The Interview Column

by

Chen Avin and Stefan Schmid

Ben Gurion University, Israel and TU Berlin, Germany
{chenavin,schmiste}@gmail.com

KNOW THE PERSON BEHIND THE PAPERS

Today: Nancy Lynch

Bio: *Nancy Lynch has been a Professor in the EECS Department at MIT since 1982. She heads the Theory of Distributed Systems research group in MIT's Computer Science and Artificial Intelligence Lab. She received her B.S. from Brooklyn College in 1968 and her PhD from MIT in 1972, both in Math. Prior to joining MIT, she served on the Math faculties at Tufts and the University of Southern California, and on the CS faculty at Georgia Tech.*

Lynch has written and co-written hundreds of research articles about distributed algorithms and impossibility results, and about formal modeling and verification of distributed systems. Her best-known contribution is the 1982 "FLP" impossibility result for distributed consensus in the presence of process failures, with Fischer and Paterson, followed by a paper with Dwork and Stockmeyer on algorithms for reaching consensus under restricted failure assumptions. Other contributions include the I/O Automata system modeling frameworks, and more recent results on wireless network algorithms and biological distributed algorithms.

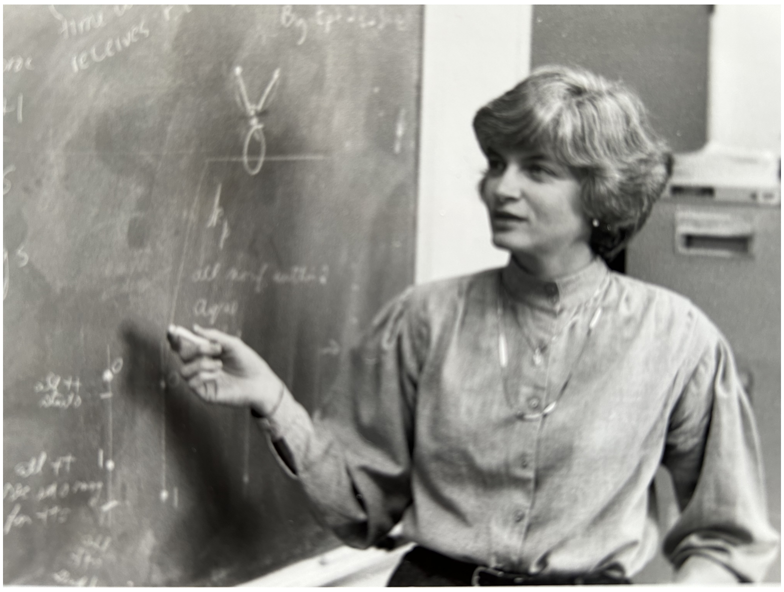
Lynch is the author of the textbook "Distributed Algorithms" and co-author of "Atomic Transactions" and "The Theory of Timed I/O Automata". She is an ACM Fellow, a Fellow of the American Academy of Arts and Sciences, and a member of both the National Academies of Sciences and Engineering. She has been awarded many prestigious prizes for her research contributions. She has supervised over 100 graduate students and postdocs, many of whom have themselves become leading researchers.

Can you give a quick summary of your main research contributions?

Nancy: Basically, starting in the late 1970s, working at Georgia Tech and MIT, I helped to develop the field of distributed computing theory. This included algorithms, impossibility results, and formal models and methods. You can read a detailed overview of much of this work in my recent paper "Building a Theory of Distributed Systems: Work by Nancy Lynch and Collaborators" [6].

We ask all interviewees to share a photo with us. Can you please tell us a little about the photo you shared?

Nancy: I found two. The first was taken early in my career, soon after I moved from Georgia Tech to MIT. So that would probably be around 1982. It shows me at the blackboard, explaining some type of argument about distributed consensus. The second was taken late in my career, during a 2019 visit to the Sorbonne, where I was honored with a "Doctorat Honoris Causa" degree.



Who were your main research mentors and how did they influence you?

Nancy: Albert Meyer, my thesis advisor at MIT, taught me how to do research, and in particular, taught me extremely high standards for selecting problems and carrying out technical work. Mike Fischer was my main collaborator during the late 1970s and 1980s, in my early work on distributed computing theory. I appreciate his great ideas and his great judgment about problems and approaches. Both Albert and Mike provided me with contacts in the field of theoretical computer science, helping me to get appointed to program committees, etc. Phil Enslow at Georgia Tech taught me about distributed systems, and suggested papers for me to read, especially those that might lead in theoretical directions. He also provided me with funding contacts.

