ACM SIGACT News Distributed Computing Column 15

Sergio Rajsbaum*

Abstract

The Distributed Computing Column covers the theory of systems that are composed of a number of interacting computing elements. These include problems of communication and networking, databases, distributed shared memory, multiprocessor architectures, operating systems, verification, Internet, and the Web.

This issue contains two items:

- An eulogy in memory of Shimon Even, who passed away on May 1st, 2004. He was a highly influential educator, and played a major role in establishing computer science education in Israel. He did pioneering work in various areas, including parallel and distributed computing. As Oded Goldreich says in his PODC 2004 proceedings eulogy, Shimon Even was a great scientist and a remarkable person. His attitudes toward research have influenced anybody who was fortunate to have an interaction with him, he always followed his own judgment and understanding, rather than the common trend or fashion. He sought simple solutions that uncover the essence of the problem. He spent much time seeking the best way to present his own work as well as the work of others in lectures and writing. This quote comes to mind, about having found a good problem to work on.

  > *Not that I quite know indeed what situations the seeking fabulist does "find"; he seeks them enough assuredly, but he discoveries are, like those of the navigator, the chemist, the biologist, scarce more than alert recognitions. He* comes upon *the interesting thing as Columbus came upon the isle of San Salvador, because he had moved in the right direction for it—also because he knew, with the encounter, what "making land" then and there represented.*
  >
  > Henry James

- The 2004 Gödel Prize is great news for the distributed computing community: it is shared between two journal papers that discovered a fundamental relation between distributed computing and topology, by Maurice Herlihy and Nir Shavit, and Michael Saks and Fotios Zaharoglou. To celebrate this, I am happy to present personal perspectives by the authors of these papers, and by Elizabeth Borowsky and Eli Gafni, that coauthored a third paper with the discovery, but only in a conference and hence not eligible for the prize. I include perspectives also by other people that produced seminal work on the same subject: Soma Chaudhuri and Shlomo Moran.

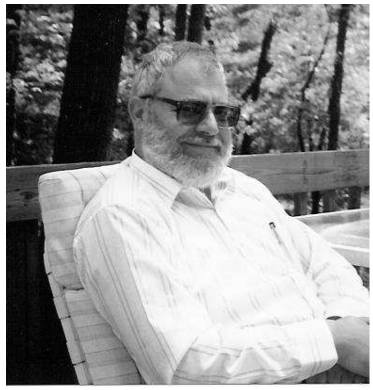
  The DISC conference is collocated this year with GETCO, a workshop on topological methods in concurrency and distributed computing. They will dedicate a joint session to celebrate the prize.

Many thanks to the contributors of this issue.

Request for Collaborations: Please send me any suggestions for material I should be including in this column, including news and communications, open problems, and authors willing to write a guest column or to review an event related to theory of distributed computing.

*Instituto de Matemáticas, UNAM. Ciudad Universitaria, Mexico City, D.F. 04510 `rajsbaum@math.unam.mx`.

Shimon Even 1935–2004



The first time I met Shimon personally was around October 1987. I went to talk with him when I was a Masters student at the Technion looking for a thesis subject. I was not aware of the tremendous prestige and strong image he had at the Technion; people used to think he advised only "geniuses" (and at the Technion people understand through examples of world class famous leaders that this word has a *very* strong meaning), and many even feared him. Simply, I liked his research interests, I knew he had written a classic textbook on graph algorithms, and I thought he was probably the best teacher of the Department of Computer Science. So very innocently I knocked on his door, and said "hi" with a smile. He welcomed me warmly, also with a smile, and for a long time told me some of his wonderful stories about famous computer scientists. He was interested to know I had recently arrived to Israel from Mexico, and told me some more stories about the old days of the country's history. He said I should choose between one of his two current interests: Crypto and Distributed Computing. I chose the latter, heard some more funny stories about his former student Baruch Awerbuch, and I left the room motivated and eager to start reading Baruch's paper "Complexity of network synchronization"— I was to present him the following week. For a young Masters student this meeting was an unforgettable experience. Later on, I was getting admiration comments from students and even secretaries, "wow, he accepted you as a student?!" and so on.

