Laudatio on the Occasion of Nancy Lynch Receiving the Van Wijngaarden Award

Frits Vaandrager

CWI Soiree, Amsterdam, February 9, 2006

It is a great honor and also a great pleasure for me to speak about the achievements of Nancy Lynch at the occasion of the 60th anniversary of CWI and the presentation of the Van Wijngaarden Awards.

The mission of CWI — and Van Wijngaarden — has always been to do cutting-edge research in Mathematics and Computer Science aimed at solving practical problems in society. From the very beginning, this mission has been criticized by some: they argue that deep, cutting-edge results in Mathematics and Computer Science cannot be applied (at least not within decades) and are of little or no use for solving the urgent problems of society. To silence these critics it is good to point now and then to researchers who did cutting-edge research and applied their results in practice, living examples so to say in support of CWI's mission. Without doubt, Nancy Lynch is such an example.

I first heard about Nancy Lynch about 20 years ago when I was PhD student at CWI. At the lunch table my colleague Paul Vitanyi was telling enthousiastically about his sabbatical, which he spent at MIT in the Theory of Distributed Systems group of Nancy Lynch. According to Paul, there could be no doubt about Nancy being the smartest woman on earth. He had stories about her enormous productivity and measured her annual research production in terms of meters of publications. So when after my PhD — which I defended in this church in 1990 — I had the opportunity to become postdoc in Nancy's group, I was of course delighted. Since my experience with women is definitely much smaller than Paul's, I find it hard to assert for this scholarly audience that Nancy really is the smartest women on earth. But of course she is very smart. Nevertheless, even more than intelligence I believe an incredible amount of energy in combination with resolve are the key factors behind her impressive career as a scientist.

Just to illustrate her working energy: I vividly remember a skiing weekend with the Lynch family in the White Mountains of New Hampshire. For a full day, we skied downhill, and then going back up again, Nancy and I, sitting next to each other in the ski-lift, discussed our latest paper. Nancy brought a draft with her and was making notes. After alternating intense physical with mental labor for a day, I was exhausted. Nancy not. She was just having fun!

Impossibility Results

In his famous 36 paper, Alan Turing defined a computational model of what we now call Turing machines and proved that there are certain problems, such as whether a given computer program halts on a given input, that cannot be solved by any Turing machine. Turing formulated the concept of "computation" as we know today and his result marked the birth of Theoretical Computer Science. Although there certainly is a big difference in the nature and impact of the accomplishments, a compasison between Turing and Lynch makes sense since both defined a computational model and proved impossibility results that helped to shape a field.

In the days of Turing and Van Wijngaarden, a computer was a huge device, filling a large room and crunching numbers in isolation. Today, computers are connected by fiber networks, local area networks, wireless networks, etc and we seek ways to build and reason about these distributed computing systems. What Alan Turing did for Computer Science in general, Nancy Lynch did for the area of Distributed Computing through the discovery of some fundamental impossibility results. The most famous of these results is reported in a '87 paper with Fischer and Paterson. The result of this paper (commonly known as FLP) is that, surprisingly, it is impossible for a set of processors in an asynchronous distributed system to agree on a binary value, even if only a single processor is subject to an unannounced crash. Although the result was motivated by the problem of committing transactions in distributed database systems, the proof is sufficiently general that it directly implies the impossibility of a number of related problems, including consensus.

The FLP result has had a monumental impact in distributed computing, both theory and practice. Systems designers were motivated to clarify their claims concerning under what circumstances the systems work. On the theory side, people have attempted to get around the impossibility result by changing the system assumptions or the problem statement. The proof technique used in FLP, valency arguments, has been used and adapted (by Nancy and others) to show many other impossibility and lower bound results in distributed computing.

I/O Automata

Nancy Lynch has been instrumental in putting the distributed computing field on a formal foundation. Starting in '79 with work (with Fischer) on modelling asynchronous shared memory systems, in '87 she introduced (with Tuttle) I/O automata, a mathematical framework for modelling and analyzing distributed algorithms. A major obstacle to progress in the field of distributed computation was that many of the important algorithms, especially communications algorithms, seemed to be too complex for rigorous understanding. Although the designers of these algorithms were often able to convey the intuition underlying their algorithms, it was typically difficult to make this intuition formal and precise. In fact, many published algorithms in the area were flawed! What the I/O automata model provided was a way of formalizing the high-level ideas of designers, and incorporating them into a proof of the detailed algorithms correctness. It is nice to note that the first overview paper on I/O automata was published via CWI in the CWI-Quarterly in September '89. Since '87, research on I/O automata has continued. Together with co-workers (including Segala, Kaynar and myself), Nancy Lynch defined a series of state-machine models which provide a much-needed mathematical basis for the field of distributed computing. The original framework has been extended to allow for the modelling of real-time and probabilistic systems, and systems that involve both discrete and continuous components, like a computer controlled car. The models come equipped with a collection of simple and practical proof methods, mostly involving composition, invariants and abstraction.

Applications

The I/O automata modelling framework has been been used extensively in the description and proof of complex distributed algorithms and systems. Nancy Lynch herself has been pioneering the application of the theory she developed to practical distributed systems problems. For example, she has used the I/O automaton model as the foundation for a theory for database transaction concurrency control and recovery. She has used I/O automata and compositional methods to model, verify and analyze algorithms used in distributed operating systems – for example, the main algorithms used in the implementation of the Orca distributed shared memory system (a logical error was found). She has modelled and verified complex communication protocols, mobile wireless ad-hoc networks, pieces of automated transit systems, a traffic collision and avoidance system for airplanes, etc etc

Conclusions

The five minutes reserved for this laudatio are way too short to give a proper overview of Nancy's contributions. In particular, I have no time to discuss all the nice distributed algorithms Nancy and coworkers developed. People who are interested in that should read Nancy's much cited textbook on *Distributed Algorithms*. So let me conclude. Nancy Lynch has made vital contributions to the theory of distributed computing, including mathematical models and proof techniques, algorithms and impossibility results. She has pioneered the use of formal, mathematical models to address practical problems in this area. Her vision, in which mathematical modelling and analysis plays a key role in handling the complexity of distributed systems, fits seamlessly with the mission of CWI. And finally (less important maybe, but good to mention in a country where woman rarely choose to pursue a career in Computer Science) she is a role model showing how women can be highly successful in our field.