKIS, 1953–1995

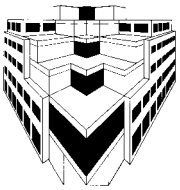
Paris Kanellakis, together with his wife Maria-Teresa Otoya and their two children, Alexandra and Stefanos, died on December 20 in the American Airlines crash outside Cali, Colombia. Paris's tragic death has created a void both at Brown and in computer science as a whole.

Paris was born in Athens, Greece on December 3, 1953; he received his undergraduate education at the National Technical University of Athens, where he was first in his class. He then went to MIT, where he received his Master's and Ph.D. degrees, and came to Brown as an assistant professor in 1981. He became a US citizen in 1988 and a full professor here in 1990.

Paris's research area was theoretical computer science. His contributions were unique both in the breadth of his interests and in his ability to carve out research programs in which his keen mathematical insight could be put at the service of practical issues. Broadly put, Paris was interested in how the formal language in which a problem is expressed affects the class of problems one can use it to attack. Most of us who have written programs feel intuitively that some problems are easier to express in one language than another. Paris worked at a more fundamental level: the languages he explored were deliberately kept simple (to make mathematical analysis possible) and the choice of language could decide not just ease of expression but whether or not a problem can be expressed at all. Furthermore, since the more expressive a language the wider the class of problems it can solve, it also follows that more expressive languages are less likely to admit efficiency—some of the programs expressed cannot be solved efficiently. Since this tradeoff is inevitable, one is always searching for languages that best balance these concerns.

Within this broad area Paris attacked a wide variety of issues. For example, computer databases require a language in which to express one's query. More recently the area of constraint programming languages attracted his attention. In a constraint language one says not merely that a particular variable is always an integer but that, say, it is an integer between certain values. Concerning efficiency, some of Paris's most important papers showed that language features previously thought unexceptionable—unification and type checking, to cite what are probably among his most important results—in fact contain pitfalls that require careful negotiation. Also, since in many cases one is interested in efficiency when using not just a single computer but rather a large collection of computers, Paris made fundamental contributions to the area of parallel processing. In all of these cases Paris worked closely with practitioners in the area here at Brown and elsewhere to ensure that his work was grounded in reality. For this and related work Paris was viewed as a leader in theoretical computer science, particularly among those with a taste for practice.

Paris was not only an intellectual leader in his field but a professional leader as well, through his willingness to organize conferences, mentor students, and generally work for the better-



ment of his intellectual community. Indeed, he was so much in demand that his CV already lists two conference committees for 1997. Paris graduated seven Ph.D. students, and the depth of the concern they are now expressing speaks volumes about the kind of advisor, and the kind of person, Paris was.

Paris Kanellakis was one of my 'guys,' as I call my professors. He was even more—he was also a good friend. He was truly a remarkable man. He could sometimes be very demanding, but in a kind way. He always took the time to ask me how things were going and how I was feeling and he would take time to listen. His work kept him very busy, especially trying to get government funding and grants. He loved his family, his job, his students and his community and he always took the time to talk to his students. He will always be remembered for being late for class—this was his trademark!

He would sometimes talk about his native Greece, his parents and his childhood, and his face would light up with one of his big beautiful smiles. His smile could light up a room. He was kind, considerate, charming, witty and had strong feelings for fair play and equality. I will never forget the last evening we worked together—he was very appreciative that I stayed late to help him. We were able to get all the work done and when we were finished I gave him a hug and wished him happy holidays. He wished me the same and told me to take care and phone him the following week. He said "Say a prayer that I get out of here OK." I replied " I'll say a prayer that you have a safe flight." As it turned out, God had other plans. I will always smile when I remember Paris. Although he is no longer with us, he will always remain in our hearts.

Mary A. Andrade, Sr. Academic Secretary

Paris put his great energy and commitment at the service of our Department and the University as well. He assumed many tasks for the Department and performed them with skill, devotion and good spirits. But occasionally this caused small problems. For many years now our Department has operated in two time zones, regular time and Kanellakis time, which uniformly ran about twelve minutes behind. Paris, it seemed, always wanted to get one more thing done before his next meeting. Thus it was standard practice at faculty meetings to schedule those items in which Paris had a large role later in the meeting. But he tended to be involved in everything, so often there was nothing to do at the beginning.

Paris combined common sense and high spirits with a deep commitment to honor and fairness. A year or two ago Brown issued some administrative guidelines that worried him on due-process grounds. He took up his pen against them; characteristically, he wrote forcefully but with good humor. To quote a few lines: "I have personal experience with non-freedom of speech and suppression of other rights. I was an undergraduate in Greece when the government was a military dictatorship and the police force was a genuine instrument of repression." But he went on to say: "Of course, any analogy is exaggerated. Rhode Island is not the Greece of 20 years ago; it is not even the Isle of Rhodes."

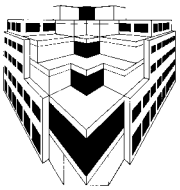
Paris had great insight into human nature and was fiercely honest. He was one of the people always consulted on tricky departmental issues because we respected his opinions and valued his insights. He also had a fine sense of humor, a wonderfully wholehearted laugh and an outgoing, energy-filled personality that drew everyone to him. He turned 42 just two weeks before his death. His accomplishments were immense even in the time he had, and we grieve for the loss of what he would have accomplished had he had more. We console ourselves with the years that he, and we, did have.

The Paris C. Kanellakis Memorial Fund has been established by the Department. Donations, payable to Brown University, may be sent to the Gift Cashier, Brown University, Box 1877, Providence, RI 02912. Please mention Paris's name on the memo line of your check.

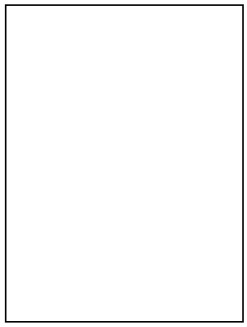
PARIS KANELLAKIS'S RESEARCH

Paris Kanellakis's research interests bridged theory and practice both within Brown's Department of Computer Science, where he managed departmental research projects linking the two, and in the wider international research community. Paris contributed original research in areas as diverse as databases, programming languages, distributed computing, fault tolerance, complexity theory, combinatorial optimization, and lambda calculus models. Underlying those contributions was a unifying theme: the use of logic, complexity, and algorithms to understand the foundations of practical systems, analyze their efficiency, and improve their functionality.

The March issue of *Computing Surveys*, to which Paris had planned to contribute an article on database theory, has been dedicated to him and contains instead an article describing his accomplishments by five of his recent close collaborators. Moshe Vardi describes Paris's work on deductive databases, Serge Abiteboul his work on object-oriented databases, Gabriel Kuper his work on constraint databases, Alex Shvartsman his work on fault-tolerant parallel



Pascal Van Hentenryck



Peter Wegner

computation, and Harry Mairson his work on complexity and type theory. In the rest of this article, we sketch some of these results (references can be found in the *Computing Surveys* article and in the Paris Kanellakis memorial page, accessible at http://www.cs.brown.edu).

The first issue of the *Journal of Logic Programming* featured an article by Dwork, Kanellakis, and Mitchell entitled "On the Sequential Nature of Unification." The paper shows that the decision problem "Do two terms unify?" is complete for PTIME; informally speaking, this means that unification cannot be speeded up with a polynomially bounded number of processors. Subsequently Kanellakis and Mitchell used the essential idea behind the proof to show that type inference in ML was PSPACE-hard, i.e., as hard as any problem that can be solved in polynomial space. This result contradicted the popular belief at the time that ML typing was algorithmically simple. His subsequent paper in collaboration with Mairson and Mitchell showed the problem to be complete for EXPTIME. Paris's most recent work on the lambda calculus (with Hillebrand and Mairson) led to a new and elegant syntactic characterization of complexity classes in terms of the type of the lambda-calculus program, a result that emerged from their research on a functional programming foundation for a logic-based database query language.

