

KNOW THE PERSON BEHIND THE PAPERS

Today: Tom Leighton

Bio: *Tom Leighton is a theoretical computer scientist whose career has taken an unusual path from combinatorics, algorithms, graph embeddings, and parallel computing to building Internet infrastructure at Akamai. He continues to teach discrete mathematics at MIT while spending his professional life at Akamai. His trajectory offers a rare perspective on how ideas from theoretical computer science can migrate into systems operating at global scale.*



Figure 1: Tom Leighton during a recent visit to the Singapore Institute of Technology. Photo supplied for the interview.

Stefan Schmid: The photo you shared with us was taken recently in Singapore. What was the occasion?

Tom Leighton: It was at the Singapore Institute of Technology, where I gave a lecture about starting a company and told the story of Akamai and its origins at MIT. The audience was mainly computer science and engineering students. SIT

is very interested in entrepreneurial activities and in inspiring students to think about the possibility of starting a company.

I've done similar things at a few universities over the years. What is interesting is that the questions and the level of interest are actually quite similar around the world. At least in technology, students today tend to be interested in many of the same things wherever you go.

Stefan Schmid: Your career itself is an unusual combination of theoretical computer science and entrepreneurship. Do you still think of yourself as a researcher?

Tom Leighton: These days I spend my time working at Akamai, not in a research capacity. I still teach every other fall at MIT, where I give lectures in the discrete mathematics class taken by computer science students. But today I have a full-time — and then some — job at Akamai.

So the daily routine is very different from when I was doing research. But of course the background in theory never really goes away. It affects how you look at problems and how you think about possible solutions.

Stefan Schmid: Before we get to Akamai, what originally attracted you to theoretical computer science?

Tom Leighton: The early work on the theory of computation and algorithms was very influential for me, and so was combinatorial mathematics, particularly the Hungarian school around Erdős. I remember being fascinated by results such as the Ajtai–Komlós–Szemerédi sorting-network result and, more generally, by the use of probability to prove that combinatorial objects exist. Those were very cool ideas.

At the time, the STOC and FOCS communities were also a wonderful source of interesting problems. I came from more of a combinatorial background, so I naturally liked combinatorial problems, and later problems connected to parallel computing and algorithms.

Stefan Schmid: How did you decide which problems were worth working on?

Tom Leighton: Mostly I worked on problems that I liked and where I felt I might be able to make some headway. It helped if a problem had been open for a while and other people had looked at it — if you solve such a problem, people will recognize the contribution.

I also liked it when a problem had some potential relevance, even if the result itself was still theoretical. If you can explain why the question might matter in the real world, that is useful. It can also help when applying for grants or jobs: even as a theoretician, it is valuable to be able to have conversations with people working in other parts of computer science.

What is interesting in retrospect is that some of the topics I liked for theoretical reasons turned out to be very relevant indeed.

Stefan Schmid: Is that the bridge to Akamai? Which theoretical questions unexpectedly became practical ones?

Tom Leighton: Yes. I had worked on parallel-computing problems and questions such as how to move data efficiently in large networks, how to replicate it, and how to decide what should be stored where and when. Those are combinatorial questions on which you can make theoretical advances.

It turns out that they are also very relevant to the Internet. When we started Akamai, many of the fundamental problems were of exactly this flavor. The context had changed enormously, of course, but the way of thinking about them was familiar. My theoretical background was very helpful in making design decisions and in deciding which approaches were likely to scale.

Stefan Schmid: What surprised you most when moving from proving theorems about algorithms to building systems that actually have to work?

Tom Leighton: There is a big difference between proving theorems about algorithms and writing code for systems that really work. They are pretty different areas, and you quickly gain a lot of respect for what systems people do.

When proving a theorem, you normally have a simplified model, an abstraction that you can get your arms around. The real world is messier and more complicated, and the code has to work in that real world. In practice, you may therefore use heuristic approaches that you would not encounter in the same way in a theorem.