One day he suggested I switch to continue working on the same subject with him toward a PhD (at the Technion there is a "direct path" option to do this switch). Again, somewhat innocently I agreed, and I literally became the "hero" of some people (and realized how far away from looking like a "genius" I looked). When I first met him I had no clue that this was his image, and it surprised people to learn that he knew nothing about me, except that I had taken his Crypto course and not excelled, and yet received me so warmly. I feel my friendship with him started that very first meeting. After having traveled with him several times in Europe and Mexico, I have my own collection of stories to tell.

Shimon's indirect influence on distributed computing (through his papers and his students) is not small. When I first met Shimon, he was working with his PhD student Baruch Awerbuch, who graduated in 1984. They published a paper together in PODC'84, titled "Efficient and Reliable Broadcast is Achievable in an Eventually Connected Network," that appeared in 1986, in *Networks* journal. As far as I know, this is his only paper in PODC, and he never attended PODC. He did attend at least one SIROCCO conference (1999), and one DISC conference (1990), at that time called WDAG: the 4th WDAG in Bari; it was then when we traveled together for the first time. He had a paper in SPAA'98, which was held jointly with PODC'98, "Layout Area of the Bitonic Sorting Network," with Muthukrishnan, Paterson and Sahinalp; he worked on interconnection networks and layouts of graphs until the end of his life publishing a number of papers in SPAA and other related conferences and journals, including six U.S. patents. The subject was invented

Figure 1: Shimon with his wife Tamar, the author and his wife Claudia traveling in la Dordogne.

in the PhD thesis of his student Yossi Shiloach in 1976, that models printed circuits and VLSI chips. Even closer to PODC was his work on systolic circuits, on information dissemination, and on computation in networks of automata. Or the paper on distributed algorithms to find sparse subnetworks with good connectivity, TCS 1998, together with Gene Itkis and myself. Also related to PODC are of course his various papers on security. Take for example "On the Security of Ping-Pong Protocols" with Dolev and Karp in CRYPTO'82, or "On-Line/Off-Line Digital Signatures" with Goldreich and Micali in CRYPTO'89. One of the key foundations of distributed computing is graph algorithms, the subject of his classic book, as well as of various important papers, such as "Computing an *st*-Numbering" with Tarjan in TCS 1976.

One of his most influential papers is "Marked Directed Graphs," in *J. Comput. Syst. Sci.*, 1971, with Commoner, Holt and Pnueli. This work is fundamental to the area of systolic circuits, and many other distributed systems that exhibit speed-independent behavior. It established the key properties of a model that has been central to the very active research community of Petri Nets, as well as to many real applications whose behavior is modeled by a marked graph. My own PhD work with Shimon showed that this is the case for the synchronizers of Awerbuch (or any synchronous algorithm running in an asynchronous environment). Another application that is modeled with marked graphs was studied by Barbosa and Gafni in "Concurrency in Heavily Loaded Neighborhood-Constrained Systems," TOPLAS 1989. Recently, in a paper in the *Distributed Computing* PODC 20th anniversary issue (Sept. 2003), Lamport tried to answer a basic question: what kind of multiprocess synchronization can be achieved in bounded time? Read/write registers require an arbiter (a device that makes a discrete decision based on a continuous range of inputs), and hence unbounded time. As a partial answer he characterized what can be solved with wait/signal registers, which are just one class of arbiter-free synchronization primitives, by finite-state deterministic processes: marked graphs. Marked graphs model systems where events are triggered by "and" conditions; they are a restricted form of Petri Nets. Formally, they consist of a directed graph where each arc contains zero or more *tokens.* A node can *fire* where each of its incoming arcs has at least one token. When it fires, one token is removed from each of its incoming arcs, and one is added to each of its outgoing arcs.

Today more than ever science preeminence for its intellectual virtues is threatened, not only by the humbug and credulousness of the layman, but from within, by a lack of rigor and commitment to the quest for truth among scholars. Shimon's way of life was exemplary in this sense, always committed to a deep respect for professionalism and truth.

