

On the teaching of programming, i.e. on the teaching of thinking.

It is said that the murderer must return to the place of his crime. I feel myself in the murderer's position, for I found in my files an unfinished manuscript, of some seven years ago, with the ambitious title: "On the organization of one's intellect in scientific research". I quote its first paragraph:

"This is a ridiculous project. It is the kind of project for which one must be slightly drunk, before one dares to undertake it, all the time knowing that one will be sober thrice, before it will have been finished. My only excuse is, that I know its ridiculousness."

Honesty forces me to admit that since the above was written, I have been sober far more than three times and that the manuscript is still as unfinished as I had left it.

* * *

Before starting with the real subject of this talk, viz. the teaching of thinking, I must dwell for a short while at the "i.e." in the title that equates the teaching of programming to the teaching of thinking. We still find the opinion that the teaching of programming boils down to the teaching of programming languages. This misconception is not restricted to the organizers of the vocational training courses in the field and their victims. On many an application for a research fellowship I find the applicant in his curriculum proudly advertizing his fluency in, or at least experience with, a wild variety of instruction codes, programming languages and systems. An organization that in my country claims a central role in the education of programming professionals gives two courses in programming: its introductory course covers FORTRAN and some ALGOL 60, its advanced course deals with COBOL! We call such courses "driving lessons". The student's attention is almost entirely absorbed by becoming fully familiar with the idiosyncrasies of those various languages and is made to believe that the more of those idiosyncrasies he understands, the better a programmer he will be. But no one tells him that all those bells and whistles --those so-called "powerful features"-- belong more to the problem set than to the solution set. Nobody tells him that at best he will become an expert coder of trivial algorithms, and that his difficulties will never arise from the problem itself, but always from the funny way the (usually given) algorithm has to be coded.

After a number of such courses, the really tough problems are way beyond his mental horizon. His difficulties are not the ones I think worthwhile to discuss, and I shall confine my attention to the difficulties of solving intrinsically hard problems, problems that come only within reach by the time that we use an adequate, elegant mini-language and our vision is no longer blurred by the wealth of mutually conflicting "powerful features" of the more baroque programming languages. Addressing an audience of computing scientists, I assume that I don't need to belabour this point any further: by the time that all knowledge of one's programming language can conveniently be formulated on a single sheet of paper and can be gotten across the limelight in about twenty minutes lecturing, it becomes clear what "programming" really amounts to, viz. designing algorithmic solutions, and that activity requires the ability to think effectively more than anything else. Hence the "i.e." in my title.

* * *

Talking about "thinking" and, even worse, its teaching is regarded by many as something very ambitious, on the verge of foolishness, a feeling which I shared seven years ago. As the reasons for such a feeling are so obvious, some disclaimers and restrictions of the subject matter seem indicated.

First of all, I am not going to speculate how the brain works, nor am I going to suggest models for it --be they neurological, physiological, philosophical or what have you--. My interest is in using the facility and the fact that some writers of detective stories insist on referring to their hero's "grey matter" has always struck me as confusing and as a symptom of bad taste; for after all, what has its colour to do with it?

Secondly, "thinking" is a word covering such a wild variety of vague activities that we should restrict it for the purpose of our discussion. (It nearly covers non-activity as well, because it is what we are doing when doing nothing while awake!) If I were addressing a PEN-congress I might try to capture aspects of what goes on in my head when I am writing poetry, but I would like to make clear that I am aware of the fact that I am now addressing computing scientists interested in programming and its teaching: my subject is restricted accordingly.

Thirdly, we should not be deterred by the circumstance that "thinking about thinking" has a markedly incestuous flavour, full of hungry snakes that satisfy their appetite by starting at each other's, or even worse: their own, tails for dinner. Unsatisfactory as this may seem to the orderly mind, we should recognize that this world of our understanding is full of such circularities. Surface tension is explained in terms of the properties of molecules, which are explained in terms of properties of atoms, which are explained in terms of electrons and a nucleus, which is explained in terms of protons and neutrons, in the so-called "drop model" held together by some sort of surface tension. But if the circle is long enough, no one notices it and no one needs to bother: each understands his part of the chain.

In order to disguise the incestuous nature of "thinking about thinking" I shall introduce different names for distinguishable thinking activities.

* * *

I intend to use the (hopefully sufficiently descriptive) term "reasoning" for all manipulations that are formalized --or could readily be so-- by techniques such as arithmetic, formula manipulation or symbolic logic, etc. These techniques have a few common characteristics.

First of all, their application is straightforward in the sense that as soon as it has been decided in sufficient detail, what has to be achieved by them, there is no question anymore how to achieve it. And whenever such a technique has been applied, the question whether this has been done correctly is undebatable.

