EATCS Fellows' Advice to the Young Theoretical Computer Scientist

Luca Aceto (Reykjavik University)
with contributions by Mariangiola Dezani-Ciancaglini,
Yuri Gurevich, David Harel, Monika Henzinger,
Giuseppe F. Italiano, Scott Smolka,
Paul G. Spirakis and Wolfgang Thomas

I have always enjoyed reading articles, interviews, blog posts and books in which top-class scientists share their experience with, and provide advice to, young researchers. In fact, despite not being young any more, alas, I feel that I invariably learn something new by reading those pieces, which, at the very least, remind me of the things that I *should* be doing, and that perhaps I am *not* doing, to uphold high standards in my job.

Based on my partiality for scientific advice and stories, it is not overly surprising that I was struck by the thought that it would be interesting to ask the EATCS Fellows for

- the advice they would give to a student interested in theoretical computer science (TCS),
- the advice they would give to a young researcher in TCS and
- a short description of a research topic that excites them at this moment in time (and possibly why).

In this article, whose title is inspired by the classic book *Advice To A Young Scientist* authored by the Nobel Prize winner Sir Peter Brian Medawar, I collect the answers to the above-listed questions I have received from some of the EATCS Fellows. The real authors of this piece are Mariangiola Dezani-Ciancaglini (University of Turin), Yuri Gurevich (Microsoft Research), David Harel (Weizmann Institute of Science), Monika Henzinger (University of Vienna), Giuseppe F. Italiano (University of Rome Tor Vergata), Scott Smolka (Stony Brook University), Paul G. Spirakis (University of Liverpool, University of Patras and Computer Technology Institute & Press "Diophantus", Patras) and Wolfgang Thomas (RWTH Aachen University), whom I thank for their willingness to share their experience and wisdom with all the members of the TCS community. In an accompanying essay, which follows this one in this issue of the Bulletin, you will find the piece I received from Michael Fellows (University of Bergen).

The EATCS Fellows are model citizens of the TCS community, have varied work experiences and backgrounds, and span a wide spectrum of research areas. One can learn much about our field of science and about academic life in general by reading their thoughts. In order to preserve the spontaneity of their contributions, I have chosen to present them in an essentially unedited form. I hope that the readers of this article will enjoy them as much as I have done.

Mariangiola Dezani-Ciancaglini

The advice I would give to a student interested in TCS is: Your studies will be satisfactory only if understanding for you is fun, not a duty.

To a young researcher in TCS I would say, "Do not be afraid if you do not see applications of the theory you are investigating: the history of computer science shows that elegant theories developed with passion will have eventually long-lasting success."

A research topic that currently excites me is the study of behavioural types. These types allow for fine-grained analysis of communication-centred computations. The new generation of behavioural types should allow programmers to write the certified, self-adapting and autonomic code that the market is requiring.

Yuri Gurevich

Advice I would give to a student interested in TCS Attending math seminars (mostly in my past), I noticed a discord. Experts in areas like complex analysis or PDEs (partial differential equations) typically presume that everybody knows Fourier transforms, differential forms, etc., while logicians tend to remind the audience of basic definitions (like what's first-order logic) and theorems (e.g. the compactness theorem). Many talented mathematicians didn't take logic in their college years, and they need those reminders. How come? Why don't they effortlessly internalize those definitions and theorems once and for all? This is not because those definitions and theorems are particularly hard (they are not) but because they are radically different from what they know. It is easier to learn radically different things — whether it is logic or PDEs or AI — in your student years. Open your mind and use this opportunity!

Advice I would give a young researcher in TCS As the development of physics caused a parallel development of physics-applied mathematics, so the development of computer science and engineering causes a parallel development of theoretical computer science. TCS is an applied science. Applications justify it and give it value. I would counsel to take applications seriously and honestly. Not only immediate applications, but also applications down the line. Of course, like in mathematics, there are TCS issues of intrinsic value. And there were cases when the purest mathematics eventually was proven valuable and applied. But in most cases, potential applications not only justify research but also provide guidance of sorts. Almost any subject can be developed in innumerable ways. But which of those ways are valuable? The application guidance is indispensable.