Can you please tell us something about you that probably most of the readers of your papers don't know?

Nancy: I finished my PhD at the young age of 24, working in complexity theory. I didn't want to keep working in that area, and was not ready yet to choose a new area to commit to for a long time. So I explored a few things for several years.

Finally, at Georgia Tech, with Phil Enslow's influence, I got excited about something that seemed worth serious investment: theory for the new area of distributed systems.

Besides working, I spent a lot of time engaging in outdoor activities with my husband and children. This was mostly hiking and skiing in New Hampshire's White Mountains. We climbed all 48 4000-foot mountains in New Hampshire, which gave us entry to the Appalachian Mountain Club's "4000-Footer Club". We still go to the White Mountains frequently, now also with our grandchildren.

Is there a paper which influenced you particularly, and which you recommend for other community members to read?

Nancy: I was influenced by a 1975 report by Paul Johnson and Robert Thomas on "The Maintenance of Duplicate Databases." I found a version on-line [3]. Also by another paper around that same time by Armin Cremers and Tom Hibbard giving a lower bound for the space requirements needed for solving mutual exclusion—this was the first lower bound I had encountered in the area of distributed computing. Unfortunately, I can't find an on-line version of this—I knew it only as a University of Southern California technical report. The early papers by Lamport are worth reading—particularly the famous "Time, Clocks, and the Ordering of Events in a Distributed System" [5].

Is there a paper of your own that you would like to recommend for the readers to study? What is the story behind this paper?

Nancy: The Fischer, Lynch, Paterson paper on "Impossibility of Consensus in a Distributed System", commonly called "FLP", is my best-known and best-cited paper [2]. It's only 9 pages long, and easy to read. The "story" behind the paper is well summarized in [6]. Briefly, Mike Fischer and I knew that this was a problem of interest in the distributed systems community. We knew that we could formalize it in theoretical, algorithmic terms, and study it theoretically. Mike Paterson joined us in working out the proof. Actually, we went back and forth, trying to produce an algorithm and an impossibility result.

The FLP paper led to many related and follow-on papers. My most important follow-on paper was "Consensus in the Presence of Partial Synchrony", with Cynthia Dwork and Larry Stockmeyer [1]. This showed practical ways around the inherent limitation expressed by the FLP impossibility result.

Other worthwhile papers are my two I/O Automata papers with Mark Tuttle [7, 8]. The I/O Automata model is a useful, general abstract model for asynchronous distributed systems. It has good facilities for composition and abstraction. However, it wasn't general enough to model all interesting distributed systems. So Dilsun Kaynar, Roberto Segala, Frits Vaandrager, and I designed Timed I/O Automata [4], which included timing and continuous behavior. Now I don't think that we can find one general model that will be suitable for modeling all distributed systems. For example, in modeling biological systems (such as insect colonies or brain networks), we may want to make a variety of different assumptions about timing, continuous vs. discrete behavior, probability, composition, and abstraction.

You could also read the paper I mentioned above [6], which covers most of my research career.

When (or where) is your most productive working time (or place)?

Nancy: I work best at home, in the morning between around 8AM and noon. I never could work well in the evening, and the afternoons are OK but somewhat less productive.

I like to focus on difficult things for a few hours each day and then do something easier. For good advice on working productively, you might want to check out Cal Newport's books or website. (Cal was my PhD student.)

How do you choose what to work on? And what kind of impact are you hoping your work will have? Did this change over the course of your career?

Nancy: I basically work on what interests me. I try to choose problems whose solution could be important for science (usually computer science but recently biology). I work on problems for which I can see a path forward, that is, for which I think I have the right theoretical tools to make progress on them. I like to explore new application areas.

What do you do when you get stuck with a research problem? How do you deal with failures?

Nancy: Well, I flounder around, trying everything I can think of. I talk to other researchers about the problem and difficulties. If I still feel stuck, I put the problem aside and plan to come back to it when a new idea appears. Sometimes that never happens. Sometimes I decide it's not the right problem to work on.

Is there a nice anecdote from your career you like to share with our readers?

