Math is a Discipline to Explain Knowledge in the Simplest Ways

Dr. Mu, Shin-Cheng
Assistant Research Fellow

Please briefly describe your background and factors triggering your interest in learning?

In the field of computer science, my interest is “programming language.” When I introduce my work to outsiders, I often described it as “computer science with heavier emphasis on mathematics.” Though, during my pre-college education, I wasn’t particularly good at math.

Since I was a kid, I started to use computer and write programs. It was after many years when I learned that during that period, gaming industry was strictly forbidden in Taiwan. Consequently, many factories began to manufacture Apple II clones for survival before switching to the production of PC clones later on. My parents bought one Apple II clone when I was ten years old. At the time, it was easy for people to describe the functions of a TV, a stereo equipment, or an air conditioner. However, few people could clearly tell the works a computer could achieve, especially in view of its limited functions. Like most kids, I only used the computer as a gaming device. Programs at that time were recorded in cassettes, which had to be loaded into a computer through a tape player. After I got tired of playing games, I would read books and magazines that came with the computer, learning to write programs. I started from simple ASCII graphics and small games, such as horse racing. Since I only recognized English letters, instead of any words, I memorized the commands letter by letter. I had to retype the programs every time when I turned on the machine, since I didn’t know how to store the programs into recording tapes. Gradually, structures of the programs became clearer in my mind and I noticed myself capable of making the programs shorter and shorter. In retrospect, I see it as a process of understanding the programs in a more abstract level.

Then, I became interested in programming language. I thought the beauty of the discipline lies in understanding a problem in an abstract manner and writing it down in proper notation, thereby producing clear solutions. The discipline tells us the importance of abstract understanding and good notation.

When I was in junior and senior high, I spent much time in Kuanghua marketplace and became a member of an underground store then, because the owner pledged to give me a copy of a LISP compiler. It turned out I was the first member of the store, which later on enjoyed prosperous business. Kuanghua Marketplace was an important channel for obtaining various kinds of software, as it housed many illegal counterfeit software firms, whose customers included students, aficionados, and foreigners. I bought the book “Inside Macintosh,” in three thick volumes, and read them by using dictionary to check vocabulary, learning...
how to write programs on Macintosh computer. Several years later, there emerged a computer craze in Taiwan and parents rushed to send their children to learn computer. At that time, my family worried that I was fascinated by computer, which could affect my performance in college entrance examination. My mother asked the boss of the computer store to advise me not spending too much time in computer. One day, he seriously told me that "I suffered a lot from low educational degree. You have to take good care of your school work."

Your research interest is Programming Language, which is not exactly a popular field in Taiwan. What makes you decide to choose this as your research field? Is there any scientist that you look up to?

After becoming a computer engineering major, those "computer books" I once had to hide and read secretly now became my homework. That was one major change. Computer Engineering was a popular discipline, but I chose a field that not many people were interested in. It might also be the reason why I later became Prof. Richard Bird's student, since he couldn't find many people who were interested in program derivation. Despite what others say, I stubbornly stayed in this field and become one of those with high educational degree. It is strange how life works; though I have chosen an unpopular field, my interest was categorized under a popular industry. Sometimes I felt it unfair for my friends, who like me stubbornly stayed in their field but do not enjoy as much resource as I do. That makes their courage all the more admirable.

It was by accident that I came by a book on functional programming. Then someone recommended to me Introduction to "Functional Programming" by Richard Bird and Philip Wadler. I found "this is exactly what I was looking for," but none of my classmates could understand what I was talking about. One day, I discovered an advertisement looking for a research assistant by Dr. Tyng-Ruey Chuang, and one of the requirements was ability for functional programming. I was so excited that I ran by leaps and bounds to my dorm and couldn't wait contacting Dr. Chuang for the position. That year, I started at IIS, Academia Sinica as an intern. Everything was so fresh and new. What I remember most is one occasion when Dr. Chuang reproved me. I was trying to explain to him an idea in front of the whiteboard and then he said that "You might know it, but you have to explain it in an understandable way for others." That taught me that only when you can explain an idea clearly, you truly understand the meaning behind it. It is similar to write a clear program.