Paris was a major contributor to the theory of deductive databases. Using tools from complexity theory, he and Cosmadakis investigated which classes of Datalog queries could be speeded up by parallel computation. Together with Cosmadakis, Gaifman, Hillebrand, Mairson, and Vardi, he studied the decidability of boundedness problem for various classes of Datalog queries (a Datalog query is bounded if its database complexity is O(1)), showing, in particular, that the boundary between the decidable and the undecidable lies between unary and binary queries. He also studied efficient bottom-up implementation of Datalog in a paper with Beeri, Bancilhon, and Ramakrishnan.

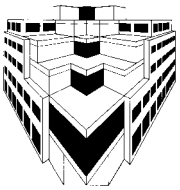
In 1989-89, Paris visited the Database Research Group at INRIA Rocquencourt and studied the foundations of object-oriented database systems, which were emerging during this period. His desire to understand the database system O2 led him to develop (in collaboration with Abiteboul, Bancilhon, Delobel, Hillebrand, Ramachandran, and Waller) an object-based data model, a new formalization of object identity, new programming tools, and new indexing algorithms. The book *The Story of O2*, edited jointly with Bancilhon and Delobel, remains a landmark study of the interconnection between theory and practice in the area of object-oriented databases.

Again in the area of databases, this time in collaboration with Kuper and Revesz, Paris developed the concept of constraint databases, in which the concept of tuples in the relational model is replaced by a conjunction of constraints. They investigated the query complexity of this scheme (which parallels in the database world the area of constraint logic programming) for various classes of constraints. Together with his colleagues and students, he was also engaged in long-term research on this topic. In particular, he and Dina Goldin were working on query algebras and indexing techniques for constraint databases to make this technology practical.

Paris was also interested in bridging the gap between abstract models of parallel computation and realizable architectures. Most parallel algorithms require a fault-free environment to perform correctly and efficiently. In collaboration with Shvartsman, and subsequently also with Buss, Michailidis, and Ragde, Paris proposed a formal notion of robustness that combines fault tolerance and efficiency and studied algorithms that remain efficient in the presence of arbitrary dynamic processor failure patterns. This research demonstrates how theoretical work on parallel algorithms, with speed-up close to linear in the number of processes, can be made practically relevant.

"Those of us who worked with Paris have lost not only an outstanding scientist but also an esteemed colleague and a dear friend"

Those of us who worked with Paris have lost not only an outstanding scientist but also an esteemed colleague and a dear friend. As a colleague, he had the poise, personality, and energy to rally communities behind him and he used these qualities to improve our academic and professional environment. We also mourn a friend with a charming and engaging personality and a Mediterranean passion; the warmth and hospitality of Paris and his family will be sorely missed.



To acknowledge Paris's contributions to computer science and the deep sense of loss felt by many of us, the Association for Computing Machinery (ACM) plans to institute an award in his memory. To recognize Paris's pragmatic approach to theoretical computer science, the award will be given for "a theoretical contribution with a significant impact on practice." The endowment for the award is provided by ACM's Special Interest Groups in Automata Theory and Databases (SIGACT and SIGMOD), from the Kanellakis family, and from individual contributions (see below).

ACM PARIS KANELLAKIS AWARD

The Association for Computing Machinery plans to institute in Paris's memory a "Paris Kanellakis Award," provided that sufficient endowment can be raised, to be given for "a theoretical contribution to computer science with a significant impact on practice." The winner will be selected by a committee appointed by the ACM Awards Committee; the award will be given at the annual ACM awards ceremony or as determined by the ACM awards committee. The award may recognize either a specific contribution or a body of work performed no more than fifteen years before the date on which the prize will be awarded.

Two of ACM's Special Interest Groups, SIGACT and SIGMOD, have contributed $10,000 each to this award and several other institutional contributions are probable. The Kanellakis family has also contributed most generously. We invite pledges to honor Paris and the tradition of scholarly excellence he embodied. The collection of the pledge will be handled by the ACM, who will be in touch with you concerning your (tax-deductible) contribution. Please indicate the amount you wish to pledge by sending email to maa@cs.brown.edu or writing to the following address: ACM Paris Kanellakis Award, c/o Mary Andrade, Box 1910, Dept. of Computer Science, Brown University, Providence, RI, 02912.

For further information you may contact: Tom Leighton, ftl@math.mit.edu; Christos Papadimitriou, christos@CS.Berkeley.edu; Moshe Vardi, vardi@cs.rice.edu; Peter Wegner, pw@cs.brown.edu

THE SCOOP !

John Savage

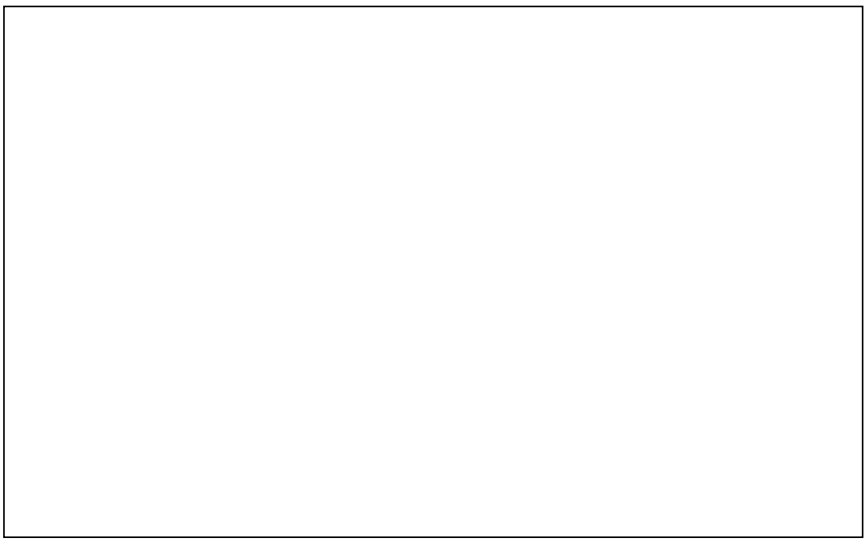
This last February was an eventful month here. First, on February 7, WICS (Women in Computer Science) was given a check for $2,000 by Margaret Cutler of Motorola's Human Resources Information Systems Group in Mansfield, Massachusetts. WICS, together with GIICS (Gender Issues In CS—'geeks'), its adjunct for men, has regular meetings to discuss computer science and gender-related issues. Their intention is to establish a more supportive environment for women in the Department and to encourage those who take introductory courses to pursue a CS degree. In addition, WICS/GIICS has organized various social activities as well as a self-esteem workshop geared towards women. They have also trained the Department's teaching assistants to be more conscious of gender issues. Their long-term goal is to equalize the proportions of men and women studying at all levels and in all areas of CS. More information is available on the Web—*http://www.cs.brown.edu/orgs/wics*. We fully expect WICS/GIICS to put this cash grant to very good use!

Then, on Thursday, February 15, the Department's Industrial Partners Program, together with Brown's Career Planning Services office, sponsored Brown's first job fair for students interested in employment in the computer industry. The event also served to kick off the Industrial Partners Internship Program (IPIP) which was announced in the fall issue of ***conduit!*** IPIP/SCOOP (Summer and Career Organizational Opportunities Program) was open to everyone with an interest in the industry, and was a smash success.