2004 Gödel Prize

The 2004 Gödel Prize is great news for the distributed computing community: it is shared between two journal papers that discovered a fundamental relation between distributed computing and topology. Moreover, Elizabeth Borowsky and Eli Gafni, coauthored a third paper with the discovery, but published it only in a conference, STOC 1993, and hence it was not eligible for the prize. The original conference versions of the two celebrated papers appeared also in STOC 1993. To celebrate this, I am happy to present personal perspectives by the authors of these papers, and also by other people that produced seminal work on the same subject: Soma Chaudhuri and Shlomo Moran.

Another celebration of the Gödel prize will take place during a joint session of this year's DISC conference and GETCO workshop. GETCO 2004 is the 6th workshop of the series, and is devoted to topological methods in concurrency and distributed computing. It has been very successful at bringing together researchers from both the mathematical and computer scientific side, showing sustained activity mainly from concurrency theorists and semanticians, who also found connections between topology and distributed computing. It is remarkable that these connections were found (or at least started its modern development) in 1993 in a CONCUR paper by Eric Goubault—exactly the same year where the three papers of the Gödel prize where published in STOC, the two of the prize and the one by Borowsky and Gafni. See the "Historical Note on Geometry and Concurrency" by Eric in HTTP://WWW.DI.ENS.FR/˜GOUBAULT/INDEX1.HTML

The papers that have been awarded the Gödel Prize for the year 2004 are

- Maurice Herlihy and Nir Shavit, "The Topological Structure of Asynchronous Computability," JACM 46, 6 (1999).
- Michael Saks and Fotios Zaharoglou, "Wait-Free k-Set Agreement Is Impossible: the Topology of Public Knowledge," SIAM J. Computing 29, 5 (2000).

The following description is taken from the SIGACT pages. The Gödel Prize for outstanding papers in the area of theoretical computer science is sponsored jointly by the European Association for Theoretical Computer Science (EATCS) and the Special Interest Group on Algorithms and Computing Theory of the ACM (ACM-SIGACT). This award is presented annually, since 1993, with the presentation taking place alternately at ICALP and STOC. The Prize is named in honor of Kurt Gödel in recognition of his major contributions to mathematical logic and of his recently discovered interest in what has become the famous "P versus NP" question. The Prize includes an award of $5000. Any research paper or a series of papers published (not reprinted) in a recognized refereed journal by a single author or a team of authors in the seven years preceding the year of the award is deemed eligible. This extended period is in recognition of the fact that the value of fundamental work cannot always be immediately assessed. The research work nominated for the award should be in the area of theoretical computer science.

Roadmap. In Section 1 there is a description of the relation between distributed computing and topology discovered by the papers, as well as of its importance. The following sections contain the personal perspectives: Section 2 by Herlihy and Shavit, Section 3 by Saks and Zaharoglou, Section 4 by Borowsky and Gafni, Section 5 by Chaudhuri, and Section 6 by Moran.

1 Overview

Silvio Micali has given a very eloquent description of the importance of these two papers (and hence of the one by Borowsky and Gafni) in his nomination to the prize, on which this section is based. They presented the first example in computer science of the use of algebraic topology and in particular homology theory, in quantifying a computational phenomena.

Modern multiprocessors, whether they communicate by message-passing or through shared memory, are inherently asynchronous: processes can be halted or delayed without warning by cache misses, interrupts, or scheduler pre-emption. In an asynchronous world, it is of key importance to understand which tasks can be computed only by having processors wait for one another (which makes them susceptible to failures) and which can be solved in a wait-free manner (which makes them highly fault tolerant). The *asynchronous computability theorem* of Herlihy and Shavit provided the first complete characterization of the circumstances under which asynchronous tasks have wait-free protocols in shared read/write memory, and therefore, by way of known reductions, to many message passing models as well.

The key difficulty in reasoning about asynchronous systems is the "exponential blowup" in the number of possible executions resulting from different interleavings of processors' actions. The great breakthrough in these works is the introduction of algebraic topology as a tool for reasoning about the multitudes of possible executions in a comprehensible way. The surprising result is that it was revealed that being "asynchronously computable" is equivalent to having a collection of executions that, when viewed as a high dimensional geometric object, do not have holes. If holes exist, it follows from the Herlihy-Shavit theorem that the given task is not asynchronously computable.

