Jennifer Tour Chayes: Basic Research, Hidden Returns >>>

Jennifer Tour Chayes

Interview of Jennifer Tour Chayes by Y.K. Leong (matlyk@nus.edu.sg)

> . . . *Bill Gates says research "is key to our long-term position."*
> - Dan Richman in *Seattle Post-Intelligencer*

Jennifer Tour Chayes has made important contributions to a newly emerging and rapidly growing multidisciplinary field that straddles mathematics, physics and theoretical computer science. Her current theoretical work on auction algorithms, self-engineered networks and phase transitions in combinatorics and computer science has found applications in the Internet and the computer industry.

After her BA in biology and physics, Chayes did a PhD in mathematical physics at Princeton. After some

Continued on page 22

Continued from page 21

postdoctoral work at Harvard and Cornell, she was all set for a distinguished career in academia at UCLA until one fateful day in 1996 when Nathan Myhrvold, then chief technological officer at Microsoft, approached Chayes and her husband Christian Borgs with an offer for them to join Microsoft Research. The rest, as they say, is history. Since then this famous husband and wife team co-founded and co-manages the Theory Group of Microsoft Research, one of the most active and vibrant groups of theoretical research in industry. In addition to the impact left by the collaborative work of Chayes and Borgs with others, the Theory Group has attracted many leading mathematical scientists as visitors, spawning fundamental research in a way that is rarely seen in industry. This unique phenomenon has been highlighted in a recent (March 2007) issue of *Scientific American*.

Chayes is probably the most striking counterexample to the myth that women is not cut out for science or that science has no place for women. Co-author of more than 80 research papers and co-inventor of 11 patents, she is the Research Area Manager for Mathematics and Theoretical Computer Science at Microsoft Research, Affiliate Professor of Mathematics and Physics at the University of Washington, a Fellow of the American Association for the Advancement of Science and a National Associate of the National Academies. She has also served as Chair of the Mathematics Section of the American Association for the Advancement of Science and as Vice-President of the American Mathematical Society. She serves on the Board of Trustees of the Mathematical Sciences Research Institute, the Scientific Boards of the Banff International Research Station and the Fields Institute, the Advisory Boards of the Center for Discrete Mathematics and Computer Science and the Miller Institute for Basic Research in Science, the Communications Advisory Committee of the National Academies, the Committee on Facilitating Interdisciplinary Research, the U.S. National Committee for Mathematics, the Association for Computing Machinery Advisory Committee on Women in Computing , the Leadership Advisory Council of the Anita Borg Institute and the International Union of Pure and Applied Physics Commission on Statistical Physics. Her capacity for research and organizational work is indeed legendary.

Chayes was invited by the Singapore Mathematical Society for its Distinguished Visitor Program in July 1999 and by IMS to give a public lecture at the Institute's program on *Random Graphs and Large-scale Real-world Networks* (1 May – 30 June 2006). During her short stay (7 – 16 June 2006) at the Institute, she was interviewed on 12 June 2006 by Y.K. Leong on behalf of *Imprints*. The following is an edited and unvetted version of this interview in which she talks with exuberance about her passion for science and mathematics, conveying forcefully the time-tested faith, if not axiomatic truth, in the inevitable and unstoppable benefits of basic research in mathematics and science.

Imprints: Your BA was in biology and physics and you waited till graduate school before deciding to specialize in one of them (mathematical physics). How difficult was it for you to make this decision?

Jennifer Tour Chayes: I have always liked many different sciences. I started out wanting to do biology, and then I did a little bit of physics – I love physics, so I decided to double-major in physics and biology. I also did a lot of chemistry as an undergraduate, one course short of a chemistry major. Mathematics was my hobby, I just enjoyed doing mathematics as science, but I didn't think of it as a profession. I thought it was fun to do mathematics. I suppose I was better in theory than in experiments, so it was probably a better idea to go into physics than into biology because at the time that I entered graduate school (in 1979) there was not a lot of theoretical biology. There was theoretical physics and there was mathematics. So I could do a lot of mathematics as well. One of the things I feel is that you don't have to make a decision to stop doing some subject in order to do another subject. I feel that I can still choose later in my career. I chose to do some computer science, and I keep thinking that maybe one day I will go back to biology. Now, more than 25 years later, there are a lot of interesting questions in theoretical biology – the field has matured so that there really is a vibrant field of theoretical biology. It has been impacted by mathematics, physics and computer science. It's always difficult for me to make a quick decision but I don't feel these decisions are permanent until you can do everything.

I: What was your area of research in your PhD thesis?

