

Copyright Notice

The following manuscript

EWD 474: Trip report visit ETH Zurich, 3–4 February 1975 by E.W.Dijkstra
is held in copyright by Springer-Verlag New York.

The manuscript was published as pages 95–98 of

Edsger W. Dijkstra, *Selected Writings on Computing: A Personal Perspective*,
Springer-Verlag, 1982. ISBN 0-387-90652-5.

**Reproduced with permission from Springer-Verlag New York.
Any further reproduction is strictly prohibited.**

Trip report visit ETH Zurich, 3 - 4 February 1975 by E.W.Dijkstra.

Invited by Niklaus Wirth I gave three lectures at the Eidgenössische Technische Hochschule Zurich. The first one (Monday 16.00 - 18.00) was reasonably successful, the second one (Tuesday 11.00 - 12.00) was bad --at the end I lost my way in a trivial proof and had to give up--, the last one (Tuesday 14.15 - 16.00) went perfectly. For some reason I was very tense: I have, for instance, completely forgotten to open each lecture (as usual) with a quotation! I also forgot the first day to invite "interrupting questions". Niklaus suggested that at my second talk I should give the audience some "homework" for the break at lunch. I did so, but at the beginning of the second talk instead of at its end: one question was so intriguing that more than one member of the audience tried to solve the problem during the lecture.

My trip from Eindhoven (dep. 9.08) to Zurich (arr. 18.14) was most comfortable. I had to change twice (Köln and Basel), but as all trains kept perfect time and I travelled light, this was no problem at all. It was my intention to prepare the lectures during the journey, but that was not entirely successful: my thoughts wandered away and I ended up reading in "Mathematics in Western Culture", a book that I can highly recommend (in spite of the sad foreword by R.Courant, that refers to today's "anti-mathematical fashion in education"). It has been written by Morris Kline. I find such accounts of the birth of new sciences very instructive and inspiring: the analogy with what happens in computing science is sometimes quite close.

Niklaus picked me up at the Zurich railway station and took me to his house where I slept the next two nights. Sunday evening he had some family over from various parts of the world, and not counting Dutch, four languages were spoken at the dinner table. (At the end of the dinner I addressed their oldest daughter upon her request in Dutch: it was truly a multi-lingual dinner!) The next evening, after dinner we' --i.e. Niklaus's wife and children and I-- gave a small "house concert" (piano, recorder, ukelele, clarina and vox humana) and I found it touching to observe the earnest devotion of the young performers. Later that evening Niklaus and I were joined by Gene Golub --the numerical mathematician who is now at Stanford-- and an American statistica called "Grace" --I am sorry that I do not remember her full name, for she contributed a fair share to an enjoyable evening-- who now came from Oxford and was on her way to Rehovot, Israel. The last evening -- knowing that I wanted to sleep on the night train-- I drank more freely. As a result I slept very well, but I am afraid that when I was woken up at 5.15., so that I could leave the sleeper at Köln at six o'clock in the morning, I did not feel too happy. (Whether I would have felt any better without the alcohol of the previous night is, of course, an open question!) At 9.04 I arrived in Eindhoven, where my wife was with the car to pick me up.

I had two unexpected, but pleasant encounters. The one was with Dana Scott, who happened to pass his sabbatical leave at the ETH Zurich. On Sunday evening Niklaus gave me a 15-page letter from Scott, which I studied before I went to bed and discussed with Scott the next afternoon, during the hours before my first performance. The other was that, after I had spoken a few minutes, I suddenly discovered Robert Fano in my audience (he was the director of Project MAC at the time that I was guest-professor at MIT); he happened to pass his sabbatical year at the IBM Laboratory in Zurich. It was a pleasure to meet him again.

The remaining time I have talked with Niklaus, his colleagues and assistants, mainly about their work and their ideas. I observed a consensus that skepticism about automatic program composition is as justified as skepticism about automatic theorem provers. (As I have always stayed far away from these subjects, I have to rely upon opinions and expectations of those with more experience or better insight in the field.) I was shown a very nicely decomposed "message switching system" designed for a "terminal": in particular the high degree of isolation of hardware-dependent parts was impressive. It had been implemented for a Hewlett-Packard machine, and a few PDP-machines were the next candidates. To write the system --i.e. nearly all of it-- in an extended version of PASCAL and then performing a "hand translation" is, indeed, the most sensible approach.