Secondly --and this is not independent of the first characteristic-- we know how to teach the techniques: arithmetic is taught at primary school, formula manipulation is taught at the secondary school and symbolic logic is taught at the university.

Thirdly, we are very good at doing modest amounts of reasoning. When large amounts of it are needed, however, we are powerless without mechanical aids. To multiply two two-digit numbers is something we all can do, for the multiplication of two five-digit numbers most of us would prefer the assistance of pencil and paper, the multiplication of two hundred-digit numbers is a

task that, even with the aid of pencil and paper, most of us would not care to undertake.

The amount of reasoning that we need when undertaking a non-trivial problem is often our stumbling block and I am therefore tempted to relate the effectiveness with which we have arranged our thoughts to the extent that we have been able to reduce the demands on our reasoning powers. Depending on how we attack a problem --and this is, of course, one of the main points in my whole argument-- the amount of reasoning needed may vary a great deal. Before proceeding, I would like to show you one simple example to drive home that message.

On a flat earth with a constant acceleration of gravity a cannon ball is shot away under a given angle and with a given initial velocity. Ignoring air resistance we may ask the question: at what distance from the cannon will the ball hit the ground again?

I have seen more than one professional mathematician solving this problem by deriving --with or without differential equations-- the equation of the trajectory and then solving it for x after substitution of 0 for y . Compared to the length of their calculations, "to their amount of reasoning", a relatively simple answer was produced.

I have seen schoolboys derive the same answer in a quite different manner. They were familiar with the first principles of mechanics but had the advantage of knowing nothing about differential equations, nor about analytical geometry and parabolas. The argument is then that, because the horizontal speed v_x of the cannon ball is constant (because gravity works vertically) the required answer is

$$t * v_x ,$$

where t is the time that the cannon ball travels. Conservation of energy then requires that the ball that has left the cannon with an upward speed v_y returns to the surface of the earth with a downward speed v_y , i.e. after a change in vertical speed of $2 * v_y$. The acceleration g of gravity

being given, the amount of time needed for that change equals

$$2 * v_y / g .$$

But that was the time that the ball had travelled, and therefore the required distance is

$$\frac{2 * v_x * v_y}{g} .$$

And this was the end of their derivation, completed without any computation of the trajectory, the maximum height, etc. (It was typical that all professional mathematicians that produced the latter solution without hesitation had already discovered it in their schooldays!)

Compared with the first solution, the second one is so much simpler that it is worthwhile to try to analyse, how this simplification could be achieved. The first solution uses the parabola that describes the trajectory, i.e. it gives the full connection between the horizontal and the vertical movement. The second solution, however, consists of two isolated arguments, a consideration about the horizontal movement and a consideration about the vertical movement, the two being connected by what we computer scientists would call "a thin interface", viz. the total travelling time. It is an example of a "modularized" argument; it is apparently characterized by concise modules connected via thin interfaces. (It is worth noticing that in our little cannon ball example the interface was expressed in terms of "time", a concept that hardly occurred in the problem statement! Refusal to introduce this concept or --what amounts to the same thing--: its immediate elimination, leads immediately to the timeless aspects of the problem, to the shape of the trajectory, i.e. to the clumsy first solution.)

I have used the computing term "modularization" intentionally. Too often in the past the justification of "modularization" of a large software project has been that it allowed a number of groups to work in parallel on different parts of the same project. And we also know from sad experience that unless the interfaces are indeed sufficiently thin, communication problems become prohibitively severe and that trying to solve these problems by improving the communication facilities between the groups is fighting the symptoms instead of the cause. Here, however, we see in all its clarity a quite different and more fundamental purpose of modularization, viz. to reduce the total

reasoning (irrespective of whether thereafter it will be done by different groups in parallel, or by a single one in succession).

How we choose our "modules" is apparently so critical that we should try to say something about it in general. History can supply us with many illuminating examples, I shall just pick one of them, viz. Galileo's discovery of the laws of the falling body. Since Aristotle, who had observed feathers, leaves and pebbles, heavy bodies fell faster than light ones. It was a first-rate discovery to separate two influences, the pull of gravity and the resistance of the air, and to study the influence of gravity in the hypothetical absence of air resistance, regardless of the fact that, due to the "horror vacui" this absence was not only unrealistic but philosophically unimaginable to the late medieval mind. Galileo could also have chosen the other abstraction, viz. to study the influence of air resistance in the equally unimaginable absence of gravity's pull, but he did not and the reason is quite obvious: he would not have known what to say about it!

