The Interview Column

by

Chen Avin and Stefan Schmid

Ben Gurion University, Israel and TU Berlin, Germany
{chenavin,schmiste}@gmail.com

Know the Person behind the Papers

Today: Laura Kovács

Bio: *Laura Kovács is a full professor of computer science at the TU Wien, leading the automated program reasoning (APRe) group of the Formal Methods in Systems Engineering division. Her research focuses on the design and development of new theories, technologies, and tools for program analysis, with a particular focus on automated assertion generation, symbolic summation, computer algebra, and automated theorem proving. She is the co-developer of the Vampire theorem prover and a Wallenberg Academy Fellow of Sweden. Her research has also been awarded with a ERC Starting Grant 2014, an ERC Proof of Concept Grant 2018, an ERC Consolidator Grant 2020, and an Amazon Research Award 2020. Recently, she received financial support from Let's Empower Austria - LEA Frauenfonds to disseminate computer science to elementary schools.*

EATCS: We ask all interviewees to share a photo with us. Can you please tell us a little bit more about the photo you shared?

LK: The picture is taken in one of my very recent and dearest activities at the TU Wien: introducing elementary school children to the art and fun of computer science. I do various puzzle solving activities with kids and develop algorithms together with them. It is energizing to see how creative and fast kids can be - either in sorting, path finding or instructing small robots. We all have a lot of fun and I hope most of the kids will stick to STEM education - at least they say so with excitement when they leave our workshops!

EATCS: Can you please tell us something about you that probably most of the readers of your papers don't know?

LK: I did my bachelor studies in image processing. I was working on optical music recognition and reconstructed music sounds from printed music sheets. It was a fun project where I did a lot of coding and used many linear algebra algorithms. I was fascinated by the topic for almost two years, before I entered the field of symbolic computation.

EATCS: Is there a paper which influenced you particularly, and which you recommend other community members to read?

LK: I cannot really name one single research paper that influenced me the most. However, the paper that pushed me to start work in formal verification is the master thesis "Program Verification with the Mathematical Software System Theorema" by Martin Kirchner, RISC-Linz, Austria, 1999 (Technical Report 99-16). I read this thesis while being a master student at RISC-Linz and I liked the combination of logic, math and software engineering. Thanks to this thesis, I read the seminal works of Edgar Dijkstra, Robert Floyd, and Tony Hoare - as well as many other related papers and works.

EATCS: Is there a paper of your own you like to recommend the readers to study? What is the story behind this paper?

LK: I particularly like my TACAS 2008 paper "Reasoning Algebraically About P-Solvable Loops". This paper summarizes my PhD thesis and it was written in a quite stressful situation. I actually wanted to prove a stronger result than the one presented in the paper: essentially, my goal was to prove that the strongest inductive invariants of so-called polynomial-solvable loops are computable. However, I could only prove this under additional restrictions on the loop semantics and was quite disappointed that time that I could not get stronger theoretical results. It was a tricky situation: I could not complete the proof of the general case, but I also did not succeed in finding a counterexample. As such, in my TACAS 2008 paper I posed the general case as an open challenge for the future. I am glad I have done so because since then quite many undecidable and/or hardness results have been established upon algebraic invariant synthesis. Notably, in our POPL 2024 on "Strong Invariants Are Hard", we have just proved that generalizing my PhD thesis results is not trivial, hitting the computational limits of the famous Skolem problem from number theory. Even more than 15 years after my PhD, the open challenge of my TACAS 2008 paper brings in new research directions to further explore.

EATCS: When (or where) is your most productive working time (or place)?

LK: Even though I dislike waking up early, early morning hours are the best for me. After dropping my kids at school, I usually have 8-10am as the two hours in which I am free to do what I want, on a daily basis. I actually enjoy working late, but this I can only do for limited periods, triggered by deadlines. In addition, I have one day a week where I have no meetings, lectures or other events. I enjoy this day usually in my office.

EATCS: What do you do when you get stuck with a research problem? How do you deal with failures?

LK: I try to read up more on related work and find a colleague to discuss and brainstorm. Failure is relative and is part of research. I would therefore not really say that I/we failed on a research topic, but consider that the chosen methodology did not work as expected. In such cases, I try to understand what and why was different; if I manage to do so, then I failed “well" and have a very good learning outcome.

EATCS: Is there a nice anecdote from your career you would like to share with our readers?

LK: I come from the Hungarian minority of Transylvania, Romania. I manage to confuse people all the time when I stress that I have double citizenship (Romanian and Hungarian) but only one nationality (Hungarian). Quite many administrative forms do not distinguish between citizenship and nationality, so I end up writing a short explanatory text on a form where most likely only one word is needed.

Funnily, when I mention Transylvania, many people associate my background with vampires and Dracula. When I then tell them that I work on a research project called Vampire, they just believe this has been my destiny. It was however quite a coincidence that I started working on the Vampire theorem prover in 2009.

EATCS: Do you have any advice for young researchers? In what should they invest time, what should they avoid?

LK: There are many advices one gets and one should filter out among these advices. Actually, it is impossible to follow all advices. The one that I follow (and tell my students) is rightfully generic: Do what you are the best at! One should not just follow hypes, rather be persistent on own interests. Positioning these interests within trendy research topics is beneficial, but should not be your driving wheel.

EATCS: What are the most important features you look for when searching for graduate students?

LK: Curiosity and flexibility. One should be passionate about curiosity-driven research and be open to new challenges and collaborations. Most research in computer science is collaborative and one should be respectful to colleagues.

It can be hard for students to assess whether they are “ready" for research. I therefore actively approach some of my best students from my lectures at the TU Wien. I talk to them, invite them to group meetings, and try to initiate joint research topics of mutual interest.

EATCS: Do you see a main challenge or opportunity for theoretical computer scientists for the near future?

LK: I see the opportunity of a continuously evolving, hard research field. There are always new problems that need new solutions; once a problem is solved, we are happy to take up new, harder problems. It is like an endless loop: if one is passionate about research, one never gets bored but remains eternally happy in solving problems. Theoretical computer science is a safe place to be in, it will always exist and will always be challenged by new computer science applications and practices.

Please complete the following sentences?

- *Being a researcher* is what I enjoy and am good at.
  When solving a research problem, the community will be interested in your solution. Research is rewarding, but you need to be persistent.

- *My first research discovery* was different than the rest of all my other research results. It was on image processing, although it already used some kind of automated reasoning based on linear algebra.

- Being curious *is key to being a happy academic.*

- *Theoretical computer science in 100 years from now* will be even more important than today, solving even harder problems than the ones we face now.