The turnout was excellent. Nearly 300 crowded the Department's space from 3 to 7pm to interview with twenty companies. Students were delighted to be able to interview with so many companies so quickly and easily, and were impressed with the variety of areas represented by the companies. At the end of the evening, the recruiters were exhausted and elated—exhausted by the crush of so many interested students, and elated because so many bright, articulate and well-qualified students had appeared at their tables and booths. Many recruiters emphasized their enthusiasm for our students by holding up sheaves of 70-80 resumes before me. During the day, we polled recruiters to find out whether they found chatting with faculty over lunch and viewing student demos valuable or whether they would rather have 'cut to the chase' and spent more time interviewing students. Their response was unequivocally positive—faculty interac-

tions and demos were what made the IPIP/SCOOP event so much more than just a job fair.

A November *Brown Daily Herald* article about on-campus recruiting quoted Bill Smith '76, Director of R&D for IPP Partner Electronic Book Technologies, as saying: "We're a software company and Brown has one of the best computer science departments in the nation, if not the world. My experience with people coming out of Brown is that they can do really good work from day one. The quality of the program and quality of people I've seen come out over 20 years is like no other."

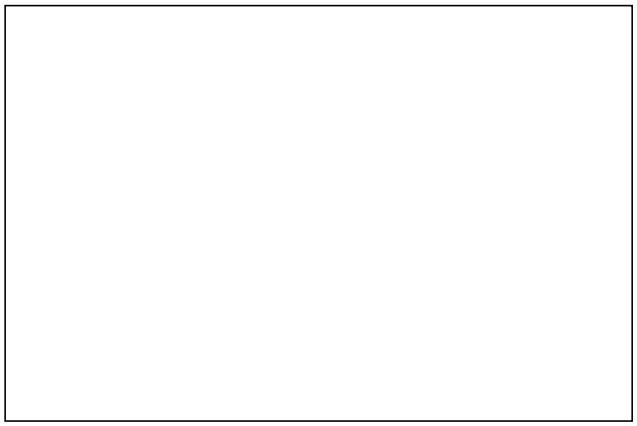


Assembled for an informal check presentation for Women in Computer Science are l to r: Peter Lauro, Development; Suzi Howe, CS; Valerie Green, WICS; Li Markakis, WICS; Eugene Charniak, Chairman, CS; Margaret Cutler, Motorola; Tashana Landray, WICS; and John Savage, CS.

IPIP/SCOOP was also an occasion for many former students to revisit their *alma mater* as recruiters. We were very pleased to welcome Pam Promisel '85, Stephanie Mossburg '92 and Fausto Monacelli '95 from American Management Systems; Bidemi Carroll '95 and Darek Kozlowski '94 of Oracle; Teri Carilli '84 of Cognex; Jon Monsarrat, Ph.D. student, of Second Nature; Guy Sanchez '80 of GTECH; Jeff Vogel '90 and Ed Bielawa '96 of Electronic Book Technologies; Patrick McTurk '95 of Fusion Systems Group, Bruce Munroe '83 and Nicole Yankelovich '83 of Sun Microsystems; Bharathi Subramanian Ph.D. '95 and Phil Thrift '79 of Texas Instruments; and David Durfee '87 and Ph.D. '92 of Bay Computer Associates. We welcome all the recruiters, Brown grads or not, and hope they will return in the fall with some new Brown grads to help with the next recruiting effort.

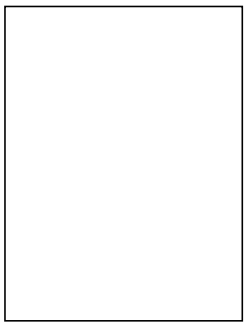
The IPIP/SCOOP day began in the morning as recruiters set up their tables and booths. Around noon a buffet lunch was served for recruiters and faculty members in the atrium. Demos, which our corporate visitors found especially intriguing, were then given by students—these included a demo by CS169 students of their class projects of last semester: Doors '95, an object-oriented operating system; a fluid flow visualization demo by the Graphics Group; and a demo of the Helios system, a constraint system for solving nonlinear programs. After this, recruiters moved to their tables and at 3pm the onslaught of students began, continuing unabated until 7pm. Suzi Howe, Manager of IPP, and Sheila Curran and Ellie Applegate of Career Planning Services teamed together to coordinate and advertise this highly successful event.

We plan to repeat this IPIP/SCOOP event at least once every year. A word to current and prospective IPP Partners: IPP companies are given the choice locations at IPIP/SCOOP, a small but important advantage. We look forward to increased interest in IPP as a result.



Representatives from Industrial Partner companies surrounded by students at the highly successful IPIP/SCOOP kickoff event

TOPOLOGY AND DISTRIBUTED COMPUTATION



Maurice Herlihy

Introduction

The problem of *coordinating* concurrent processes remains one of the central issues in modern distributed and parallel computing. Coordination problems arise at all scales in distributed and concurrent systems, ranging from synchronizing access to shared data objects in tightly coupled multiprocessors, to allocating data paths in ATM networks. Coordination is difficult because modern parallel and distributed systems are inherently *asynchronous*: processes may be delayed without warning for a variety of reasons, including interrupts, preemption, cache misses, communication delays, or failures. These delays can vary enormously in scale: a cache miss might delay a processor for fewer than ten instructions, a page fault for a few million instructions, and operating system preemption for hundreds of millions of instructions. Coordination protocols that do not take such delays into account run the risk that if one processor is unexpectedly delayed, then the remaining processes may be unable to make progress. Such problems become increasingly severe as systems scale.

"a remarkable aspect of this approach is that it relies extensively on concepts taken from algebraic topology, a field of mathematics widely considered to have few, if any, practical applications"

This article focuses on new mathematical techniques for analyzing and evaluating different kinds of coordination techniques. These techniques were developed in collaboration with my colleagues Sergio Rajsbaum of UNAM, Mexico, and Nir Shavit of Tel Aviv University, Israel. To my mind, a remarkable aspect of this approach is that it relies extensively on concepts taken from algebraic topology, a field of mathematics widely (though erroneously) considered to have few, if any, practical applications.

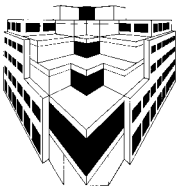
In many multiprocessor systems, processors communicate by applying certain operations, called *synchronization primitives*, to variables in a shared memory. These primitives may simply be reads and writes, or they may include more complex constructs, such as the *swap* operation, which atomically writes a value to a variable and returns the variable's previous contents, *fetch-and-add*, which atomically adds a given quantity to a variable and returns the variable's previous contents, or *compare-and-swap*, which atomically tests whether a variable has a given value and, if so, replaces it with another given value.

Over the years, computer scientists have proposed and implemented a variety of different synchronization primitives, and their relative merits have been the subject of lively debate. Much of this debate has focused on perceived ease of implementation, perceived ease of use, and personal taste. This article describes recently developed conceptual tools that make it possible to provide a mathematically rigorous evaluation of the computational power of various synchronization primitives. This emerging theory could provide the designers of computer networks and multiprocessor architectures with mathematical tools for recognizing when problems are unsolvable, for evaluating alternative synchronization primitives, and for making explicit the assumptions needed to make a problem solvable.

Decision Tasks

Our discussion focuses on a simple but important class of coordination tasks called *decision tasks*. At the start, processors are assigned private *input values* (perhaps transmitted from outside). The processors communicate with one another (for example, by sending messages or by applying operations to a shared memory), and eventually each processor chooses a private *output value* and halts. The decision task is characterized by (1) the set of legitimate input value assignments and (2) for each input value assignment, the set of legitimate output value assignments.