A bit more technically, the authors introduce a new formalism for tasks, based on notions from classical algebraic and combinatorial topology. In this formalism a tasks is associated with a pair of high-dimensional geometric structures called simplicial complexes representing its possible inputs and outputs. The asynchronous computability theorem states that a task can be solved in a wait-free manner if and only if its input and output complexes can be mapped to one another by a simplicial map, a function with simple regularity properties that is the topological equivalent of saying that one complex can be deformed into the other by folding and stretching, but without introducing holes!

The power of the theorem is that it hides all the hard work of reasoning about concurrent executions, one need only look at a task's specification and verify if a simplicial map exits to know if a solution without waiting exits or not. Although the characterization is quite general, it has concrete applications. In particular, it yielded the first impossibility results for several long-standing open problems in distributed computing, including renaming and set agreement.

Topology is one of the youngest fields in mathematics, and is considered by many to be The Representative of modern mathematics. Unlike more established fields such as calculus, it has yet to find a "natural" area of application, one in which the formalism and tools it provides are fundamental and cannot be done without. The authors have provided the first such field, asynchronous computability, one in which topology, and specifically homology theory, are indispensable. Having read these papers, one might venture to say that the homology and continuity properties of asynchronous computations unfolding in time are not a man made formalism but a phenomena of nature that the authors have skillfully exposed.

2 Personal Perspective by Herlihy and Shavit

Many an enterprise has its origins in boredom. PODC 1992 was held in Montreal (during the Russian coup), and we (Maurice and Nir) drove there and back from Boston. On the way back, we were bored, and instead of playing twenty questions we decided to try and come up with an algorithm for three-process two-set agreement. The exercised passed the time during the long drive, and, not surprisingly in retrospect, we made little progress on an algorithm.

Intrigued and frustrated, we continued to work on 3-process 2-set agreement. We alternated our attention between algorithms and lower bounds. One approach to lower bounds that seemed particularly promising was the graph-theoretic approach pioneered by Biran, Moran, and Zaks (BMZ). Try as we might, however, we were unable to see how to generalize graphs, which inherently model pair-wise relations, to wait-free 3-process 2-set agreement, which seemed to require capturing more complex relation among multiple processes and values.

Maurice remembered that as an undergraduate Math concentrator, he had take several classes in Algebraic Topology, where the principal mathematical tool was the notion of a simplicial complex, a simple generalization of a graph used as a discrete analog to surfaces and other geometric objects.

Not long after shaking the dust off the old textbooks, it was clear to us that simplexes and simplicial complexes provided exactly the right elementary tools for modeling n-process decision tasks such as k-set agreement. Informally, a 2-dimensional simplicial complex is a geometrical figure constructed by gluing together a collection of vertices, edges, and solid triangles in a "nice" way.

Armed with this simple formalism, we were able to formulate *simplex agreement*, a simple and universal way of characterizing any one-shot decision task. Informally, simplex agreement is a kind of approximate agreement: each process places a token on a vertex of the complex, and the processes asynchronously move their tokens until they rest on a single vertex, a single edge, or a single triangle.

We also devised a simplex agreement algorithm that we thought for a while solved 3-process 2-set agreement. Eventually, however, we realized that our algorithm could fail to terminate in an execution where processes' tokens "chased" one another around a hole in the simplicial complex representing 2-set agreement. This observation suggested that high-order connectivity (that is, whether or not the complex had holes) was somehow related to solvability. Our basic insight, that wait-free read/write computations have "no holes", followed right away. (As always, the formal details took a little longer.)

In retrospect, a key step was our realization that we could use these elementary mathematical tools known to every undergraduate mathematics concentrator to provide a more natural way to model distributed computation than the graph-theoretic tools used by most researchers in our community. We think this story illustrates a "hole" in undergraduate Computer Science education. Everyone studies graphs, but few study equally important (and no more difficult) concepts such as simplexes and simplicial complexes associated with Combinatorial Topology. Perhaps this should change.

