Vladik Kreinovich

The University of Texas at El Paso

Tseytin's last name comes from the German word Zeit — time. And like time itself, he was always moving, almost changing his interests, always producing results and ideas in new areas.

He started as a mathematician interested in applicability of his results. Mathematics has many applications — in physics, in engineering, in biology, everywhere, and in all these applications, it is not enough to prove that the solution exists — we need to actually find this solution. It is not enough to prove that there exists a trajectory of going to the Moon — we need to compute this trajectory. So Tseytin joined Andrei Andreevich Markov Jr. and Nikolai Aleksandrovich Shanin in developing a mathematical formalism in which existence would mean the existence of a computational method for constructing this object. They called this formalism constructive mathematics. At first, the main focus of this research area was on the methodological issues: most theorems in this area were either negative (that there is no general algorithm for solving a problem) or rather trivial. Tseytin proved the first non-trivial (and unexpected) positive result: that every computable function from reals to reals is computably continuous, i.e., there is an algorithm that for every x and for every $\varepsilon > 0$ produces a δ for which $|x - x'| \le \delta$ implies $|f(x) - f(x')| \le \varepsilon$. In the mid-1950s, when he proved this result, there was little connection with the West, so — as happened many times in those days — this theorem remained unknown to Western researchers and was independently rediscovered there. At that time, the situation seemed clear: if there is an algorithm, this means that the problem is solvable — and if this algorithm takes too long to finish now, let us wait a few years, when computers will become faster.

Tseytin was interested in applicability, so not only he designed algorithms, he enjoyed programming them and making them work. And in this process, he realized that not all algorithms are feasible — if an algorithm requires exponentially many steps, e.g., 2^n steps, then already for reasonable n, it would require more time than the lifetime of the Universe. In late 1960s, he showed that some algorithms do require exponential time — for example, resolution method, a widely known (and still actively used) method for checking whether a Boolean formula is satisfiable. This was an interesting result on its own — it started the whole field of proof complexity. But as with constructive mathematics, Tseytin tried his best to go beyond negative results and prove something positive — and he succeeded. He showed that for some formulas for which any resolution-method proof is exponentially long, a feasible proof can be obtained if we introduce a new variable which is equivalent to a combination of the original ones — what in mathematics in general would be called a new definition. From the methodological viewpoint, this result was revolutionary — it went contrary to what we were all being taught in math, that one does not argue about definitions. We all used to laugh at a geological paper that described a heated discussion of what is a proper definition of some geological epoch. We did not value much those who come up with definitions, only those who prove results. And Tseytin's theorem confirmed the intuition that a few had — that definitions are often critically important, that a proper definition can be as important as a good theorem.

This feasible-vs.-exponential time distinction seemed to bring a new clarity in his vision of the world: algorithms requiring exponential time are not feasible, but faster algorithms are. Unfortunately, his continuing programming experience showed that this seemingly natural division does not exactly capture what is practically feasible: sometimes an exponential-time algorithm is feasible in all practical cases, and sometimes a linear-time algorithm is not practical at all. And here, he could not find a clearer separation (and, to his defense, nor could anyone else). And so Tseytin practically stopped proving results, he did not even want to talk about his previous theorems — since, as he explained, he lost his original clarity.

And he jumped into computing. I was lucky to take a course on Operating Systems with him when I was a student at St. Petersburg University. It was a very unusual course. It is easy to make such a course very mathematical — I have seen many textbooks that do exactly this, with queuing theory etc., similarly to how we were taught physics at our Math Department: a lot of equations and proofs, not much of physical intuition. This is not how Tseytin taught this class. Yes, he explained all the ideas, techniques, and algorithms — and it was so up-to-date, even ahead-of-date, that many years later when I myself had to teach such a class, I used many notes from Tseytin's lectures. His class's main emphasis was on challenging open problems. He encouraged all of us to come up with new non-trivial questions, new challenges, new ideas — this was what he valued more than reproducing what he taught. But of course, to come up with new ideas, we had to learn what is there now — and many of us spend many efforts on his class (and learned more from it) than from other computer-related classes. In general, I learned a lot from his teaching style.

And he continued to come up with new ideas, ideas that will probably years to be appreciated. Let him rest in peace.