Perhaps the simplest example of a decision task is *consensus*. Each processor starts with an input value and chooses an output value. All processors' output values must agree, and each output value must have been some processor's input value. If the input values are boolean, the task is called *binary consensus*. The consensus task was originally studied as an idealization of the problem of committing a distributed transaction, where a number of database sites must



agree on whether to commit or abort a distributed transaction.

A natural generalization of consensus is *k-set agreement*. Like consensus, each process's output value must be some process's input value. Unlike consensus, which requires that all processes agree, *k*-set agreement requires that no more than *k* distinct output values be chosen. Consensus is 1-set agreement.

Another interesting example is the following *renaming* task. For input values, each processor is assigned a unique identifier taken from a large range (such as a Social Security number). For output values, the processors must choose unique values taken from a much smaller range. Renaming is an abstraction of a variety of resource allocation problems.

To solve a decision task, a processor executes a program called a *protocol*. Because processors are subject to sudden delays and because halting one processor for an arbitrary duration should not prevent the others from making progress, we require that each processor finish its protocol in a fixed number of steps, regardless of how its steps are interleaved with those of other processors. Such a protocol is said to be *wait-free*, since it implies that no processor can wait for another to do anything. We focus here on wait-free protocols.

Connections with Topology

A decision task has a simple geometric representation. Assume we have n+1 processes, $P_0,...,P_n$, each assigned a different color. A processor's state before starting a task is represented as a point in a high-dimensional Euclidean space. This point, called an *input vertex*, is labeled with a processor color and an input value. Two input vertices are *compatible* if (1) they have distinct colors and (2) there exists a legitimate input value assignment that simultaneously assigns those values to those processes.

Fig.1: Input and output complexes for 2-processor consensus

For example, in the binary consensus task described earlier, input values are either 0 or 1, so any two input vertices are compatible if and only if they have distinct colors. We join any two compatible input vertices with an edge, any three with a solid triangle, and any four with a solid tetrahedron. In general, any set of (k+1)-compatible input vertices spans a k-dimensional simplex (called an *input k-simplex*). The set of all possible input simplexes forms a mathematical structure called a *simplicial complex*. We call this structure the task's *input complex*.

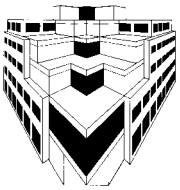
The notions of an *output vertex*, *output simplex*, and the task's *output complex* are defined in the same way, simply replacing input values with output values. The decision task itself is defined by a relation Δ that carries each input n-simplex to a set of output n-simplexes. This relation has the following meaning: if S is an input simplex, T is an output simplex, and the processors start with their respective input values from S, then it is acceptable for them to halt with their respective output values from T.

We now review each of the tasks described above. The input complex for binary consensus is constructed by assigning independent binary values to n+1 processes. We call this complex the *binary n-sphere*, because it is topologically equivalent to an n-dimensional sphere (the reader is encouraged to verify this claim). The output complex consists of two disjoint n-simplexes, corresponding to decision values 0 and 1. Figure 1 illustrates the input and output complexes for two-processor binary consensus.

Next, consider the 2-set agreement task for three processes. The three processes must choose at most two distinct values. It is not hard to see that the output complex for this task consists of three binary 2-spheres "linked" in a ring; see Figure 2.

Finally, consider the *renaming task*, in which each processor is given a unique input name taken from a large name space and must choose a unique output name taken from a much smaller name space. Figure 3 shows the output complex for the three-processor renaming task using four output names. Notice that the two edges marked *A* are identical, as are the two edges marked *B*. By identifying these edges, we can see that this complex is topologically equivalent to a torus.

We have shown how to specify a decision task with a geometric model. We now do the same for the protocols that solve such tasks. Recall that a protocol is a program where each processor starts out with a private input value, communicates with the other processors, and then chooses an output value based on the results of the computation. When the protocol has "heard enough," it chooses its output value by applying a *decision map* to its local state.



The set of all possible executions also defines a simplicial complex. Each vertex is labeled with a processor color and a *history*, the sequence of operations (with results) executed by that processor. Two vertices are compatible if they have distinct colors and some protocol execution exists in which those processes observe those histories. This is the protocol's *protocol complex*. More precisely, for every input simplex S, any protocol induces a corresponding protocol complex $P(S)$. The union of these complexes is the protocol complex for the protocol.

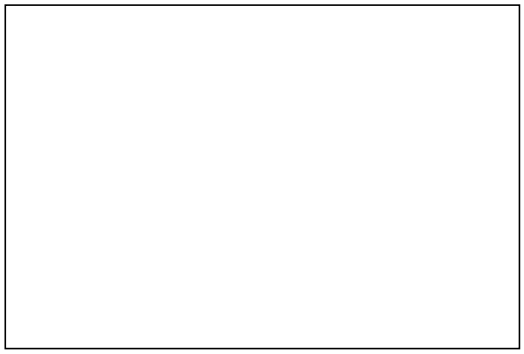


Fig.2: Output complex for (3,2)-set agreement

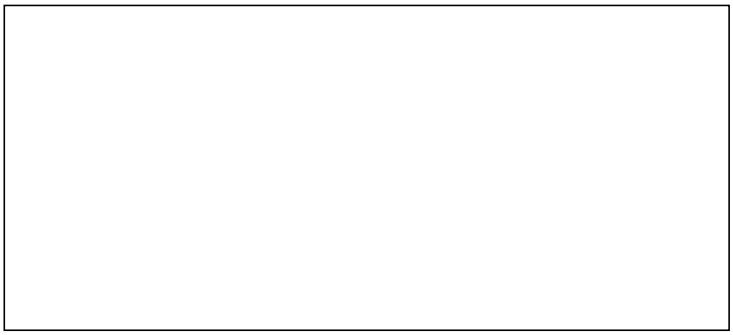


Fig.3: Output complex for 3-processor renaming with 4 names

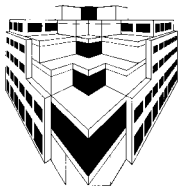
What does it mean for a protocol to solve a decision task? Recall that a *decision map* δ carries each history h to the output value chosen by the protocol after observing h. The decision map induces a map from the protocol complex to the output complex: $\delta(\langle P,h\rangle) = \langle P,\delta(h)\rangle$. We are now ready to give a precise geometric statement of what it means for a protocol to solve a decision task. Given a decision task with input complex I, output complex O, and relation Δ, a protocol solves a decision task if and only if, for every input simplex $S \in I$ and every protocol simplex, $T \in P(S)$, $\delta(T) \subset \Delta(T)$.

This definition is simply a formal way of stating that every execution of the protocol must yield an output value assignment permitted by the decision task specification. Roundabout as this formulation may seem, it has an important advantage. We have moved from an operational notion of a decision task, expressed in terms of computations unfolding in time, to a purely combinatorial description expressed in terms of relations among topological spaces. It is typically easier to reason about static mathematical relations than ongoing computations, but more importantly, this model lets us exploit classical results from the rich literature on algebraic and combinatorial topology. It is therefore unnecessary to prove each claim from scratch, a departure from the customary practice in some areas of computer science.

To prove that certain decision tasks cannot be solved by certain classes of protocols, it is enough to show that no decision map exists. We can derive a number of impossibility results by exploiting basic properties that any decision map must have. In particular, any decision map is a *simplicial map*: it carries vertices to vertices, but it also carries simplexes to simplexes. Simplicial maps are also *continuous*: they preserve topological structure. If we can show that a class of protocols generates protocol complexes that are "topologically incompatible" with the task's output complex, then we have established impossibility. Conversely, if we can prove that the decision map exists, then we have shown that a protocol exists.

