PHYS.ORG

# The complexonaut

April 9 2014, by Larry Hardesty

Scott Aaronson, an associate professor of electrical engineering and computer science Credit: Bryce Vickmark

When he was in elementary school, Scott Aaronson, like many mathematically precocious kids of his generation, dreamed of making his own video games. He had only the foggiest notion of what that entailed, however.

"I could try to imagine making my own game—I could draw a picture of what it should look like—but how does it come to life?" Aaronson recalls. "Maybe there's some factory where they do all kinds of complicated machining to make Mario move around in the right way. Then a friend showed me this spaceship game that he had on his computer, and he said, 'Here's the code.' Well, what is this? Some kind of summary of the game? 'No, no, this is the game. If you change the code, the spaceship will do something different.'"

"I like to say that for me, this was like learning where babies came from," Aaronson adds. "It was a revelation. And I was incredibly upset at my parents that they hadn't told me earlier that this exists. Because I was already 11, and other kids had known programming since they were 8, and how would I ever catch up to them?"

As that anecdote attests, Aaronson was a young man in a hurry. Also at 11, he taught himself calculus, because he was intrigued by the mysterious symbols in a babysitter's calculus textbook. The next year, when Aaronson's father—a science writer turned public-relations executive—was transferred from Philadelphia to Hong Kong to spearhead a new marketing push by AT&T, Aaronson enrolled in an English-language school that offered him the opportunity to skip a grade and leap several years ahead in math.

When he returned to the United States as a high-school freshman, however, Aaronson chafed at what he saw as the constricting dogmas of public education, getting poor grades and butting heads with teachers. So he enrolled in a yearlong program for gifted high-school students at Clarkson University, and, that winter, applied to colleges. In what would have been his junior year of high school—and over his mother's objection that he'd have trouble fitting in socially—he entered Cornell University as a freshman.

## Theoretical attraction

Despite this accelerated trajectory, however, he never lost the sense that, as a programmer, he still lagged behind his peers. At Cornell, he was part of a team of undergraduates who wrote control algorithms for robots competing in the RoboCup robotic-soccer tournament. "We won for two years, not thanks to me at all," Aaronson says. "I loved the mathematical part, but when it comes to software development, when it comes to making your code work with other people's code, and documenting code, and meeting deadlines, other people were just going to be so much better at this than I was."

The summer before his year at Clarkson, Aaronson had attended a math camp in Seattle where he had learned about the P = NP problem—the central problem in computer science—from one of its most prominent theorists, Richard Karp. "P" is a set of problems that can be solved relatively quickly, and "NP" is a set of problems whose solutions can be verified relatively quickly. For many problems in NP, however—notably those known as "NP-complete"—finding solutions appears to be a prohibitively time-consuming task.

Most mathematicians believe that P does not equal NP—that being easy to verify doesn't make a problem easy to solve. But nobody's been able to prove it.

When he was working with the RoboCup team, Aaronson says, "someone would mention offhandedly that we want the goalie to be able to move this way, and I'd start thinking about whether that's NP-complete. And maybe two weeks later, I'd be able to prove that it's NP-complete, but by then no one cares, anyway. They've moved on to a different way of doing it."

Already intrigued by theoretical questions of computational complexity, Aaronson learned from a fellow Cornell student about Shor's algorithm, perhaps the most important theoretical result in quantum computing.

Quantum computers are devices, still largely hypothetical, that would harness the strange behavior of matter at extremely small scales to perform computations. Discovered by Peter Shor in 1994, Shor's algorithm is a quantum algorithm for factoring large numbers, one of the canonical NP problems that is easy to verify but apparently very hard to solve. Shockingly, Shor was able to show that for a quantum computer, solving the problem would be almost as easy as verifying it is for a classical computer.

"My first reaction was, 'OK, this is probably some obvious crap that is getting hyped by the media,'" Aaronson says. But he had to know for sure, and he threw himself into the study of quantum computing. He came away convinced that, indeed, quantum computers would rewrite the rules of computational efficiency.

## Quantum complex

The relationship between complexity—the classification of algorithms according to their execution time—and quantum physics has remained at the center of Aaronson's research since. He did his graduate work at the University of California at Berkeley so that he could study with Umesh Vazirani, one of the pioneers of quantum complexity theory. And now, as a tenured professor in the Department of Electrical Engineering and Computer Science at MIT, he finds himself a colleague of Shor, who, since the announcement of his algorithm, has joined the MIT mathematics faculty.

Aaronson believes that his own most important research includes his first paper on quantum complexity theory, written when he was a graduate student, which provided the first lower bound—minimum theoretically provable execution time—for a problem known as the "collision problem." Integrally connected to the cryptographically important question of cryptographic hashing, the collision problem asks whether a given mathematical function is one-to-one—every input produces a unique output—or two-to-one—every output can be produced by either of two inputs. Although subsequent researchers raised the bound, they used a variation on the same technique that Aaronson had developed.

Aaronson and Avi Wigderson of the Institute for Advanced Study in Princeton also proved that anyone hoping to answer the question of whether P = NP must first surmount an obstacle that they called "algebrization." "If you want to advance the field to where it could address the [P = NP] question, then you have to look at our current proof techniques and the barriers that are preventing them from getting us where we want," Aaronson says. "There were two previous times where we had to identify a barrier—the relativization and the natural-proofs barriers—in order to even start to think about what techniques were going to get around it." Algebrization is another such barrier. "Once you've clearly identified what the barrier is," Aaronson says, "then your mind is much freer to think about how to get around it."

Most recently, Aaronson and his student Alex Arkhipov described an optical experiment that, if performed successfully, could, for the first time, use quantum mechanics to execute a calculation that's infeasible with conventional computers.

As for whether Aaronson is more of a quantum-computing researcher or a computational-complexity researcher, he finds the question impossible to answer. "Often, even when I'm working on a purely classical question, it's a classical question inspired by something I'm trying to do in the quantum world," Aaronson says. "But then, with quite a few of the quantum-computing problems that I've worked on, it's ended up that the core of the difficulty was something in classical complexity theory. They're very, very linked."

*This story is republished courtesy of MIT News* *(web.mit.edu/newsoffice/), a popular site that covers news about MIT research, innovation and teaching.*

Provided by Massachusetts Institute of Technology

Citation: The complexonaut (2014, April 9) retrieved 12 May 2024 from https://phys.org/news/2014-04-complexonaut.html