3 Personal Perspective by Saks and Fotios Zaharoglou

We learned of the problem from Soma Chaudhuri, who originally formulated the problem, and recognized its importance to understanding the power of fault tolerant algorithms. We tried for quite some time to find a wait-free algorithm for the 2-consensus problem for 3 processors, and eventually our failure convinced us that it was impossible. Our attempts to generalize the known impossibility proofs for 1-consensus fell short. Those proofs work by exploiting the uncertainty that processors have about the interleaving of processor steps, and and we hit on the idea of focusing on the structure of this uncertainty. This led us to the abstract problem: for which subsets of schedules is it possible to design a wait-free protocol which accepts precisely those schedules in the subset? The combinatorial structure of these sets, which we called “knowable”, was very interesting. We were able to reformulate the problem of designing an algorithm for wait-free k-consensus as the existence of a collection of knowable sets satisfying certain constraints. As detailed in section 3 of our paper, we observed a close syntactic analogy between this system of constraints, and the system of constraints on a collection of open subsets of $\mathcal{R}^n$ which was proved impossible to satisfy by Knaster, Kuratowski, and Mazurkiewicz. Formally, we could show that the set of knowable sets did define the open sets of a topological space, but we wanted to show a formal isomorphism between that topology and the topology of a simplex in $\mathcal{R}^n$. We tried unsuccessfully to find an explicit metric on the set of schedules that would demonstrate this isomorphism (we still wonder if there is such an explicit metric!). One proof of the KKM theorem is via Sperner’s combinatorial lemma (which had already appeared in Chaudhuri’s work on the k-consensus problem) and we were able to accomplish our goal by finding a Sperner-type lemma for the knowable set topology.

It’s very nice to have our work share the Gödel prize with that of Herlihy and Shavit. It is very unfortunate that the prize rules (restricting selection to journal papers) resulted in the exclusion of Borowsky and Gafni, who certainly share credit for these developments.

4 Personal Perspective by Borowsky and Gafni

In January of 1992 Eli was working on the problem of renaming. The standard renaming algorithm allowed n anonymous processors to rename themselves uniquely into a name-space of size $2n-1$. It occurred to Eli that if the n processors could split themselves into 2 non-empty sets, then one could use the standard renaming algorithm to rename into a name-space of $2n-2$ by allowing one group to rename from slot 1 upward, and the other to rename from slot $2n-2$ downward. It had been conjectured for years that renaming into this small range is impossible. With this he realized that he had reduced the impossibility of renaming into another problem.

Eli called Yehuda Afek about the reduction, and Yehuda brought to his attention the problem of k-set-consensus formulated by Soma Chaudhuri. Consensus (all processors agreeing on a single value) is impossible if even one processor might fail. Soma generalized the consensus problem and asked whether it was solvable if processors could narrow their outcomes to 2 values, or k values, instead of just one. She gave an algorithm to solve k-set consensus in the face of $k-1$ failures, and conjectured it could not be done in the face of k failures (a direct analog of the standard FLP impossibility result). At this same time, Elizabeth was looking for a research topic and together we started to work on proving set-consensus is impossible.

The first step we took was to concentrate on wait-free computation. Namely, to prove k-set-consensus is impossible with k failures, let us restrict the system to $k+1$ processors. Then perhaps by reduction we could use the impossibility of the wait-free case to imply the impossibility of the general case (n processors and at most k failures). Within two weeks we landed on the technique of such a reduction, which was later named the BG simulation.

The elegance of the BG simulation encouraged us that we were on the right track so we proceeded to work on the wait-free impossibility. Eli realized that by Koenig’s Lemma we could limit the problem further and concentrate solely on the case where each processor takes exactly m steps for some number m. With this, we began developing graphs of the full-information views of processors in m-step read-write computation - a structure that Herlihy and Shavit in their paper called correctly, *protocol-complex*.

Soma's work had raised the specter for using Sperner's Lemma to color subdivided simplexes, so we set to prove that there is a model of computation perhaps stronger than read-write which give rise to this structure. This led us (concurrently with Saks and Zaharouglo as it turns out) to the discovery of Immediate-Snapshots. The Immediate-Snapshot graphs looked like subdivided simplexes, but we had a hard time proving it. We had not at this time limited the model to Iterated Immediate Snapshots (motivated by Herlihy and Shavit work a few years later) which allow the simplex proof to go through.