A complex has *no holes* if any sphere embedded in the complex can be continuously deformed to a point while remaining inside the complex. (More technically, the complex has trivial homotopy groups.) It has *no holes up to dimension d* if the same property holds for spheres of dimension d or less. (Notice that when d is zero, this condition means the complex is connected.) For example, a two-dimensional disk (e.g., a plate) has no holes, and a two-dimensional sphere (e.g., a basketball) has no holes up to dimension one, because any loop (e.g., a rubber band) on the sphere can be deformed to a point. By contrast, a torus has no holes only up to dimension zero—it is connected, but not every 1-sphere (loop) placed on the surface can be deformed to a point.

The protocol complexes for read/write protocols have a remarkable property: for any input simplex S, the protocol complex $P(S)$ has no holes. This property holds for any read/write protocol, no matter how many variables it uses or how long it runs. This property is a powerful tool for proving impossibility results. Careful analysis of renaming shows that if fewer than $2n+1$ output values are possible, then the output complex has a hole. Moreover, any decision map must "wrap" a particular sphere in the protocol complex around that hole in such a way that the sphere's image cannot be



continuously deformed to a single point. Because the protocol complex has no holes, however, that sphere can be continuously deformed to a point in the protocol complex. Because the decision map is continuous, the image of that sphere can also be contracted to a point, and we have a contradiction. The same kind of analysis shows that a variety of fundamental synchronization problems have no wait-free solutions in read/write memory.

Fig.4: Existence condition for read/write protocols

This topological model also yields a "universal" algorithm that can be used to solve any task that can be solved by a wait-free read/write protocol. Any decision task can be considered as a kind of "approximate agreement" task in which each processor chooses a vertex in the output complex, and the processors negotiate among themselves to ensure that all processors choose vertices of a common simplex. This task, which we call "simplex agreement," provides a simple normal form for any decision task protocol.

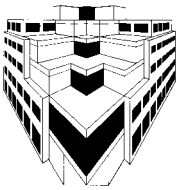
We can combine these two notions to give a complete characterization of the decision tasks that can be solved by wait-free read/write protocols. Because the exact conditions require some technical definitions beyond the scope of this article, the focus here is on the underlying intuition. A decision task has a wait-free read/write protocol if and only if the relation Δ can be "approximated" by a continuous map on its underlying point set, in the following sense. Given the input complex I, construct a new complex, $\sigma(I)$, by subdividing each simplex in I into smaller simplexes. If v is a vertex in $\sigma(I)$, define *carrier* (v) to be the smallest simplex in I that contains v. The decision task is solvable in read/write memory if and only if there exists a subdivision $\sigma(I)$ and a simplicial map $\mu : \sigma(I) \rightarrow O$ such that for each vertex $v \in \sigma(I)$, $\mu(v) \in \Delta(carrier\ (v))$. Informally, this condition states that it must be possible to "stretch" and "fold" the input complex so that each input simplex can cover its corresponding output simplexes.

This condition is shown schematically in Figure 4. The top half of the figure illustrates the relation Δ for a generic decision task, and the bottom half shows how Δ can be approximated by a simplicial (continuous) map μ.

Other Kinds of Protocols

Although read/write protocols have considerable theoretical interest, real multiprocessors typically provide more powerful synchronization primitives. The topology of protocol complexes for such protocols is more complicated. For example, Figure 5 shows the protocol complexes for two simple protocols in which processors communicate by applying test-and-set operations to shared variables. Casual inspection shows that these protocol complexes differ from their read/write counterparts in one fundamental respect: they have one-dimensional holes. Nevertheless, they do resemble them in another respect: they are connected. In general, any protocol in which (n+1) processors

Fig.5: Protocol complexes for some test-and-set protocols



communicate by pair-wise sharing of test-and-set variables has a protocol complex with no holes up to dimension $\lfloor n/2 \rfloor$.

In a recent paper, Sergio Rajsbaum and I analyzed the topological properties of protocol complexes for a family of synchronization primitives called *k-consensus* objects, which encompasses many of the synchronization primitives in use today. The larger the value of k, the more powerful is the primitive. The protocol complex for any protocol in which processes communicate via k-consensus objects has no holes up to dimension $\lfloor n/k \rfloor$. So at one extreme, when $k = 1$, the complex has no holes at all, and at the other extreme, the complex becomes disconnected. As k ranges from 1 to $n+1$, holes appear first in higher dimensions and then spread to lower dimensions. A surprising implication of this structure is that there exist simple synchronization primitives that are *incomparable*: it is impossible to construct a wait-free implementation of one from the other.

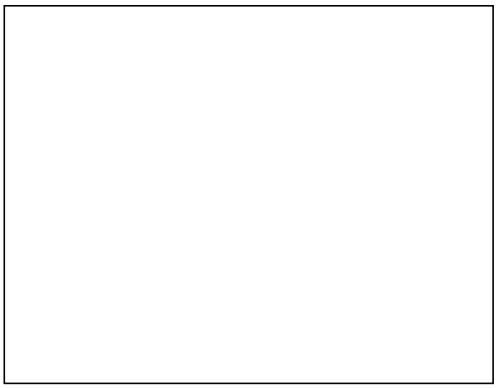


Fig.6: Contractible and non-contractible loops

Decidability

We say that tasks are *decidable* in a given model of computation if there exists an effective procedure for deciding whether a task has a t-resilient protocol in that model. In a recent paper, Sergio Rajsbaum and I give a complete characterization of the circumstances under which tasks are decidable in a variety of models of computation. Biran, Moran, and Zaks had shown that tasks are decidable in the message-passing model if at most one processor can fail. We were able to show that their result is tight: message-passing tasks become undecidable if two or more processors can fail. An (m,k)-set agreement object is any object that allows m processes to solve k-set agreement. If processes communicate by (m,k)-set agreement objects, where $k > 2$, then tasks are decidable if and only if, no more than one processor can fail. If $k = 2$, then tasks are decidable if and only if, fewer than m processes can fail, and if $k = 1$, tasks are decidable if and only if fewer than $2m$ can fail. Our proof exploits the undecidability of the *contractibility* problem, which requires deciding whether a loop L in a finite simplicial complex K can be continuously deformed to a point (see Figure 6).

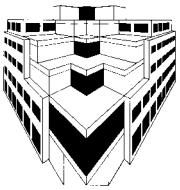
Conclusions

We believe this topological approach has a great deal of promise for the theory of distributed and concurrent computation, and that it merits further investigation. It has already produced a number of new and unexpected results, and has illuminated an unexpected connection between the emerging theory of concurrent computation and the well-established theories of algebraic and combinatorial topology.

