

# Educational Matters

Hans-Jörg Kreowski

May 1997

Meanwhile, I am teaching theoretical computer science (TCS) for more than 20 years. In all those years, I have tried very hard to present the important topics and to explain their significance. I know that I have learnt a lot and improved my understanding of the matter while preparing and presenting the courses. But what about my students? Most of them succeed in doing the exercises and in passing the exams if the questions are straightforward enough and refer strictly to the material at hand (may this be class room notes or text book chapters). Nevertheless, there is much room for doubts. Most students (at least in Bremen) dislike or fear theoretical computer science and do not hesitate to say so. They doubt its usefulness and are sure that it is not needed in practice. If it happens in a course that this position is weakened in any way or even some interest in theoretical questions arises, then the students start to feel and regret their deficiencies in mathematics. They believe that nothing can be done about it, unfortunately. I do rather not describe my frustration and pain when I raise questions like “Is there any function that cannot be computed by a C++ program?” or “If the number of operations per second a computer performs is doubled by a new technology, how much is gained in running programs for route planning, scheduling, etc.?”.

The description may sound overpessimistic because I do not mention the more hopeful indications. Nevertheless, I may not be the only one who is not always confident of the what, how and why of teaching theoretical computer science or of its effect to students. Hence I would like to propose a broad 25th-EATCS-birthday discussion on the subject in this column.

Call for contributions

## On the (im-)possibilities of teaching theoretical computer science

In the times of WWW and internet surfing, multimedia and virtual reality at the dawn of the information society, theoretical computer science seems no longer up to date and teaching it more difficult than ever. All readers of the *Educational Matters Column* are invited to submit notes, reports, case studies, position statements, essays, short stories, observations,

provocations, jokes, etc. on aspects of teaching theoretical computer science, its possibilities and impossibilities. The following list of topics is not exhaustive.

- *General.* How is TCS taught at your institute or in your country? How should it be done? What are the aims and scope? Or is it necessarily a vain attempt?
- *Contents.* Is it still meaningful to teach traditional concepts like Chomsky grammars, Turing machines, finite automata, and such? Or should the stuff from the 30s to the 60s be replaced by more recent subjects, notions, and results?
- *Relation to other areas.* Which relations of TCS to other areas should be pointed out in TCS courses. How and to which detail? Or is this out of scope, and should be done elsewhere?
- *Use of tools.* Many concepts, results and proofs in TCS are difficult to understand. Is it helpful to use tools for computation and visualization? Are there any such tools around? Do they work properly? What should they look like? What should they support? Or would the use of tools in TCS teaching distract from the important aspects?
- *Use of multimedia.* The usual lecture combines already various media: transparencies, black and/or white boards, class room notes and/or text books. Is it helpful to employ fancier media like video and internet? Should a lecture on TCS turned into a multimedia show? Or is the traditional one-person show with a series of monologues still entertaining enough?
- *Students.* Is there a minimum level of knowledge and understanding of TCS all students must reach? If so, how should this level be defined? Is it enough to know the main definitions and results, or should we insist in a deeper understanding?  
Or should TCS courses be addressed to the best students only because most students in computer science are not gifted enough to get the meaning of TCS?

Contributions should be sent preferably by email (before 29 August 1997 for the next issue or any time later for following issues) to

Prof. Dr. Hans-Jörg Kreowski  
Fachbereich Mathematik und Informatik  
Universität Bremen  
Postfach 33 04 40  
D-28334 Bremen  
email: kreo@informatik.uni-bremen.de  
Fax: ++49-421-218-4322