C: I proved theorems about several different systems in solid state physics. The questions were very mathematical, having to do with random statistics. A lot of what I did in graduate school, even what I still do, has to do with phase transitions – special points in a system where there is a qualitative change in what is going on in the system.

I: You applied these ideas to algorithms too. It's very surprising, isn't it?

C: Yes. Any system, when it is large enough, starts to exhibit a kind of average behavior. When I change my parameters in the system, the behavior of the system sometimes changes dramatically. That's a mathematical definition of what happens at a phase transition. The nature of algorithms changes very dramatically when you change certain parameters. A system can go from being solvable (a very efficient algorithm) to not being solvable in a short period of time. So I find phase transitions in algorithms also. At

Continued on page 23

Continued from page 22

first, I was a little surprised, but I was also very excited by the connection because when I saw the connection, I had already been working on phase transitions in physical systems for 15 to 20 years. It was very exciting to me that some of the phenomena that I understood very well were manifesting themselves in very different applications.

I: Were you first attracted to problems in theoretical computer science through the mathematics or was it the other way around?

C: If it has to be one or the other, I suppose it was the mathematics first and then the theoretical computer science but, in fact, it was the physics first, and then the mathematics, and then the theoretical computer science. Finding systems that have very interesting phase transitions, I was picking them up because I love phase transitions. They seem to have some new applications for theoretical computer science.

I: You taught in the universities before joining Microsoft. Did you experience any kind of "culture shock" in this career transition?

C: Yes, I suppose you could call it "culture shock". I think it's good to experience culture shock … it was very different from the university. Things happen on a much faster time scale. One day the company is interested in one thing, and then the Internet comes along and we shift. It's a much faster time scale than that of mathematical physics. Also there are people who really care about tearing the door apart … I find all of these very, very interesting. I could choose to participate in it or I could have a more academic group at Microsoft. I feel that I got the best of both worlds. Actually when I first told my colleagues from academia that I was leaving academia to go to Microsoft, everyone of them thought I was crazy. Now many of them think that I am very lucky, but at that time almost all of them thought I was crazy because they didn't believe that I could continue to do fundamental research. But Microsoft is very interested in fundamental research. Last week, we were giving a presentation to Bill Gates on some of our research. He was very interested in the mathematical details and he asked all kinds of questions about the mathematical details. I think there's real benefit for a company to have fundamental research because you never know what is going to be important.

I: Was there any time frame for a product or objective?

C: No, we are a very theoretical group. Microsoft has a huge development organization. There are thousands and thousands of developers. They are the ones who worry about the product time scale. In research, we worry much more about trying to expand the horizons to see where the world is going to be 10 years from now, 50 years from now. I don't think it makes sense for a company to try to have its research organization compete with its development organization. We don't have a pressure to do anything on a product cycle time scale. But sometimes, once in a while, we do get things into products, and I find that exciting too.

I: If it's not considered confidential, could you tell us a little about how you came to be involved in the founding of Microsoft's Theory Group and something about its structure – for example, is it localized in one particular place? Are there many permanent members and so on?

C: The way it started at Microsoft is that I was doing what I thought was very theoretical research on phase transitions and computer science. I told the chief technology officer at Microsoft about this research. He was a classmate of mine at Princeton when I was getting my PhD and he actually did his PhD in quantum gravity, which is much more theoretical than anything I did. But he left quantum gravity and did more classical things. I was telling this to him and he was saying to me, "Oh, you should come to Microsoft. You should do this at Microsoft." And I said, "Oh, that makes no sense." Then he kept encouraging me. Finally, my husband Christian Borgs, who is also a mathematical physicist, and I looked at Microsoft and we thought that this was a company that did care about fundamental research, even though at that time there was no research lab there. We believed them when we looked at some of the other research that was being done there. The only thing was that they thought we would have a very small group and we thought we would have a larger group. So there was some talking to do to make sure that we would have a group large enough to cover mathematics and physics effectively. This was started in 1997.

I: Like Bell Labs – they have fundamental research labs too.

C: Actually, at the time that Bell Labs was getting less fundamental in research, Microsoft was becoming more fundamental. The structure of the group … we have a relatively small number of permanent members (10). We have 8 postdocs who stay for a period of about 2 years. We have about half a dozen long-term visitors who stay anywhere from a few months to 2 years. Our visiting professors may come and spend a year or two years. Just like the IMS here, we have many short-term visitors (about 200 short-term visitors per year) – people who stay from one day to one month. We don't have workshops but we thought that if we were going to cover mathematics, physics and theoretical computer science and not hire hundreds of people, the best thing to do would be to bring in a lot of visitors and talk to them, do research with them and tell

Continued on page 24

Continued from page 23

them to send their students to us. We have a lot of summer interns also. So it feels more like an institute than a normal research group.