THE RED PENCIL

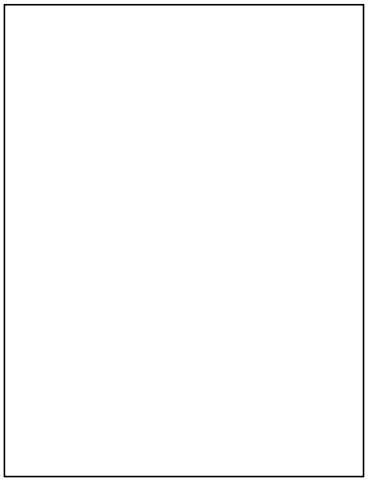
Few can claim greater longevity in the Department than Computer Science's Department Manager, Katrina Avery. Trina came in '80, just after the *crème de la crème* of the Applied Math and Engineering faculties coalesced to create the Department of Computer Science. She continues today as its administrative backbone, historical resource and mainstay and enjoys a formidable reputation as departmental editor *extraordinaire.*

After what she deems a very happy childhood in Cambridge, MA, Trina graduated from Radcliffe, class of '61. (Coincidentally, the mother of Anne Morgan Spalter (one of Andy van Dam's colleagues associated with the NSF/STC program) was a fellow Cliffie and dormmate of Trina's. Anne's office is opposite Trina's and for some time Trina experienced *déjà vu* every time she saw her until the resemblance finally struck a chord and she made the connection.) Despite her Yankee bearing, Trina likes to tell of her family's modest beginnings. Her Swedish grandfather was a blacksmith and her father, while at public school in Exeter, NH, was singled out for his scholarly potential by a benefactor and enrolled at Exeter Academy. In high school and as a young adult, Trina's talent for singing and playing the piano blossomed. After college she worked part-time at MIT and took the freshman course at the New England Conservatory. In a recent academic venture she audited three semesters of Anglo-Saxon, the high point of which was reading *Beowulf*. She



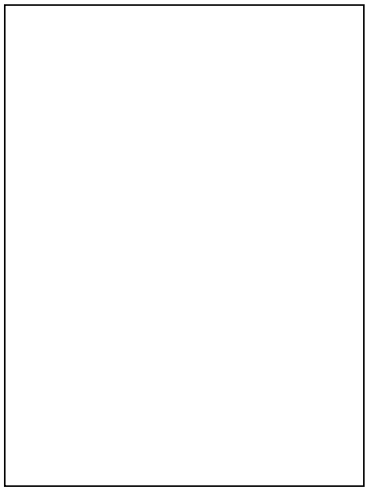
cites the sudden realization that 'Time's wingèd chariot' was fast on her heels for deciding to pursue her lifelong interest in early English—she hadn't studied anything earlier than Chaucer prior to taking this course.

Trina came to Brown in '65 as a faculty wife. She began working in the Applied Math Department, where she first met Andy. In '68 her daughter Jessica was born and in '72 she returned to Applied Math as a secretary. During this stint she began taking courses in Greek on a dare, getting to the point where she was able to earn graduate credit. She decided to forge ahead with a Ph.D. and for the next several years was a graduate student. However, despite her love of Greek and Latin, she realized just how much she hated writing and that getting a doctorate was merely an opportunity to do more of it! So she opted out with an ABD and a greater self-awareness that focused her preference for rewriting the work of others into her exceptional skill as an editor. This had first been tapped by a friend of her parents at MIT who was writing a textbook. He thought a Radcliffe English major would be perfect to pull the volume together (for college freshmen on metallurgy) since it was pitched at a beginner's level.



Trina and Ralph—a cat who was nursed by a dog!

Trina grew up with cats and is known as a serious cat lover. Over the years she has had many long-lived cats and currently has four. Besides her feline fancy, she is a fine cook ('it's much more fun than vacuuming'), a member of the Providence Singers (alto and treasurer), and an avid scuba diver. She took up diving in the '70s and has since introduced Tom Doeppner to the sport. Tom's enthusiasm was contagious and they're now keen enough to venture to exotic places—Solomon Islands, Borneo, Papua New Guinea, as well as the Caribbean. As if the above weren't enough to keep her busy, she also works out three times a week (step aerobics) with her CS colleagues and takes yoga classes regularly. Whenever possible, she enjoys spending time at her favorite Boston museum, the Gardner, and at the Metropolitan in New York City, where her daughter lives.



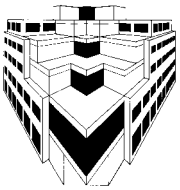
Lost in the blue in Borneo

The first faculty members in CS were Andy van Dam, Tom Doeppner, Bob Sedgewick and Peter Wegner from Applied Math and Eugene Charniak, John Savage and Steve Reiss from Engineering. Trina joined CS at a time when the Department was very much inventing the wheel; van Dam was on a crusade to reform the world with computers—he was often right—and there was much *esprit de corps* and excitement. She misses that a bit, but not the immense panics and deadlines attendant on learning how to pull together critically important proposals and budgets for University Hall. Once the Program in Computer Science became the Department of Computer Science, it moved to Kassar House on Thayer Street—a non-smoking building. This posed a problem for Trina, then a long-time smoker (her first puff at age five resulted in her and a friend burning down an empty chicken coop); however, she was able to quit rather quickly because both Andy and Tom Doeppner bribed her with $5 each for every week she didn't smoke—it worked very well—her last puff occurred in Italy while attending a conference with Tom in August '82 just after a hike in the Swiss Alps! Trina's first CS office was one third the size of her current space, with Andy's on one side and John Savage's on the other—definitely where the action was. Eventually, an addition was built on to Kassar House. During excavation a store of 19th-century glass bottles was unearthed. Since no one else expressed an interest, they now grace Trina's kitchen. The Department ultimately moved to its present location in the CIT Building and Trina bore some of the brunt of dealing with the changes getting larger entails, such as the need for creating procedures for the free-wheeling way of doing business previously enjoyed.

The CS faculty hold her in high regard, as the following remarks attest:

Says Tom Doeppner, "When I arrived at Brown as a member of the Division of Applied Mathematics in '76, I was told of this woman's wonderful reputation for returning papers she'd typed along with notes. For example, she pointed out to Ulf Grenander that there was a bug in one of his equations (Trina claims that the bug was spotted for reasons of syntax, not semantics); she pointed out to a few people that terms had not been adequately defined in their papers. I've since come to rely on her for, among other things, copy-editing almost everything I write. Indeed many others do also—she's copy-edited a number of books for Addison-Wesley and MIT Press, for researchers in Korea and Europe as well as us locals.

When Trina first started working at the CS department, she was very much anti-high tech-



nology. She's been slowly and grudgingly adapting her views since then. Her daughter was shocked to discover that her mother now has a Web page (http://www.cs.brown.edu/people/kha). Trina still occasionally does the odd typing job, though rarely these days. Once, in '81, some kid came by her office asking if she would type his paper. She said she was pretty busy right now, so it would cost him the earth even if she could find time for it. The kid decided against it and left. During this exchange, she didn't notice the strange expressions on the face of ugrad Randy Pausch, now on the faculty at the Univ. of Virginia, who was sitting in her office. Randy pointed out that she'd just turned down John F. Kennedy Jr. who, though perhaps not being able to afford the earth, could possibly manage RI!"

According to Andy, he always introduces Trina as the world's best red pencil and views her as indispensible when it comes to reviewing any kind of document, proposal, book, article or news blurb because she not only reads it more incisively than anyone else would, but also manages to understand it. When there is red ink on practically every sentence and Trina says 'not bad,' that is high praise indeed!

Says chairman, Eugene Charniak, "Everyone comments on how fast Trina can copy-edit. However, in my opinion, this arises from a less noticed fact, that she reads faster than anyone I know. I find it discouraging when, after laboriously plowing through a book (most recently "How Buildings Learn"), I loan it to Trina and get it back two days later read cover to cover.

To hundreds of CS undergraduate and graduate students Trina is the keeper of the departmental pursestrings and a force to be reckoned with—somewhat aloof but always knowledgeable and often engaging. To faculty she is a valuable historical resource, an invaluable copy-editor, and to many, a very good friend.

Trina and Selectric—still somewhat technologically resistant!

