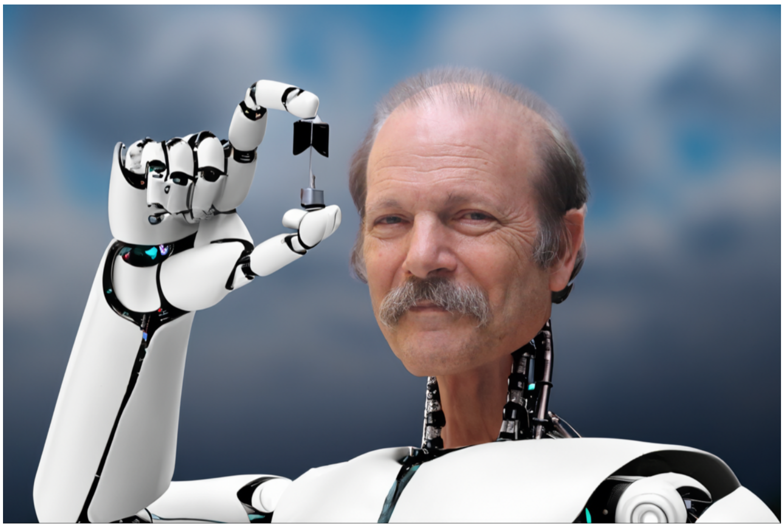
Know the Person behind the Papers

Today: Moshe Vardi

Bio: *Moshe Y. Vardi is University Professor and the Karen Ostrum George Distinguished Service Professor in Computational Engineering at Rice University, where he is leading an Initiative on Technology, Culture, and Society. His interests focus on automated reasoning, a branch of Artificial Intelligence with broad applications to computer science, including machine learning, database theory, computational-complexity theory, knowledge in multi-agent systems, computer-aided verification, and teaching logic across the curriculum. He is also a Faculty Scholar at the Baker Institute for Public Policy at Rice University.*



EATCS: Nice to meet you, Moshe! This is actually the first live interview we conduct, usually the EATCS interviews are done offline.

MV: I very much prefer it live and interactive! This gives us more flexibility and you can ask additional questions spontaneously. This is also the reason why I don't teach with slides but on the board, "old school". This allows me to react to students better and also slows me down – a good control mechanism for my teaching speed.

EATCS: We ask all interviewees to share a photo with us.

MV: Most photos of me are made when the Rice University photographer shows up because he noticed that '*you look older than the picture on your website!*' I share two pictures. The first one was created by my son, for a poster for my lecture on robots and jobs. The second picture shows me with my wife, Pam, feeding a llama in Machu Pichu in 2018.

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

MV: I am a second-generation Holocaust survivor. I served 5 years in the military and was an artillery officer. This was a typical trajectory for people with a physics background like me.

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

MV: One of my all-time favorite papers has a shortest possible title: *Alternation*, by Chandra, Kozen, and Stockmeyer. It is a very beautiful paper. I first read it as a paper on complexity theory but later it turned out that the concept of *alternating automata* (which were also introduced in that paper) became an important tool in my work on program verification. Alternation is really another way of game play, where players take turns: a major theme of my research. What was new in this paper is the idea to make games a computational construct. This idea is now also used intensively for synthesis. Alternation is also powerful because it is very close to logic. For example, the initial translation from LTL to Büchi automata is significantly simplified if taught or implemented via the concept of alternating automata.