At the same time, theory gives you a good sense of what should scale and what kinds of approaches are promising. That was a big advantage for us. And when we started the company, we had some terrific developers — among the best MIT had to offer. Theoreticians normally do not spend all their time building production systems; that is a different skill set. Combining the two was very powerful.

Stefan Schmid: Can you give us an example where an elegant algorithm had to evolve once it met Internet scale?

Tom Leighton: One recurring problem is assigning a user to a server depending on the content the user is trying to retrieve. Over time we went through a series of algorithms for doing that.

Initially, some of the approaches involved max-flow and min-cost algorithms. But those became too expensive to scale. At one point we used a variation of stable marriage. We knew these algorithms collectively, so we could keep evolving the solution and trying better approaches.

But the criterion in the end is not whether you have proved a theorem about the algorithm. The question is: Does it really work in the real world? What performance do you get? How much time does it take? At Akamai we are constantly measuring how the content-delivery network performs and looking for ways to make it better. That is an ongoing process.

Stefan Schmid: So has your notion of a “good algorithm” changed?

Tom Leighton: The objectives are different. In research, a beautiful theorem can be the result. In a production system, the result has to work, and it has to keep working as the system and the environment change.

But I would not say the two worlds are opposed. Theoretical training can be extremely useful precisely because it gives you a disciplined way of thinking about scalability and about the structure of a problem. You then combine that with measurement, engineering experience, and sometimes heuristics to handle everything that the model left out.

Stefan Schmid: You have now seen both sides. What qualities did you look for in students, and what do you look for in people at a company?

Tom Leighton: For students, I looked for people who were smart, hard-working, and interested in the same kinds of problems I was interested in.

For employees, the task is different. Being smart and hard-working still matters, and so does wanting to solve hard problems. But teamwork becomes especially important. We are a business, so making the customer happy is also very important. You want people who can solve difficult problems together and who fit the culture of the organization.

Of course, you never really know in an interview how someone will work in a team. You usually find that out later. But you can talk to people about what they enjoy doing and sometimes get a sense of whether the environment will be a good fit.

Stefan Schmid: Has your own way of working changed over the years?

Tom Leighton: I used to work late. Then I think having kids ended that! I gradually became more of a morning person, and these days I would say the morning is my most productive time.

But I am also doing a very different kind of work now. I am not sitting down in the morning trying to prove theorems. So it is hard to compare directly with my research years.

Stefan Schmid: Looking at theoretical computer science today from your unusual vantage point, where do you see particularly interesting opportunities?

Tom Leighton: AI is obviously going to change a lot of things in practice. One interesting question is whether theoretical researchers can contribute on the algorithmic side to making AI computations more efficient.

A lot of the action is connected to numerical analysis and linear algebra, including very large matrix computations. AI is unusual from a traditional STOC- and-FOCS perspective because there often is not a proof that something works — empirically, it just seems to work. That makes the problem more challenging for a community that is used to proving theorems.

But perhaps one can isolate simpler problems, understand them rigorously, and find provable algorithmic improvements that turn out to matter in practice. There is an enormous amount of money, computing power, and energy being devoted to AI. If we can make those computations more efficient, that would be very valuable.

Stefan Schmid: Is AI already changing how work is done at Akamai?

Tom Leighton: Yes, all over the place. It is useful for creating and synthesizing documents, and it is useful for coding. Developers can do much more efficiently now. I expect it to be useful on the financial side as well. I would say it is helpful in most roles, and we are still just at the beginning.

Stefan Schmid: What about quantum computing? Is that another area where theoreticians should be looking for the right questions?

Tom Leighton: I think there is an interesting question there: Will quantum computing really be useful at scale, and are there many important things you can do with it besides factoring and the number-theoretic applications we already know about? It is a very interesting computational model, and perhaps someday large-scale quantum computers will be built. What else can we do with them?

At the same time, encryption protocols are being changed precisely so that they are not vulnerable to the known quantum attacks. At Akamai we support post-quantum cryptographic protocols, and some customers with particularly sensitive information are already converting because they do not want data collected today to be decrypted five or ten years from now if breakthroughs occur. Ultimately, I think everything will switch over.