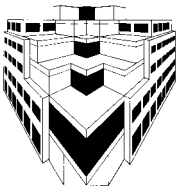
THE 16TH IPP SYMPOSIUM

Maurice Herlihy

On October 19, 1995, the Department hosted its 16th Industrial Partners Program technical symposium. Entitled "Parallel Architectures in Today's Marketplace," the symposium brought together speakers from IBM, Sun Microsystems, Digital Equipment Corporation, Microsoft, and Motorola. The marketplace for multiprocessor machines is undergoing accelerating changes. While several companies specializing in large-scale multiprocessors have recently encountered difficulties, others are doing quite well by adding parallel machines to existing product lines. The symposium focused on the technical issues underlying these trends as well as on issues likely to influence the future of the industry.

David Douglas of Sun Microsystems led off the program with the talk "Mainstream parallelism: taking sides in the SMP/MPP/cluster debate." He observed that today's commercial multiprocessors fall into three broad categories: massively parallel processors (MPPs), shared-memory multiprocessors (SMPs), and clusters. He argued that SMPs dominate the market now and will continue to do so for the foreseeable future as they move into higher-end uses. MPPs and clusters will not disappear, but will increasingly become niche players, possibly competing against one another. Although there are cogent arguments why MPPs should be cheaper, more scalable, and more reliable than SMPs, these arguments do not seem to work in practice.

Marc Snir of the IBM T.J. Watson Research Center followed with an overview of the IBM SP product line. The SP architecture is based on a balance between commodity and custom technology. It has its roots in a research project aimed at building a massively parallel teraflop scientific supercomputer and in technology developed to manage workstation clusters. The current SP product still reflects this duality. The commodity technologies are well suited to small, cost-conscious markets: they have a lower development cost and shorter time to market. Unfortunately, an architecture based entirely on commodity technology does not



provide adequate performance and scalability. The approach taken in the SP product line is to combine commodity workstation-cluster technology with a few critical custom technologies. Snir finished by pointing out that scalable parallel servers are here to stay, and the only practical approach to designing such architectures is to make significant reuse of commodity technology.

Bill Bolosky of Microsoft gave the audience a glimpse of the kinds of massively parallel applications that will, he believes, drive the marketplace of the future. He described the Tiger video file system, a distributed, fault-tolerant file system that provides files at a constant and guaranteed bit rate. Tiger balances the load by striping files over all the disks in the system and by introducing stream startup latency to avoid resource contention among different streams. Tiger is implemented on a collection of PCs connected by an ATM network.

Symposium speakers, back row, l to r: Bill Bolosky, Microsoft; Tony Dahbura, Motorola; Joel Emer, DEC; David Douglas, Sun. Front row, l to r: Maurice Herlihy, host; Marco Annaratone, DEC; Marc Snir, IBM

After lunch, Joel Emer of DEC described some of the technology underlying modern high-performance microprocessors. Frequently, because of timing and implementation constraints, high-performance microprocessors use small, direct-mapped, on-chip caches. With such caches, the miss rates are typically worse and show greater variability than comparably sized set-associative caches. On the other hand, set-associative caches typically have longer latency and therefore frequently worse net performance than direct-mapped caches. Joel described the multiprobe cache, an intermediate design that combines many of the benefits of both approaches: a multiprobe cache is a direct-mapped structure that is accessed serially with different hashing functions, or probes, that look in each of the locations in which the data can be held. Joel reviewed techniques developed by his group for data and instruction caches that ensure fast access times and a high percentage of successful first probes.

Marco Annaratone of DEC gave a talk entitled "Scalable computing, parallel processing, and the information technology market" describing the views of his research groups on the structure and future of the parallel marketplace. He argued that the choice of CPU is driven more by application availability than by cost, and that a well-designed network interface, logic, etc. add little to the overall cost. The key to success is to focus on tomorrow's applications, and a focus on programmability and usability is essential. In a market driven by applications, small-scale SMPs will be the winners in the coming years, although the preferred programming paradigm is yet to be determined. Nevertheless, things can change suddenly with the emergence of new "anchor applications" and increasing connectivity.

Tony Dahbura of Motorola gave a brief history of the Monsoon and StarT projects, two collaborative efforts between Motorola and MIT to develop advanced parallel architectures based on dataflow architectures and languages. The projects underwent many shifts in response to changes in the industrial and research communities. Tony discussed how this project was affected by ongoing changes in commercial and research multiprocessing, and closed with some reflections on the delicate balances needed to do industrial research in the '90s. The day ended with a lively panel discussion.

Eugene Charniak

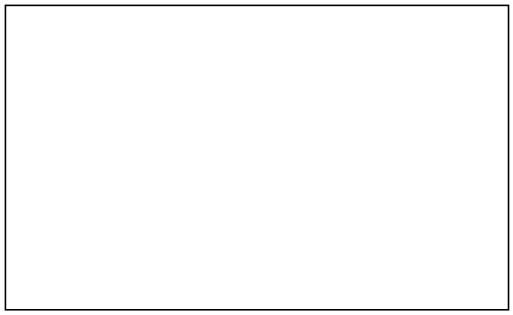
In the last six months Eugene took his first trip to Korea as an invited lecturer at the Third Natural Language Processing Pacific-Rim Symposium. He also gave two invited talks at US universities. Finally, he agreed to be on the program committee for four conferences this year, as well as being the outside reviewer for a Ph.D. thesis, not noticing that the due dates for the Ph.D. thesis and three of the four conferences were within three weeks of one another. At this point he has done three of these four tasks and never wants to read another conference paper!

Leslie Kaelbling

Leslie is currently on sabbatical visiting Harvard and MIT. Last semester she gave invited talks at an impressive list of thirteen different European locales. This semester she is on six program committees—American Association for AI, Senior Member; International Conference on Machine Learning, Senior Member; IEEE International Conference on Pattern Recognition, IEEE/RSJ International Conference on Robots and Systems, Simulation of Adaptive Behavior and Uncertainty in Artificial Intelligence. She's reading lots of papers!

Franco Preparata

In August Franco gave an invited talk in plenary session at WADS '95 (Workshop on Algorithms and Data Structure) in Kingston, Ontario. He was also plenary speaker at ISAAC '95 (International Symposium on Algorithms and Computing) in Cairns, Australia, last December. Subsequently he was one of three lecturers at a week-long advanced school on parallel computation near Bangkok, Thailand. Franco was also recognized by his colleagues at Kyoto University who have published a volume (in Japanese) entitled "Professor Preparata's Lectures on Parallel Computation."



Van Dam receives his honorary doctorate from the president of the Technical University at Darmstadt

John Savage

John has been reappointed to the MIT Corporation Visiting Committee for the Department of Electrical Engineering and Computer Science.

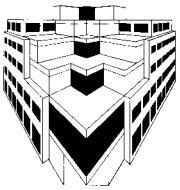
Roberto Tamassia

In December Roberto gave an invited talk at the University of Seville, Spain. He is on the program committee of AVI '96 (Workshop on Advanced Visual Interfaces).

Andries van Dam

Last November Andy was flown out to the Pixar screening of the movie "Toy Story" in San Francisco as one of Steve Jobs' guests. Attending the black-tie reception that followed were business friends of Pixar, movie stars and a raft of CS alumni. Andy gives the movie five stars and counts it an incredible technological achievement; he has seen it twice since his return and particularly enjoys seeing the names of former graphics group members scrolling by in the credits—Ronen Barzel '84, Galyn Susman '86, Eben Fiske Ostby '79, Michael Shantzis '86 and David Salesin '83. Since the movie, Pixar has hired yet another alum, Kurt Fleischer '82. Co-author and Director of "Toy Story" John Lasseter signed Andy's copy of the Toy Story book "Thanks for the students!" Jobs wrote "You helped make this so." Says van Dam, "The whole thing was a real high, and the food and drink were good too."