Two weeks before the STOC93 deadline Elizabeth saw the light. She realized that Sperner's Lemma will go through even if the structure is a surface (or in topological parlance - manifold, as observed later by Attiya and Rajsbaum), and that was something we could prove. Sperner's Lemma worked, the $k+1$ processor wait-free protocol-complex could not be colored with fewer than $k+1$ colors, and thus there was some execution of the protocol where $k+1$ values must be chosen. The paper was complete, k-set consensus was impossible.

Of course we are disappointed that as a result of a lack of a journal version, we did not share in this prestigious prize, but the rules are the rules, and we have nobody to blame but ourselves. In a way, our proceeding version is very readable and consequently we did not feel the urgency to polish it. We would like to particularly acknowledge the influence of Soma Chaudhuri, for without her (with the involvement of Saks as we were told later) mentioning the idea of Sperner's Lemma in this context we would not have gone as far as we did. We are honored that work we shared in developing is so recognized.

5 Personal Perspective by Soma Chaudhuri

I was always intrigued by the Fischer Lynch Paterson impossibility result for consensus, even more so by the subtle technique involved in the proof. Consensus was a fundamental problem in distributed computation that was impossible even with a single failure; could I characterize problems that were 1-solvable but not 2-solvable (and, in general, k-solvable but not k+1-solvable)? An obvious approach seemed to be to try and generalize the consensus problem itself. While consensus requires all processors to decide on the same value, we could increase the number of choices allowed to the processors, by now requiring them to choose one of two different values. This was the 2-set consensus problem. It was easy to come up with an asynchronous algorithm that solved the 2-set consensus problem in the presence of one fault. This generalized to any k; k-set consensus was solvable in the presence of $k-1$ faults.

The next step was to come up with the lower bound; to prove that k-set consensus is impossible given k faults. I tried to follow the FLP approach, which proves the existence of an initial bivalent configuration, and then proves inductively that starting at a bivalent configuration, it is possible to take an infinite number of steps and still remain in a bivalent configuration.

I started by trying to prove that an initial 3-valent configuration (one from which 3 distinct values could be decided) exists for the 2-set consensus problem in a system of n processors and at most 2 faults. FLP used a similarity chain argument to prove the existence of an initial bivalent configuration given 1 fault. Each pair of adjacent configurations were similar given 1-fault runs, producing a 1-dimensional chain of edges. I noticed that, while analyzing the pairwise relation between configurations was sufficient for single failure runs, the correct analog for 2-fault runs was a 3-way relation, giving rise to a 2-dimensional geometric structure of unit triangles.

I constructed a triangle on a grid, where each grid point corresponded to an initial configuration, with the three corners representing configurations where all processors have the same initial value. Each unit triangle of three configurations should have some run that is common to all of them, which imposed a certain valency consistency requirement on each configuration. Using an inductive argument, I could prove that some unit triangle would have an inconsistent valency, thus proving the existence of a 3-valent configuration.

The proof seemed cumbersome though, and I didn't see how to easily generalize this to k faults. This seemed like a graph coloring problem and I asked Mike Saks if the problem and solution seemed familiar. It didn't take him long to tell me that it looked like I had proven the 2-dimensional version of Sperner's Lemma! He then pointed me to the general version of Sperner's Lemma and it became clear that, by studying a structure of k-dimensional simplices, I could prove the existence of an initial (k+1)-valent configuration for k-set consensus.

To continue with the inductive argument as in FLP, I tried to come up with a structure for the set of reachable configurations. Since FLP was still using the similarity chain argument in the inductive proof, though in a more subtle way, it seemed that I could do the same using the k-dimensional simplex structure. However, I was unsuccessful in doing this. In hindsight, I realized that I should have studied the wait-free version of the problem rather than the fault-tolerant version that I was concentrating on!

My partial result was published in PODC 90, and two years later, three groups of researchers (Saks & Zahoroglu, Herlihy & Shavit, and Borowsky & Gafni) were able to prove the impossibility result, using Sperner's Lemma and topological concepts. I was excited that my problem had got so much attention, even though I was not able to solve it myself. I am delighted to see that this work has now won the Gödel Prize!