That makes the broader algorithmic question about quantum computing even more interesting: what other applications will justify the model?

Stefan Schmid: You mentioned that you still teach discrete mathematics. If a student asked whether computer science — and theory in particular — is still a good choice today, what would you say?

Tom Leighton: I think theory is important. At the same time, understanding AI to the extent that we can is probably very important now, because it is going to

change a lot of things and consume a lot of resources. There is real value in finding ways to make it more efficient.

In technical positions at Akamai, people come from different backgrounds, but there is a lot of computer science and engineering. So yes, computer science remains very relevant.

Stefan Schmid: What advice would you give a young theoretical computer scientist who is tempted to step outside academia — perhaps even to start a company?

Tom Leighton: Keep an open mind. I remember hearing a Nobel laureate speak when I was probably a teenager. The advice was essentially: as you go through life, keep your eyes and ears open and your brain turned on, because you never know when you are going to encounter a situation or a problem where you can make a real positive difference.

I think that also connects to a growth mindset. Just because you could not solve something yesterday does not mean you will not solve it tomorrow. Keep after it and keep thinking about other ways to get there.

And do not be afraid to get out of your comfort zone and try something different. That is especially true if you are thinking about experimenting with entrepreneurship rather than following only the straight academic path. Research and entrepreneurship are different experiences, and some skills transfer while others do not, but both can be challenging and rewarding ways to make a difference.

Stefan Schmid: Before we finish, let us go much further back. What were the first research problems you tried to solve?

Tom Leighton: In high school I tried to prove Goldbach's conjecture and to prove that there are infinitely many twin primes. I did not succeed with either one!

I did write a very rudimentary computer model that led to predictions about how many twin primes there might be in an interval, but I never proved anything. I cannot really count that as a research discovery. Later I worked at the Bureau of Standards on some zero-one matrix problems, and some papers came out of that. But it was really in graduate school at MIT, when I started working on graph-embedding and parallel-computing problems, that I began proving real theorems.

Stefan Schmid: Is there one of those old problems that you still occasionally think about?

Tom Leighton: Yes. My thesis involved graph embedding. One version of the question is this: take an n -node planar graph of bounded degree, say degree three, and embed it in a grid. Vertices are mapped one-to-one to grid nodes, and graph edges are routed along paths in the grid. Two routed edges may cross, but they cannot share a grid edge. How large does the grid need to be?

Using the Lipton–Tarjan planar separator theorem, you can get an upper bound with an $O(n \log^2 n)$ -size grid. In my thesis I showed that some planar graphs require an $\Omega(n \log n)$ -size grid. So there is still a logarithmic gap between the bounds, and, as far as I know, nobody knows the right answer.

Those are the kinds of problems I liked to work on. Maybe it is not the most important problem to solve today, but it is still a beautiful question.

Stefan Schmid: Is there something readers of your old theory papers may not know about you today?

Tom Leighton: I like tennis. I play, although certainly not at the professional level! And if I happen to be in a city where there is a tennis tournament, I sometimes go and watch. Federer was always the family favorite. These days, probably Alcaraz.

Please complete the following sentences.

- *Being a researcher or an entrepreneur*
are both great experiences. They are different, and some skills from one transfer to the other, but both are challenging and rewarding ways to make a difference.
- *My first research discovery*
is hard to identify. In high school I tried Goldbach’s conjecture and the twin-prime conjecture, unsuccessfully. It was really in graduate school at MIT, working on graph embeddings and parallel computing, that I started proving real theorems.
- *The key to being a happy academic or entrepreneur*
is that you like what you are working on.
- *Computer science in 100 years*
I have no idea. Just look at the explosion of what has happened in the last few years with AI — and we are only at the beginning.

Stefan Schmid: Tom, many thanks for this conversation. Any final words for the Bulletin’s readers?

Tom Leighton: Just say hello to a lot of folks I haven’t seen in a long time.