In December Andy was awarded an honorary doctorate from the Technical University at Darmstadt, Germany, and within a month of his return received notice from his *alma mater*, Swarthmore, that he would be receiving an honorary doctorate at their commencement ceremony in June. In addition, we have just learned that Andy has been elected to the National Academy of Engineering, one of the highest professional distinctions given to an engineer. Membership honors those who have made important contributions to engineering theory and practice and those who have demonstrated unusual accomplishment in pioneering new and developing fields of technology.



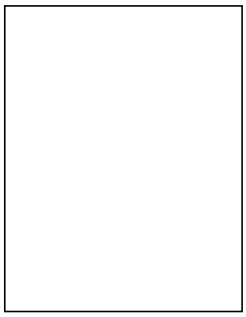
Pascal Van Hentenryck

This winter Pascal became an expert in architecture—more precisely, in snowman architecture! This spring he will be an invited speaker at PACT '96 (Practical Applications of Constraint Technology) and at the SIAM conference on optimization.

Peter Wegner

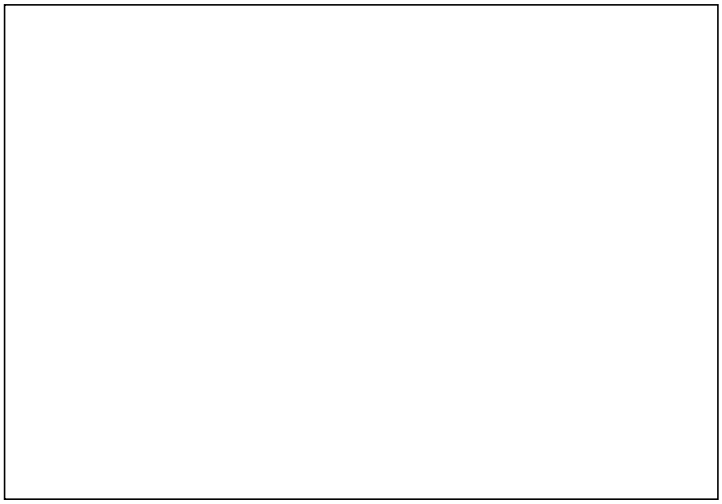
Peter was an invited speaker at a January workshop in Rome on the Foundations of Computing and at an April workshop near Bologna on Coordination Models. He is organizing the ACM 50th anniversary workshop on Strategic Directions in Computing at MIT in June, will participate in an ONR workshop on automated formal methods in Oxford in June, and will be conference co-chairman of the European Conference on Object-Oriented Programming (ECOOP) in Austria in July. He edited the special 50th-anniversary *Computing Surveys* issue on Perspectives in Computing, featuring 70 short articles on the status of subfields of computing, that will appear in May.

FROM THE CHAIRMAN, Eugene Charniak



Eugene Charniak

A happy event since the last ***conduit!*** was the receipt of a largish check from the Digital Equipment Corporation. For several years now DEC has been selling the programming environment DEC Fuse. In fact, DEC licensed this software from Brown, where in its previous incarnation it was Prof. Steve Reiss's Field system. This license agreement has been a good deal for both DEC and Brown, and both of us have made some money (as has Steve Reiss). However, success has overtaken Field/Fuse: the message-passing scheme that is its foundation has become commonplace for such tools, and there now exists an industry-wide standard for the technology. DEC wanted to be able to replace the Fuse message-passing scheme with the standard, and more generally wanted to be able to take bits and pieces of Fuse and combine them with other products. Thus DEC offered to buy out the royalty agreement, and on October 26 Chuck Piper from DEC came to Brown with a check, happily received by Bill Jackson from BURF (Brown University Research Foundation), John Savage (who was instrumental in negotiating the agreement), Steve Reiss, and myself.



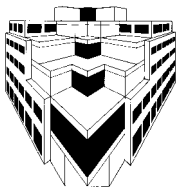
l to r: John Savage, Eugene Charniak (happily wielding check!), Steve Reiss, Chuck Piper from DEC and Bill Jackson, Brown University Research Foundation

Since the fall issue of ***conduit!*** we've had three Ph.D. defenses—**Ashim Garg**, whose topic "Where to Draw the Line" was hosted by Roberto Tamassia; Ashim is now a post-doc in the Department. **Cindy Grimm**, known for her pet cockatoo, Tia (see spring '93 issue of ***conduit!***), her prowess as an Aikido and Hapkido black belt and a major talent at ballroom dancing, has been working on "Modeling Surfaces of Arbitrary Topology Using Manifolds," and her defense was hosted by John Hughes. **Michael Littman**'s defense, on "Algorithms for Sequential Decision Making," was hosted by Leslie Kaelbling; Michael is looking for a teaching position.

Probably the most unusual event of the last semester was a fax from Robert Stern, a Hong-Kong-based reporter from CNBC Asia. He was trying to track down a program he'd heard was developed at Brown for chicken sexing.

It turns out that sexing chickens is an important but difficult art in the poultry business: only the females are worth raising, since the males stoutly refuse to lay eggs, but it is nearly impossible for the uninitiated to distinguish between day-old male and female chicks.

We here were intrigued. At the very least it would make a good topic for ***conduit!*** Unfortu-



nately, to the best of my knowledge, no such work had been done in this Department, and since the program was reported to be neural-network-based I should have been aware of it. Just in case, however, I sent an email inquiry to the faculty. I got back one (ultimately unfruitful) lead to a former undergraduate, a few ribald comments, and some concerns about the Brown sexual discrimination policy.

I also got in touch with Jim Anderson (Chair of the Cognitive Science Department), who is a neural networks maven here at Brown. Jim did not know of any neural-network program attacking the sexing problem, but he was aware of a landmark paper on the topic by Irving Biederman and Margaret Shiffrar. He sent us a copy which we forwarded to Mr. Stern, and we also got in touch with Mr. Biederman. Noted Stern, "The paper was ominously silent on the fate of the male chicks...." Replied Biederman "As I understand it, except for a very lucky and happy few, they are recycled." Stern's last words on the subject were "I'll pause for a moment's reflection on the noble hecatombs of male chicks before I tuck into my next *coq au vin.* Breast regards." The Biederman-Shiffrar paper is loaded with insights into the problem, but perversely we were intrigued by the fact that the research was sponsored by the U.S. Air Force Office of Scientific Research, as well as by the last line of the paper's abstract, which notes the parallel between "learning to sex chicks and learning to classify tanks as friend or foe."

Kate Sanders last appeared in ***conduit!*** (along with her sexless rubber chicken) in Volume 3 Number 2 (I hope all of you folks are remembering to bind your issues!). The occasion was her receiving her Ph.D.; at that time she was off to the University of Maryland on a one-year postdoc. Kate has now taken a position at Brown in CIS, where she is serving as senior analyst and working for Steve Carmody, one of Andy's first students. It is a pleasure to have her back at Brown!

conduit!

A publication of
The Computer Science Department
Brown University

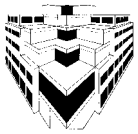
Inquiries to: ***conduit!***
Department of Computer Science
Box 1910, Brown University
Providence, RI 02912
FAX: 401-863-7657
PHONE: 401-863-7610
EMAIL: sjh@cs.brown.edu
WWW: http://www.cs.brown.edu/publications/conduit/

Suzi Howe
Editor-in-Chief

Katrina Avery
Editor

Jeff Coady
Technical Support

Department of Computer Science
Brown University
Box 1910, Providence, RI 02912



conduit!

NON-PROFIT
U.S. Postage
PAID
Providence, RI
Permit #202

Printed on recyled paper

Address changes welcomed