Nancy: I was a member of the famous "All-Women Program Committee" for STOC 1989. Christos Papadimitriou was the committee chair, and he decided to try to form a program committee consisting entirely of women. In those days, it was not so obvious that that was feasible, because there were not that many women working in theoretical computer science. There was some actual controversy about this effort, with arguments in Sigact News about whether such a committee would be sufficiently qualified to evaluate everyone's papers.

The members of the committee, in addition to Christos and myself, were Fan Chung, Cynthia Dwork, Faith Fich, Shafi Goldwasser, Debbie Joseph, Maria Klawe, Vijaya Ramachandran, Eva Tardos, Avi Wigderson, and Frances Yao. (Avi Wigderson was very enthusiastic about the project, and wanted to be included.) A very illustrious group of researchers.

Well, the committee did a great job, and the conference turned out great. I think it drew very favorable attention to the women in the field. There was one thing, though. I have never attended such an intense PC meeting! It seemed like everyone had read all of the submitted papers, and had very, very strong opinions about them. Discussions weren't cut off. The result was that the (one-day, in-person) meeting went on for more than 12 hours, running well past a normal dinner hour. It was exhausting!

Do you have any advice for young researchers? In what should they invest time, what should they avoid?

Nancy: My approach is to look for problems of practical significance for some important discipline, and then work on them using good theory.

What are the most important features you look for when searching for graduate students?

Nancy: My students have generally found me, so they already have read some of my papers and have some knowledge and interest in theory of distributed systems. Other than that, they should be very smart and hard-working, but that pretty much describes all MIT students. They should be friendly, cooperative, helpful to other students, so they contribute to a nice research group.

Do you see a main challenge or opportunity for theoretical computer scientists for the near future?

Nancy: Personally, I favor good theory applied to topics of practical importance, such as biological systems. For example, studying insect colony behavior can lead to theoretical study of swarms of robots and other coordinated activities. I am also interested in algorithms that explain how brains accomplish various tasks. Theoreticians should study these at a high level of abstraction, while still trying to capture essential properties of the behavior of actual brains.

Learning is another very important topic, though there is already quite of bit of theoretical study related to learning. Of this work, I prefer the study of learning algorithms that are like those used in actual brains, rather than those that arise in Artificial Neural Networks.

Please complete the following sentences?

- *Being a researcher...*

  Being a researcher is a great job. You have the freedom to think of new ideas and work them out. You get to work with students, postdocs, and other colleagues on research problems. You can also write books, travel and give talks, plan workshops and conferences...you have so many interesting opportunities.

- *... is key to being a happy academic.*

  The key to being a happy academic is not to get overwhelmed by all the interesting opportunities. Focus on one or two at a time. Also, as an academic, you are constantly presented with requests to do stuff. Be sure to say no to enough of these.

- *My first research discovery...*

  My first result was the main result in my thesis on abstract complexity theory. It showed the existence of problems that don't help in solving certain other problems. The argument was a complicated diagonalization construction. In distributed computing theory, my first results were algorithms and lower bounds on the space requirements for solving mutual exclusion, in a paper with Jim Burns, Mike Fischer, and others.

- *Theoretical computer science in 100 years from now...*

  I have no idea what it will be like. I have no idea what computer science will be like in 100 years. Neither does anyone else.

References

[1] C. Dwork, N. Lynch, and L. Stockmeyer. Consensus in the presence of partial synchrony. *J. ACM*, 35(2):288–323, Apr. 1988.

[2] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, Apr. 1985.

[3] P. Johnson and R. Thomas. The maintenance of duplicate databases, 1975.

[4] D. K. Kaynar, N. A. Lynch, R. Segala, and F. W. Vaandrager. *The Theory of Timed I/O Automata, Second Edition*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2010.

[5] L. Lamport. Time, clocks, and the ordering of events in a distributed system. *Commun. ACM*, 21(7):558–565, July 1978.

[6] N. Lynch. Building a theory of distributed systems: Work by Nancy Lynch and collaborators, 2025.

[7] N. A. Lynch and M. R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In F. B. Schneider, editor, *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, Vancouver, British Columbia, Canada, August 10-12, 1987*, pages 137–151. ACM, 1987.

[8] N. A. Lynch and M. R. Tuttle. An introduction to Input/Output Automata. *CWI-Quarterly*, 2(3):219–246, September 1989. Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands. Technical Memo MIT/LCS/TM-373, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, November 1988.