I: Is there a place where the whole group is stationed?

C: We are basically stationed in Redmond. When I first went to Microsoft, we only had research in Redmond, the company's headquarters. After I was there for a few years, Microsoft opened up a few other research labs. There's a lab in Cambridge, England and in Silicon Valley. There's now a large lab in China. But the vast majority of the research is done in Redmond. Bill Gates feels it's better to have most of the researchers there so that they can interact more with the policy people and with him.

I: Can anybody apply to visit for a long term?

C: People can apply, but we don't have so many long-term visitors. If you are working with someone at Microsoft, it's more likely that you will get long-term visitor status, or if someone at Microsoft is very interested in what you are doing. Other people often come for short-term visits and if we find that there is common interest, then they come back for longer-term visits. Microsoft funds the visitors. Visitors are paid for various reasons. One of them is that some of our visitors have come up with some very valuable intellectual property for Microsoft. We find that discussing a problem – even a theoretical problem – with someone may turn out to have applications for Microsoft and if we weren't paying them, we wouldn't have a right to that idea.

I: What happens if a person develops an idea while that person is at Microsoft but doesn't fully develop it until the person has left Microsoft.

C: Well, while they are at Microsoft when they develop a valuable idea, then we can file a patent with them on the basis of what they did at Microsoft. Patents don't have to be on fully blown ideas. A patent is usually less than a paper. In an academic paper, you try to work everything out. In a patent, even if you have an idea but you haven't worked out everything, you can still get a patent for it. Now if someone starts something at Microsoft and we feel that it's very interesting, we will sometimes ask them if they would like to stay under a contract with us, maybe one day a week they develop that idea even when they go home, and we pay them for that.

I: Are you talking about patenting of ideas? That's unusual.

C: Well, you patent algorithms but algorithms are really just ideas on how to do something.

I: There's no hardware involved?

C: No, there's no hardware involved. Now, I was surprised as a mathematician, a basic scientist, when I did my first patent. You probably heard stories of patents that have very little in them, like the "one-click patent" at Amazon – people always use that as an example. But there are more substantial ideas than that even though not every detail is thought out. In fact, when an idea is very broad, it's often more valuable. Ideas in their early stages are more valuable because they are broad and then little pieces can be patented as refinements of that. For many years at Microsoft, I patented almost nothing. In my first 8 years, I think I did very little patents. Now in the last year, I have done 12 patents because I happened to be working on something that has a lot of applications for Microsoft. We look at every idea and ask, "Does it make sense to patent it to the extent it is involved?" If we think that it might be used in a Microsoft product, then we just protect ourselves with a patent.

I: Does it mean that if you have patented an idea, you may be constrained not to reveal the details when you write a paper about it?

C: Not at all. That's why you should patent it. Once you have patented it, you can tell it to the whole world because then you own the rights to it. Different companies have different ways of dealing with it. There has been a lot of criticism against certain technology companies because they don't patent. They just keep secrets. That's very hard on their scientists because then their scientists are not able to publish and not able to talk to people and not able to be scientists. For us, we make the decision. We look at something and each individual makes his own decision. No boss ever tells them. If you think this is useful for the company, then you patent it. Sometimes the day before I submit a paper, I would give something to the lawyer and say, "File a patent on this before tomorrow because I'm submitting this paper to a workshop tomorrow." This allows you to pass it to anybody because your rights to that idea are protected. In fact, it gives you much more freedom than being secretive about it.

I: Have any of your patents brought in any personal wealth?

C: It's hard to tell what the direct relationship is. Most of the patents I have done are very recent and some of them have to do with new ideas on the web. I think that some of those ideas are valuable to the company. It surprises me. I didn't think I was going to do math that is going to be passed onto the bottom line. It shows that it makes sense for a company to have a basic research outfit because you don't know what's going to be important. It turns out that

Continued on page 25

Continued from page 24

algorithms are important and mathematicians are good at doing algorithms.

I: So it benefits the company more than you personally? The company has the rights to the patent.

C: I feel that it's fair. I don't teach. I get to travel. I'm well-compensated. I have freedom to invite collaborators. For me it's a very good trade off. I love my life. I'm happy that I'm able to do something that is worthwhile to the company to justify the expenses.

I: You mentioned that there are only 10 members. Are they mostly mathematicians or physicists?

C: Mostly mathematicians and theoretical computer scientists.

I: What about logicians?