I mentioned computer engineering above for a reason. Computer science is different from natural science like physics, chemistry, biology. Computers are artifacts, not "naturefacts." Hence the importance of computer science and engineering as a natural area whose integral part is computer science.

A short description of a research topic that excites me at this moment in time (and possibly why) Right now, the topics that excite me most are quantum mechanics and quantum computing. I wish I could say that this is the result of a natural development of my research. But this isn't so. During my long career, I moved several times from one area to another. Typically it was natural; e.g. the theory of abstract state machines developed in academia brought me to industry. But the move to quanta was spontaneous. There was an opportunity (they started a new quantum group at the Microsoft Redmond campus a couple of years ago), and I jumped upon it. I always wanted to understand quantum theory but occasional reading would not help as my physics had been poor to none and I haven't been exposed much to the mathematics of quantum theory. In a sense I am back to being a student and discovering a new world of immense beauty and mystery, except that I do not have the luxury of having time to study things systematically. But that is fine. Life is full of challenges. That makes it interesting.

David Harel

Advice I would give to a student interested in TCS If you are already enrolled in a computer science program, then unless you feel you are of absolutely stellar theoretical quality and the real world and its problems do not attract you at all, I'd recommend that you spend at least 2/3 of your course efforts on a variety of topics related to TCS but not "theory for the sake of theory". Take lots of courses on languages, verification AI, databases, systems, hardware, etc. But clearly don't shy away from pure mathematics. Being well-versed in a variety of topics in mathematics can only do you good in the future. If you are still able to choose a study program, go for a combination: TCS combined with software and systems engineering, for example, or bioinformatics/systems biology. I feel that computer science (not just programming, but the deep underlying ideas of CS and systems) will play a role in the science of the 21st century (which will be the century of the life sciences) similar to that played by mathematics in the science of the 20th century (which was the century of the physical sciences).

Advice I would give a young researcher in TCS Much of the above is relevant to young researchers too. Here I would add the following two things. First, if you are doing pure theory, then spend at least 1/3 of your time on problems that are simpler than the real hard one you are trying to solve. You might indeed succeed in settling the P=NP? problem, or the question of whether PTIME on general finite structures is r.e., but you might not. Nevertheless, in the latter case you'll at least have all kinds of excellent, if less spectacular, results under your belt. Second, if you are doing research that is expected to be of some practical value, go talk to the actual people "out there": engineers, programmers, system designers, etc. Consult for them, or just sit with them and see their problems first-hand. There is nothing better for good theoretical or conceptual research that may have practical value than dirtying your hands in the trenches.

A short description of a research topic that excites me at this moment in time (and possibly why) I haven't done any pure TCS for 25 years, although in work my group and I do on languages and software engineering there is quite a bit of theory too, as is the case in our work on biological modeling. However, for many years, I've had a small but nagging itch for trying to make progress on the problem of artificial olfaction — the ability to record and remotely produce faithful renditions of arbitrary odors. This is still a far-from-solved issue, and is the holy grail of the world of olfaction. Addressing it involves chemistry, biology, psychophysics, engineering, mathematics and algorithmics (and is a great topic for young TCS researchers!). More recently, I've been thinking about the question of how to test the validity of a candidate olfactory reproduction system, so that we have an agreed-upon criterion of success for when such systems are developed. It is a kind of common-sense question, but one that appears to be very interesting, and not unlike Turing's 1950 quest for testing AI, even though such systems were nowhere in sight at the time. In the present case, trying to compare testing artificial olfaction to testing the viability of sight and sound reproduction will not work, for many reasons. After struggling with this for quite a while, I now have a proposal for such a test, which is under review.

Monika Henzinger

- Students interested in TCS should really like their classes in TCS and be good at mathematics.

- I advice young researchers in TCS to try to work on important problems that have a relationship to real life.

- Currently I am interested in understanding the exact complexity of different combinatorial problems in P (upper and lower bounds).

Giuseppe F. Italiano

The advice I would give to a student interested in TCS There's a great quote by Thomas Huxley: "Try to learn something about everything and everything about something." When working through your PhD, you might end up focusing on a narrow topic so that you will fully understand it. That's really great! But one of the wonderful things about Theoretical Computer Science is that you will still have the opportunity to learn the big picture!