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

MV: My second most-cited paper is on the complexity of relational queries. I published this paper right after my PhD, but most of the research I conducted during my PhD, as a side project. It started with me being puzzled that there are seemingly contradictory complexity results for relational queries in the literature. One result claimed that the complexity of first-order queries is PSPACE complete, and another result claimed that the complexity of existential second-order queries is NP, which seemed strange as NP contains PSPACE but second-order logic is more expressive than first-order logic. So I realized that there are different ways to measure the complexity of queries, related to the expression complexity or the data complexity. Looking back, this was the first paper on what we call today multivariate complexity theory or parameterized complexity, studying how the different

input variables impact the complexity. I I sometimes also call it logical algorithmics, which I see as an attractive alternative way of developing algorithms, which fits database queries very well: rather than devising problem-specific algorithms one-by-one, there is a meta-algorithm and a to-be-studied property; the problem-specific algorithm is then obtained by compilation of this meta-algorithm. This idea came from relational database theory, which was the first topic of my research.

EATCS: Are there still open research challenges in this area today?

MV: Yes, for example in the context of constrained satisfaction problems. I am very interested to understand how we can use tree decompositions to evaluate queries faster. 20 years ago researchers were sceptic about this idea, as computing tree decompositions itself is a hard problem. However, now we not only have approximation algorithms but also good tools, and it is time to revisit these ideas, which may become tractable.

In general, I believe we need to revisit our definitions of tractability and complexity theory in general. Every year, tools like SAT solvers are getting significantly better. Problems that we thought are intractable, are now tractable in practice. We are now able to solve very large SAT instances that come from real-life applications.

What we thought is hard, is not hard. At the same time, what we thought is easy is not always easy: for several polynomial-time problems, we currently only have high-degree polynomial algorithms with huge constants.

We start to realize that complexity theory, one of the most beautiful theories we currently have, offers us only limited insights into the actual hardness of problems in practice. Ariel Rubinstein, the famous Israeli economist, wrote in his book on *Economic Fables* that our fundamental economical models can *only inform* economic decision making in the real world; the latter often comes with additional constraints and ultimately is also about values, politics, etc.

This is similar with complexity theory, which can provide us with guidelines and explanations, but not accurate predictions of practical run times.

EATCS: You are very productive and active on many fronts. Can you share with us how you manage your time? And when is your most productive working time?

MV: I am not a good time manager. I often juggle many balls, at any point of time several balls are on the floor. I just hope that the important balls on the floor will scream and catch my attention!

One has to learn to say no. At least I try to start with an empty inbox every year.

I am not a night owl, and I think sleep is very important, trying to be disciplined. One has to realize that a career is a marathon, not a sprint. I tried once, 30 years ago, to do an all-nighter to finish a paper, but I realized at midnight that my writing gets bad. So I never tried it again.

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

MV: When I was a postdoc, I had a single-author FOCS paper. At the reception of the conference, before my presentation, a colleague came to me and said that he really liked the paper. However, he got stuck in one of the proofs. I went to my room and noted a bug. I tried to fix it but I could not do it. So I had to give my presentation the next morning and say that a proof was wrong. Not a nice experience for a young postdoc! Fortunately, it turned out a few years later that my theorem was correct, but a new proof idea was necessary.

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

MV: People are different, and everyone has to figure out themselves what works for them. Also my students are very different, and I am not a good predictor of their success. I mainly judge students on their technical capabilities, which, however, are not necessary or sufficient for success. There are people who are technically not so strong but really good at managing their careers; and there are technically very strong researchers that don't act wisely. We are working within a social context, within a community, and we have to be aware of this. Think of Sheldon Cooper of Big Bang Theory, who is very smart but whose view of how the world works is very imperfect.

I have also seen very smart young researchers who take the dangerous strategy of focusing on solving a single very hard open problem. I always recommend to diversify the research. It is often difficult to guess upfront how hard a problem is going to be.

Generally, it is important to be able to create a vision and tell a story. I always ask my students to give a dry run for their talks, and try to deliver a story. All cultures love stories, and there is a universal 3-Act structure: introduce the characters, create a crisis for the characters, and resolve the crisis. I suggest to also use this structure for presenting your research.

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

MV: I don't search for students, they usually come to me. They typically took my class and I never declined an interested student. In contrast to Europe, in the U.S., the department selects the students and I usually try to work with my students as long as I can, and at least bring them to a masters degree.