C: We don't have any logicians, but we do have some combinatorialists. Certain parts of combinatorics are very close to logic. In Microsoft Research, there is one group that was started by someone who does logic. He's now doing other things but he was a logician. He's James Gorbit. He came from the University of Michigan and he started a group on abstract state machines but he did logic for many years. So there were people who did logic at Microsoft.

I: Any plans to get a logician into your group?

C: We try to get smart people into our group. If there's a brilliant logician, then we'll hire a logician. If there's a brilliant topologist, then we'll hire a topologist. I think it's much more who the person is, the quality of their work, rather than the subject because what people work on changes.

I: Does the Theory Group select only problems that are immediately relevant to computer software and technology?

C: No, absolutely not. We do basic research just like what you would do in a math department or a computer science department or a physics department. Sometimes we would talk to people in products and if the problems that they have are interesting mathematically, then we will look at those problems. We are really motivated by basic research.

I: Is collaboration more important or is the individual encouraged to work freely according to his or her own interests?

C: Definitely we want everybody to follow their own interests. On the other hand, we really do like collaboration. So you might have a couple of people going off doing something that doesn't have anything to do with what everybody else is doing. But when we hire people, we try to hire people who like to collaborate because we feel that there is a lot to be gained by collaboration. Also, we have many visitors who take advantage of our visitor programs and we expect people to collaborate a lot. I think that in trying to cover so many different fields with a small number of people, it's important to have people who like to collaborate because they can then bridge the gap.

I: Which do you think is more decisive for advances in computing – a conceptual revolution in theoretical computer science or a technological revolution in computer hardware?

C: I think they actually go hand in hand. As in many other sciences, when you work in experimental sciences, you see that there is an advance in experiments and then there is an advance in theory and then there is advance in experiments, and they go hand in hand. You find the theoreticians being inspired by the changes in the hardware and the people who build hardware inspired by the software revolution. Something has come up in hardware now – Intel and some other companies have produced the so-called "multi-core" chips. In such a chip there is a potential for parallel computation. That requires a true revolution in software. Intel was really like Microsoft in coming up with new software for multi-core chips so that people will want to buy those chips. Here is a hardware revolution and now we have several groups at Microsoft trying to figure out how to use these multi-core chips. There are a lot of very interesting theoretical problems and they ask people from the Theory Group to come to talk to them. Also there are other changes that are brought on by changes in software. A lot of the revolutions we have seen in computer science recently have been done by theory people who work at search engines. The two young guys who started Google were theory students at Stanford and they came up with the first algorithm for search engines. If you look at the whole field of web hosting, which is how to deliver content rapidly and is very important to the web, the web would not be as big as it is were it not for web hosting. Size would be going down left and right whenever people try to log on to them. Akamai, the biggest web hosting company in the world, was formed by a theoretician Tom Leyten and his students. These were revolutions in theory and software, and made hardware follow along. We are building this whole structure of the Internet on the web because there were some software ideas.

I: You mentioned the chip by Intel. That is a technical achievement. Was it necessary to do that? That's just making it smaller, isn't it?

Continued on page 26

Continued from page 25

C: Well, it's not just making it smaller. Having many processors on one chip, the multi-core chip is qualitatively different from the old chips. Within each multi-core chip, you can do parallel computation. It requires a completely different kind of software on a machine language level. It is a real revolution, and I think that as we learn how to take advantage of that, we will find many incredibly new applications, just like now we learn to take advantage of increased inexpensive storage. We come up with voice applications and video applications to take advantage of Moore's Law – the increase in storage and increase in computability.

I: Do you think there is an intrinsic limit to computing power, either theoretically or technologically.

C: It would be interesting if there were a theoretical limit, something like a Heisenberg uncertainty principle for computation. It is certainly true that we will never be able to stay on more bits of information than the number of atoms in our universe. At a certain point we are going to be limited at the atomic scale. If you try to think of something along the Heisenberg uncertainty principle, you might think of the limitation in speed. However, with parallel computation, which is so much faster, and quantum computation which is a kind of parallelism, I'm not sure if there is an intrinsic limit beyond the atomic scale. That's a very interesting question.

I: What about quantum computers?

C: We actually had for many years in our group some people working on quantum computing. They have now spread out to form their own larger group. I think it's a very interesting idea. The error-correction aspect of quantum computing is the most challenging aspect. Mike Friedman, who is within our group for many years and has now formed his own quantum computing group, is working on a different model for quantum computation in which you build the physical system so that it doesn't generate errors and so that it automatically corrects for errors. He's working with experimental physicists who are trying to build these things. Nanoscience in computing is also very fascinating. There are a lot of experimental advances in nanoscience and theoretical advances in quantum computation that will help us with our computing power in the future.