After deciding what I was interested in, the road lying ahead of me also became much narrower. When most of my classmate became computer engineers, I needed to study aboard to pursue my interest. A year before applying for school, I read "Algebra of Programming" by Bird and Oege de Moor. I couldn't imagine that one could approach programming in such a way. I thought it would be amazing if I could learn it. But could I make it? My math wasn't that good. Anyway, I sent my application and was lucky enough to become one of his students. During those years at Oxford, I was spoiled in academic study completely. Although we all had different topics, we were all speaking the same language. Math, at least the part that we all needed, is not that inaccessible. Math is a discipline that tries to explain knowledge in the simplest ways. I had only realized that till college. It seems that all my math education below high school is a waste. It's a pity that by the time I graduated, only a few people were interested in "program derivation."

"It's your turn to carry the torch," they said.

Richard Bird is the ideal scholar in
Great Idea

To promote the development and use of open source software, Open Source Software Foundry (OSSF) provides a forum that facilitates interaction and collaboration between open source software developers, and serves as a bridge to link developers with industry and academia.

tel.: (02) 2788-3799 ext. 1469, 1477,
e-mail: contact@openfoundry.org
Web site: www.openfoundry.org

Blender

Blender is a 3D graphics application released as free software under the GNU General Public License. It can be used for modeling, UV unwrapping, texturing, rigging, water simulation, skinning, animating, rendering, particle and other simulations, non-linear editing, composing, and creating interactive 3D applications, including games.
Blender is available for a number of operating systems, including Linux, Mac OS X, and Microsoft Windows.

Blender is a 3D graphics application released as free software under the GNU General Public License. It can be used for modeling, UV unwrapping, texturing, rigging, water simulation, skinning, animating, rendering, particle and other simulations, non-linear editing, composing, and creating interactive 3D applications, including games. Blender is available for a number of operating systems, including Linux, Mac OS X, and Microsoft Windows.

Great Idea

my mind. His speech was powerful and clear. Even at the level of full professor and at the age close to retirement, he still wrote his own program. He could solve my technical issues faster them me. Oftentimes I would make up a solution, which always made him frown, and he would redo the problem from scratch, organizing everything in a clear and graceful way. Unlike most full professor in Asia who doesn’t do their own research anymore, he is my idol. I wish that one day when I am at that level, I would still work on my own research. It was years later when I learned the truth, that it was due to lack of funding and resource that forced Bird to do everything by himself.

My other idol is Edsgar Dijkstra. His paper is direct to the point and such a pleasure to read. He makes you believe that you need computational science for the inspiration of mathematic. Like his paper, he is a passionate and persevering person, and has a strong sense of right and wrong, which may be his major drawback too. A scholar who was a colleague of Dijkstra at a school in Holland asked me, “Do you teach Dijkstra’s stuff in Taiwan?” The scholar told me “No one teaches his stuff at our school anymore. He had offended everyone before he left.”

What is the goal of your study? What difficulties have you encountered and what part of human life do you want to improve?

This is a tough question. Knowing that there are many things out there that I still don’t understand makes me extremely uncomfortable. When I was still a child, I often felt lost in many problems. Then gradually I understood more things and one program after another becomes clearer to me. That’s when you realize how much you still don’t know, so I keep pursuing the next goal till today.

People are born with curiosity and the sense of aesthetics. Curiosity often makes you sleepless. I often lay awake in bed at night, trying to figure out the answer to a problem, and once I had the answer, I would jump up and down with joy and couldn’t wait to share it with someone else. I often told others that I’m in the entertainment industry. You can also interpret it as “research is the roots of happiness.”

What advice would you give to prospective students of information technology?

Recently, I truly felt that due to the training we have received, we are captivated deeply by our predecessors’ achievement. Therefore, we regard those newly developed areas too “engineering-oriented,” without the beauty and depth that we look for. But we have to understand, that the “beauty” which we have known and learned to appreciate resulted from the effort of our predecessors in organizing messy stuff.

What we should do is to concentrate on those unpolished areas and do our best to discover the hidden beauty. I believe this is exactly what Dijkstra means when he said “Beauty is our business.”