The advice I would give a young researcher in TCS Keep working on the problems you love, but don't be afraid to learn things outside of your own area. One good way to learn things outside your area is to attend talks (and even conferences) outside your research interests. You should always do that!

A short description of a research topic that excites me at this moment in time (and possibly why) I am really excited by recent results on conditional lower bounds, sparkled by the work of Virginia Vassilevska Williams et al. It is fascinating to see how a computational complexity conjecture such as SETH (Strong Exponential Time Hypothesis) had such an impact on the hardness results for many well-known basic problems.

Scott Smolka

Advice I would give to a student interested in TCS Not surprising, it all starts with the basics: automata theory, formal languages, algorithms, complexity theory, programming languages and semantics.

Advice I would give a young researcher in TCS Go to conferences and establish connections with more established TCS researchers. Seek to work with them and see if you can arrange visits at their home institutions for a few months.

A short description of a research topic that excites me at this moment in time (and possibly why) Bird flocking and V-formation are topics I find very exciting. Previous approaches to this problem focused on models of dynamic behavior based on simple rules such as: Separation (avoid crowding neighbors), Alignment (steer towards average heading of neighbors), and Cohesion (steer towards average position of neighbors). My collaborators and I are instead treating this as a problem of Optimal Control, where the fitness function takes into account Velocity Matching (alignment), Upwash Benefit (birds in a flock moving into the upwash region of the bird(s) in front of them), and Clear View (birds in the flock having unobstructed views). What's interesting about this problem is that it is inherently distributed in nature (a bird can only communicate with its nearest neighbors), and one can argue that our approach more closely mimics the neurological process birds use to achieve these formations.

Paul G Spirakis

My advice to a student interested in TCS Please be sure that you really like Theory! The competition is high, you must love mathematics, and the money prospects are usually not great. The best years of life are the student years. Theory requires dedication. Are you ready for this?

Given the above, try to select a good advisor (with whom you can interact well and frequently). The problem you choose to work on should psyche you and your advisor!

It is good to obtain a spherical and broad knowledge of the various Theory subdomains. Surprisingly, one subfield affects another in unexpected ways.

Finally, study and work hard and be up to date with respect to results and techniques!

My advice to a young researcher interested in TCS Almost all research problems have some difficulty. But not all of them are equally important! So, please select your problems to solve carefully! Ask yourself and others: why is this a nice problem? Why is it interesting and to which community? Be strategic!

Also, a problem is good if it is manageable in a finite period of time. This means that if you try to solve something open for many years, be sure that you will need great ideas, and maybe lots of time! However, be ambitious! Maybe you will get the big solution! The issue of ambition versus reasonable progress is something that you must discuss with yourself!

It is always advisable to have at least two problems to work on, at any time. When you get tired from the main front, you turn on your attention to the other problem.

Try to interact and to announce results frequently, if possible in the best forums. Be visible! It is important that other good people know about you. “Speak out to survive!”

Study hard and read the relevant literature in depth. Try to deeply understand techniques and solution concepts and methods. Every paper you read may lead to a result of yours if you study it deeply and question every line carefully! Find quiet times to study hard. Control your time!

A field that excites me: the discrete dynamics of probabilistic (finite) population protocols Population Protocols are a recent model of computation that captures the way in which complex behavior of systems can emerge from the underlying local interactions of agents. Agents are usually anonymous and the local interaction rules are scalable (independent of the size, n, of the population). Such protocols can model the antagonism between members of several “species” and relate to evolutionary games.

In the recent past I was involved in joint research studying the discrete dynamics of cases of such protocols for finite populations. Such dynamics are, usually, probabilistic in nature, either due to the protocol itself or due to the stochastic nature of scheduling local interactions. Examples are (a) the generalized Moran process (where the protocol is evolutionary because a fitness parameter is crucially involved) (b) the Discrete Lotka-Volterra Population Protocols (and associated Cyclic Games) and (c) the Majority protocols for random interactions.

Such protocols are usually discrete time transient Markov Chains. However the detailed states description of such chains is exponential in size and the state equations do not facilitate a rigorous approach. Instead, ideas related to filtering, stochastic domination and Potentials (leading to Martingales) may help in understanding the dynamics of the protocols.

