4.2 How to Communicate Proofs or Programs
Dirk Siefkes

4.2.1 An opening question

Why is formalization so much fun to do or to talk about, and so boring to listen to? How excited most of us get when we prove a new theorem or develop a program, or even more when we explain it to someone else! The “someone” may be a fellow researcher, or a class of students, or the audience at a conference. But do we show the same excitement if we are the someone, but are not ourselves involved in the question, or at least active in the field? Why do students shun mathematical lectures? Why do we fall asleep during a talk? With writing and reading papers it is not so different. Or am I the only one who loves to do mathematics, but tries to avoid reading the papers of others?

Any formalization seems to have this effect: it makes things easier, but less fun. On the freeway we drive smoother than on secondary roads, but have trouble to stay awake. To follow another person’s formal built-up is like that: we go straight ahead, fenced in, no decisions are necessary or even wanted. When on the other hand we search for a proof or design a program, we have to be creative. The situation is similar with writing or talking about a result: When we reproduce a proof, we re-live the excitement of finding it – or of understanding it if it is not our own.

So this is an easy answer to the question I started with: Following our own path is fun, being ordered is not. Gregory Bateson\(^1\) describes the dual pair of form and process. Processes happen. They happen in given forms, as cars run on roads; and they produce forms, as traffic results in freeways, which in turn allow increased traffic. We formalize to impose order in the world, and thus to give orders to others. When I present a proof, the others have to believe me; they even have to follow my way towards the theorem. When I write a program, it is first the machine that has to follow me. But soon it is the other guy who has to, especially when I am a theoretician.

There is much effort now in software engineering to replace, or at least to complement, the product-oriented view by the human-centred view\(^2\). What kind of anti-thesis is this? How is it possible that people concentrate on products instead of humans? My product is my baby, that is, an image of myself. In presenting it, I put myself into the center without having to blush. As Gordon Pask formulated it in a conversation during this conference: “The prevailing view in science is the I-I-I-perspective as opposed to the I-you-perspective. This is what needs to be changed.”

\(^1\) Bateson, 1972, Bateson, 1980
\(^2\) See for example Nurminen, 1988.

C. Floyd et al. (eds.), Software Development and Reality Construction
© Springer-Verlag Berlin Heidelberg 1992
4.2 How to Communicate Proofs or Programs

How can we hand over what we have learned if not in the form of a product? Is there a way to induce the processes we have gone through, in the partner? Again, this sounds like a moralistic question. What has formalization got to do with morals? It seems that my rash answer to the opening question – as rash answers tend to do – posed new questions rather than answering the old one.

Let us start afresh from safer ground with a more traditional question.

4.2.2 What is a proof?

Every logician can answer that one: proofs are trees. In a proof we proceed from axioms, assumptions, hypotheses through logical steps to the statement to be proved. Thus the proof tree contains the axioms at the leaves, the logical steps at the branching points, and the theorem at the root. Quod erat demonstrandum.

Such proofs do not occur in praxis, as every logician would admit. In principle, however, any mathematical proof could be broken down into logical steps, and thus changed into a tree. We just have to blow away the accidentals, delete the ephemeral, concentrate on the essentials, work out the underlying structure, and bingo! there is the proof tree – as illustrated in Fig. 4.2–1.

Mathematicians proving theorems proceed on a different level. Not because they are careless or lazy, but because by a formal proof they would not prove anything. We could check its correctness, even by a machine. But we would be unable to understand it, and thus would not understand the theorem.

Traditionally one distinguishes between rational and rhetorical speech. In rational speech we “prove” something; that is, we proceed from evident axioms to derived assertions in self-explanatory steps. Everyone in his right mind has to accept what we say. In rhetorical speech we try to “make clear” our ideas directly, by referring to personal experience, using pictorial language, and drawing analogies, in order to convince the audience. Whether we succeed depends more on how we say it than on what we say. In particular we ourselves do not have to believe in what we say. In the rational tradition rhetoric becomes the art to fool the audience; a “rhetorical question” is a fake question.

Paul Feyerabend mocks the rational tradition when he explains why it has been so successful. “Proofs are stories that tell themselves” is his definition. Story-tellers had better have their stories well told if they want people to believe them. Persons proving a statement on the other hand need not take such pains: The concepts they work with are so general that they already contain the whole argument; thus it develops by itself. In his books Feyerabend demonstrates that the scientist himself does not proceed rationally, but rather like an artist: following the fashion, subject to partialities, looking for advantages. Is the mathematician proving a theorem an artist, too?

Obviously, we are never completely rational when we do a proof. The axioms themselves cannot be proved; we have to “show” them directly, in rhetorical

---

3 [Feyerabend, 1984]