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

MV: The computer science field is still very young and evolving. For example, complexity theory is far from perfect and in a crisis. We need a new theory that lowers the discrepancy to the real world. I see the current situation as a huge opportunity: now we can become inventive again! We also had this big successes around machine learning, generative AI, and deep learning, which we don't understand well fundamentally. We need more theory to shed light on these approaches and why/when they work.

There are many success stories about theories which became very relevant in practice. One of my most successful research results, related to automata-theoretic model checking, actually built upon fundamental results that were originally derived in purely theoretical studies, without specific applications in mind. The elegant theories developed in the context of decision procedures for monadic structures or fixpoints, turned out to have very practical applications and gave rise to industrial tools.

There is also this debate about beauty. For example, physicists consider string theory beautiful, but some believe that it led us astray. Beauty does not always lead to usefulness. But in case of automata-theoretic model checking, the simplicity and beauty was actually important to make it practically relevant, as it is easy to understand and implement. I believe that it is not about beauty but about simplicity, for implementation.

I believe that good theory has huge potential. One of the success stories in computer science, relational databases, relies on query languages based on first-order logic theory. Initially, systems researchers were very concerned that this approach will never perform well and hence not be practical. In fact, there were big controversies at that time, for example inside IBM. And now declarative query languages are the foundations of our Western Civilization. It was one of the research contributions that received the Turing Award the fastest.

EATCS: Can you recommend some source of information that you enjoy (e.g., a specific blog, podcast, youtube channel, book, show, ...)?

MV: One of the books that I like very much is *The Universal Computer: The Road from Leibniz to Turing*, by Martin Davis. It tells the story how computing arose, going back all the way to Leibniz. Usually we teach the students mainly how the algorithms and theories work, but not much about the history how these concepts were developed. I think it is a mistake not to include the historical dimension. All these great ideas were created by human beings, and students should understand also this dimensions.

Actually I also gave a similar talk, *From Aristotle to the iPhone* (`https://www.youtube.com/watch?v=iWWqUIzDIeQ`), in which I try to tell this story in one hour. I gave this talk in different versions. Once I was asked to give it at UCLA, in a more lively version. So I added one line to the abstract: *this is a story where the protagonists usually die young, miserably, or both.* And this attracted a big audience, and I realized how much people are interested in the human dimension. I also gave this talk at the UNESCO Logic Day. The interest was so large that our Zoom licence had to be extended multiple times to accommodate about one thousand listeners.

Please complete the following sentences?

- *Being a researcher...* I tell you a story. When I was in the army, we had to practice parachute jumps. Once I was last to get onto the plane, and so I had to be first at the door from where the jump takes place. Standing there for minutes in front of the open air, engines roaring, ready to jump. These moments between plane and the "nothingness", is a most alive and exciting feeling. This is a beautiful metaphor for research: we are standing at the edge of the unknown, and all we have to do is jump, into an exciting space. I want my students to feel the same.

- *My first research discovery...* was actually accidental. I was a master student and did not know what to explore. I attended some seminar and I got to present a paper, which at the end raised an open question. This problem intrigued me and this became my master thesis. And then this topic even evolved into my PhD thesis.

- Being resilient ... *is key to being a happy academic.* Being professor is not an easy job. People get their PhDs at the end of their 20ties, and then need to be productive researchers for 40 years. Topics evolve significantly over these time frames, and even if you have an excellent reputation, it stays competitive and your papers still get rejected. In fact, sometimes the expectations on you are even higher if you are more successful. You also need to be able to cope with the freedom that one has.

- *Theoretical computer science in 100 years from now...* It is hard to give predictions, as the field is still young, compared to other disciplines. It still needs to mature. For example I am very curious how complexity theory will evolve. Will it be a different theory? Or will the current one be refined? My hope is that we will have theories that give better answers. I think the best is yet to come!