I: Do you know whether there is anyone who has built a prototype of a quantum computer?

C: I know that there are some quantum gates that people have made, but unfortunately those are the ones in which errors have to be corrected in the gates. So they have very limited power at the moment.

I: So the quantum computer is more like a dream rather than a reality.

C: Quantum cryptography, I think, will be used before quantum computers. It is actually a rather promising method of cryptography.

I: What about biological types of computers? Has anybody come up with anything like that?

C: Using DNA and things like that? There are a number of people working on them. I sense more excitement about them a few years ago than I sense now. I think that there are some limitations to those things. I think they can help us possibly in the next few generations of micros. Beyond that we need something more than biological computation.

I: You are also working on auction theory. Are you more concerned with the optimal algorithms for auction strategies rather than with auction theory per se?

C: I'm working on algorithms for auctions and for game theory in general. I'm looking at algorithmic game theory. It's a very interesting field and that's the field in which I've been filing a lot of patents. There's a lot of interest to Microsoft – very much an area in which we're competing with Google and Yahoo and some other companies. I think there are fascinating questions there. You have to come up with methods for dealing with auctions very quickly. Whenever you put a search term into a search engine, there is an auction that takes place in a millisecond. You don't even notice the time, but all of these ads that appear on the web site of the search engine are a result of an auction having taken place when you enter that term. So you need very efficient auction algorithms. Our group came up with some methods that help to prevent click fraud by coming up with algorithms in which we understand what the incentives are to commit click fraud and getting rid of those incentives.

I: What is click fraud?

C: Click fraud occurs when, for example, you are one book seller and I am another book seller, and we are both putting ads on a search engine under the term "book". Now you don't pay for your ads unless somebody clicks on your ads, and I don't pay for mine unless somebody clicks on mine. So if I go and click on your ad, then it costs you money. But I'm not a real buyer or a potential buyer. I'm just trying to run my competitor out of business. There are a lot of problems with click fraud, but we came up with certain algorithms which get rid of a lot of click frauds. There are a lot of interesting problems like puzzles, and it's really a lot of fun. In fact, just last week we were showing some of

Continued on page 27

Continued from page 26

that to Bill Gates. It's of interest to Microsoft and it's also nice mathematics.

I: Does this type of problems generate very theoretical mathematics?

C: Yes. In auction theory, if you try to auction different items to different people who want different bundles of items, those kinds of auctions are very complicated mathematically. They are called combinatorial auctions, and the number of combinations blow up very rapidly. There are fascinating deep theoretical questions – very difficult mathematical problems, NP-hard to approximate. We were also showing Bill some of the answers to those problems, which are still at the theoretical level, but it is important that we try to understand them.

I: What about applying those things that you are doing to economics?

C: Yes, we are working economics into all of this. We are doing algorithmic game theory, which brings together computer science and economics. Actually, we have had several good economists as visitors. We have been talking to them a great deal because I think there is very interesting mathematics there.

I: Have you ever thought of going back to biology?

C: I have thought about it actually. I talked to some biologists about it. There are several areas, all kinds of things in network theory, pathways to various enzymes that are close to the network questions that I'm working on in the context of the Internet and the World Wide Web. The state of diseases is certainly a biological question. There are all kinds of fascinating data-mining questions when you look at the genome. If we could use those data more efficiently, there is no question we would have cures for a lot of the diseases that plague us. I'm definitely thinking about that. I certainly want to go back to biology before my career ends.

I: One final mundane question. Did you face any kind of barriers generated implicitly by the "traditional gender mind-set" when you first joined academia or industry?

C: The first thing is that I try to ignore it. I try not to pay attention to that. I think I became more and more aware of them when I began to have students and when I began to have to make important judgment, because then I study the facts of the barriers in other people, for other people. I think there are two types of barriers: one is that there are a few people, not too many, who don't think that women are cut out for science. The president of Harvard made some very incorrect and politically stupid comments about that. I would have a pretty easy time dealing with that because if somebody thinks I'm stupid, I very quickly show them that I'm not stupid. If someone thinks you are stupid and you are not, it makes them very foolish. That is very easy to take care of. You just do very good work and no one can question that. There is another aspect of it, which is the leadership aspect. Are people comfortable with women leaders? I think that that happens in every male-dominated field. Over the years you have very confident women who take on leadership roles. After that happens, then you see changes. The changes are brought about by individuals who go in there and do such a good job that it's a moot point.