6 Personal Perspective by Shlomo Moran

The influential paper of Fischer, Lynch and Paterson on the impossibility to reach consensus in the presence of one failure (1982), raised the following question: Is there a way to decide from the specification of a distributed task, if the task can be solved in the presence of one failure (or in short: if the task is "1-solvable")? My interest in this question goes back to 1984-5, when Yaron Wolfstahl, after giving a seminar presentation of FLP, had chosen it as the subject of his Master thesis.

Relation of "connectivity" to "1-solvability" was established by Yaron and me as follows: we modeled "distributed tasks" as multivalued input-output functions, which map each input vector (a possible initial assignment to the processors) to a set of possible output vectors. The "input graphs" and "output graph" are then obtained by adding edges between input (output) vectors which differ in exactly one entry (ie, are at Hamming distance 1). In such graph, $n - 1$ entries of each vector determine the connected component it belongs to. Using FLP, this was shown to imply, informally, that a task with connected input graph and disconnected output graph is not 1-solvable.

The main obstacle in extending the above to a precise characterization of 1-solvable tasks was that connectivity alone does not carry enough information on the task's input-output mapping. This was solved in 1987-8 with Ofer Biran (then a Master-transferred-to-direct-PhD student) and Shmuel Zaks, by replacing connectivity of the whole graph with connectivity of the image of each input vector (pointwise connectivity), and in addition requiring that for each input vector with one unspecified entry, there is a similar output vector, so that each completion of the missing entry in the input vector corresponds to a similar completion of the output vector.

As the resulting characterization was based on decidable properties of the input-output mapping, it provided a constructive way to decide whether a (finite) distributed task is 1-solvable. This had few nice applications— my favorite one is a universal fault-tolerant algorithm for 1-solvable tasks, which provided exponential speed up in the time complexity of some previously known algorithms.

The problem of recognizing "1-solvability" is naturally generalizes to recognizing "k-solvability," ie tasks solvable in the presence of k failures. As will be noted soon, we now know that k-solvability is an undecidable property even for finite tasks, for all k except $k = 1$. This is one good excuse why we were not able to extend our constructive characterization of 1-solvability to $k > 1$. And indeed a different technique was needed to go beyond 1-solvability— the "algebraic topology" technique that was discovered by Borowski and Gafni, Herlihy and Shavit and Saks and Zaharoglou . Apparently, a push towards this discovery was given by Soma Chaudhuri, who defined a prototype of a task which is not k-solvable— the "k-set consensus," which is a generalization of the original consensus task.

The "algebraic topology" approach in all three papers differed from our approach in few important aspects: first, it studied wait-free computations, in which all processors but one may fail (rather than concentrating on k-solvability for some fixed k). More significantly, it studied combinatorial properties of executions of unbounded length, as opposed to combinatorial properties of the input-output mapping alone, which were enough for 1-solvability. Apparently, it is this difference which enabled to extend the characterization to undecidable cases (at the unavoidable price of making it non-constructive). The most striking part of this approach is the sophisticated use of the rich theory on simplicial complexes: wait free computations were modeled as mappings from "execution complexes" (which are unbounded extensions – or subdivisions – of "input complexes") to "output complexes." Vertices in the execution complex presented local states of processors, and simplexes presented sets of local states which are realized by a single execution. Having this mapping in hand, an elegant use of Sperner's Lemma implied that k-set consensus is not k-solvable. As we all know, the resulting technique was immediately and widely recognized as a major breakthrough.

The "algebraic topology" approach had quite a few important applications. Arguably, the most surprising one is the above-mentioned undecidabilty of k-solvability for $k > 1$, proved by Gafni and Koutsoupias, using the characterization theorem of Herlihy and Shavit plus the undecidabilty of the word problem (proved in the 1950s). This makes the discovery of this approach a very appropriate candidate for, and a winner of, the Gödel prize. Congratulations to the winners - and of course also to Elizabeth and Eli, who for technical reasons were not nominated for the prize.