Some such dynamics can describe strategic situations (games): Examples include Best-Response Dynamics, Peer-to-Peer Market dynamics, fictitious play etc.

Such dynamics need rigorous approaches and new concepts and techniques. The ‘traditional’ approach with differential equations (found in e.g. evolutionary game theory books) is not enough to explain what happens when such dynamics take place (for example) in finite graphs with the players in the nodes and with interactions among neighbours. Some main questions are: How fast do such dynamics converge? What is a ‘most probable’ eventual state of the protocols (and the computation of the probability of such states). In case of game dynamics, what is the kind of ‘equilibria’ to which they converge? Can we design ‘good’ discrete dynamics (that converge fast and go to desirable stable states ?). What is the complexity of predicting most probable or eventual behaviour in such dynamics?

Several aspects of such discrete dynamics are wide open and it seems that the algorithmic thought can contribute to the understanding of this emerging subfield of science.

Wolfgang Thomas, “Views on work in TCS”

As one of the EATCS fellows I have been asked to contribute some personal words of advice for younger people and on my research interests. Well, I try my best.

Regarding advice to a student and young researcher interested in TCS, I start with two short sentences:

- Read the great masters (even when their h-index is low).
- Don’t try to write ten times as many papers as a great master did.

And then I add some words on what influenced me when I started research — you may judge whether my own experiences that go back to “historical” times would still help you.

By the way, advice from historical times, where blackboards and no projectors were used, posed in an entertaining but clearly wise way, is Gian-Carlo Rota’s paper “Ten Lessons I Wish I Had Been Taught” (`http://www.ams.org/notices/199701/comm-rota.pdf`). This is a view of a mathematician but still worth reading and delightful for EATCS members. People like me (68 years old) are also addressed — in the last lesson “Be Prepared for Old Age”...

Back in the 1970’s when I started I wanted to do something relevant. For me this meant that there should be some deeper problems involved, and that the subject of study is of long-term interest. I was attracted by the works of Büchi and Rabin just because of this: That was demanding, and it treated structures that will be important also in hundred years: the natural numbers with successor, and the tree of all words (over some alphabet) with successor functions that represent the attachment of letters.

The next point is a variation of this. It is a motto I learnt from Büchi, and it is a warning not to join too small communities where the members just cite each other. In 1977, when he had seen my dissertation work, Büchi encouraged me to continue but also said: Beware of becoming member of an MAS, and he explained that this means “mutual admiration society”. I think that his advice was good.

I am also asked to say something about principles for the postdoctoral phase. It takes determination and devotion to enter it. I can say just two things, from my own experience as a young person and from later times. First, as it happens with many postdocs, in my case it was unclear up to the very last moment whether I would get a permanent position. In the end I was lucky. But it was a strain. I already prepared for a gymnasium teacher's career. And when on a scientific party I spoke to Saharon Shelah (one of the giants of model theory) about my worries, he said "well, there is competition". How true. So here I just say: Don't give away your hopes — and good luck. The other point is an observation from my time as a faculty member, and it means that good luck may be actively supported. When a position is open the people in the respective department do not just want a brilliant researcher and teacher but also a colleague. So it is an important advantage when one can prove that one has more than just one field where one can actively participate, that one can enter new topics (which anyway is necessary in a job which lasts for decades), and that one can cooperate (beyond an MAS). So for the postdoc phase this means to look for a balance between work on your own and work together with others, and if possible in different teams of cooperation.

Finally, a comment on a research topic that excites me at this moment. I find it interesting to extend more chapters of finite automata theory to the infinite. This has been done intensively in two ways already — we know automata with infinite "state space" (e.g., pushdown automata where "states" are combined from control states and stack contents), and we know automata over infinite words (infinite sequences of symbols from a finite alphabet). Presently I am interested in words (or trees or other objects) where the alphabet is infinite, for example where a letter is a natural number, and in general where the alphabet is given by an infinite model-theoretic structure. Infinite words over the alphabet $\mathbb{N}$ are well known in mathematics since hundred years (they are called points of the Baire space there). In computer science, one is interested in algorithmic results which have not been the focus in classical set theory and mathematics, so much is to be done here.