We also talked about the teaching of programming and the position and role of computer science. My strong impression is that the way in which the mathematical department in Chicago first absorbed and then strangled computing science --Golub told the story-- is not an isolated case: such things are in danger of occurring at more places, universities and journals. Apparently it has happened already with ACTA INFORMATICA; Niklaus expressed himself very strongly (like Turski did in November), viz. that ACTA INFORMATICA is now doomed beyond salvation. The mathematicians immediately restrict and extend the subject to what they see in it. I may write a letter to the Editor, Niegel, but it won't help much, for we know his answer: he will explain the situation by saying that such are the papers he receives (and we all believe that, for they are so much easier to write!)

The other threat comes from organized user groups, that prefer complete stagnation ("the physicist's FORTRAN"). Upon closer scrutiny, their arguments are alarming. The argument for standardization is the exchange of their expensive programs, but that means that they exchange the bugs as well. (And it is somebody's law that, the more expensive a program, the greater the number of bugs.) In the old days, physicists used to repeat each other's experiments, just to be sure. Now they repeat each other's mistakes, fully automated repetition! The only justification for exchange, for sharing, is the ultra-high quality of the shared object, but now they insist upon sharing because it was expensive to make, although it is almost certainly expensive junk. On account of their desire to share, they should welcome all improvements that could raise the quality of the shared object, but they resist all change with the fallacious argument that they cannot afford to do so. I sometimes smell also the unwillingness to admit that their professional responsibility extends itself over the quality of their "vital" programs. It is frightening: here we have a mechanism that could easily kill a science on a world-wide scale! The only respectable answer of computing science is never to yield to the pressure.

In one respect I found the intellectual climate a little bit "sticky"; I do not know whether this is characteristic for the ETH Zurich or whether it is a Swiss national trait to be "solid" first and only "adventurous" as far as then allowed (and that is not very far). Part of my talk dealt with guarded commands. Now, for anyone with some understanding, it is clear that as sequencing tools they are much more attractive to use than the traditional while-do and if-then-else and if, fifteen years ago, someone had thought of them, while-do and if-then-else would perhaps never have become established the way they are now. While at other places --Albuquerque and Toronto, for instance-- it sufficed to show the difference, I felt this time more or less pressed to quantify the improvement, to demonstrate that "the improvement justified the

change". I am not preaching irresponsibility, but the danger of such a climate is, of course, that you lose the ability of having day-dreams, just for fear that you can never turn them into reality. In view of this "stick-to-what-you-have" attitude it is a marvel that Niklaus managed to get PASCAL implemented at all! (The design of PASCAL itself has, of course, been heavily influenced by the local facilities and political situation. But, how else could it be?) It is, in view of the prevailing attitude of "clinging to the soil" remarkable, how the computing science there has managed to remain relatively unaffected by the awful properties of the CDC-machine they have to use. They have probably been saved by knowing its flaws very well; usually the obligation to use a poor machine ruins a computing science department. They have survived!

10th February 1975
 NUENEN

prof.dr.Edsger W.Dijkstra
 Burroughs Research Fellow

P.S. The problem that intrigued parts of my audience was the following. Consider for $X > 0$ and $Y > 0$ the following program part:

```
x:= X; y:= Y; u:= Y; v:= X;
do x > y → x:= x - y; v:= v + u
  [] y > x → y:= y - x; u:= u + v
od;
print((x + y)/ 2); print((u + v)/ 2)
```

The knowledge of Euclid's algorithms suffices to see that the first number printed is $\text{gcd}(X, Y)$; the question was to discover the functional dependance of the second number printed on X and Y and to prove it. (It is, of course, the type of "inverted question" that I detest, but letting people struggle with it makes them more receptive for the beauty and the power of the invariance theorem.)