The moral of the story is two-fold and, after some consideration, clear. Firstly, whenever an intellectual subuniverse, such as that of Galileo's falling bodies, is created, it is only justified to the extent that you can say something about that subuniverse, about its laws and their consequences. Secondly, the subuniverse must be concise and, therefore, we must have a "thin" interface. Without assumptions, laws or axioms or how you call them, one can say nothing about the subuniverse, so something must be taken into account. But it should be the minimum with the maximum gain, for the more is dragged into the picture, the less concise our reasoning that relies on it: at some stage in the game, some sort of Law of Diminishing Returns comes into action and that is the moment to stop extending the module of our reasoning.

The relevance of this type of considerations to our own trade of programming I can easily illustrate with an example from my own experience, viz. the introduction of the concept of cooperating sequential processes as a way of coming to grips with the problem of operating system design. It is one thing to decide to try to parcel out what should happen under control of an operating system as what happens under control of a set of cooperating se-

quential processes. It is quite another thing to decide to regulate their cooperation without any assumptions about their different speed ratios, not even between such processes for which such speed ratios are known well enough. But the decision is easily justified: only by refusing to take such analogue data into account could we restrict our subuniverse to one for which discrete reasoning was sufficient. The gain was two-fold: firstly, our more general considerations were applicable to a wider class of instances, secondly, the simplicity of our subuniverse made the study of phenomena such as deadlock and individual starvation feasible. Furthermore it showed the great benefits that could be derived from the availability of primitives catering for the so-called "mutual exclusion".

This idea of trying to reduce the demands on our reasoning powers is very close to a philosophical principle that since many centuries is known as "Occam's Razor": if two competing hypotheses explain the same phenomenon, the one that embodies the weaker assumptions should be preferred. (And we must assume that William of Occam knew how to compare the relative weaknesses of competing hypotheses.)

Of one thing we should always be aware. Our reasoning powers get strained by a case-analysis in which we have to distinguish between a great number of different cases. When the number of cases to be distinguished between builds up multiplicatively, they get quickly strained beyond endurance and it is almost always a clear indication that we have separated our concerns insufficiently. (From this point of view it is not surprising that many, who have thought, feel that the technique of the so-called "Decision Tables" is self-defeating. In view of their tendency towards exponential growth, the advice to use them seems hardly justified.)

In the mean time we have encountered a very different kind of thinking activity. Besides the reasoning that actually solves the problem, we have the --hopefully preliminary!-- thinking that reduces that amount of reasoning. Let us call it "pondering", thereby indicating that, ideally, it is done before the actual reasoning starts. It is, however, intended to include as well the "supervision" during the reasoning, the on-going "efficiency control".

The ability to "ponder" successfully is absolutely vital. When we encounter a "brilliant, elegant solution", it strikes us, because the argument, in its austere simplicity, is so shatteringly convincing. And don't think that such a vein of gold was struck by pure luck: the man that found the conclusive argument was someone who knew how to ponder well.

Reasoning, we know, can be taught. Can we also teach "pondering"? We certainly can teach pondering, as long as we do not flatter ourselves with the hope, that all our students will also learn it. But this should not deter us, for in that respect that subject "pondering" is no different from the subject "reasoning", for also for the latter subject holds that some students will never learn it.

Among mathematicians I have encountered much skepticism about the teachability of pondering, but the more I see of the background of that skepticism, the less discouraging it becomes. Sometimes the skepticism is no more than expressing the justified doubt, whether anyone can learn how to ponder well, but as just stated, that need not deter us: let us focus our attention on that part of the population that could learn how to ponder provided that they are taught how to ponder. I see no reason why that part of the population should be empty. Sometimes the skepticism is the result of the inability to form a mental picture of how such pondering lessons would look like, but that inability should not deter us either, for it is so easily explained. Today's mathematical culture suffers from a style of publication, in which the results and the reasoning justifying them are published quite explicitly but in which all the pondering is rigorously suppressed, as if the need to ponder were a vulgar infirmity about which we don't talk in civilized company. (And if the author has not already suppressed it, there is a fair chance that the editor will do so for him!) In the nineteenth century --read Euler, who quite openly and apparently with great delight mentioned all his hunches with what is now a surprising frankness!-- we did not suffer from such a cramped style. And as a result of this fashion to regard explicit reference to one's pondering as "unscientific", many contemporary mathematicians even lack the vocabulary in which they could describe it and this lack makes them very unconscious about their own methodology. Their way of pondering being unknown to themselves, it becomes some-

thing "unknowable" and highly personal, it becomes regarded as a gift with which someone must be "born". And here we find the third source of skepticism: the mere suggestion that pondering could be taught is experienced as a threat upon their ego.

To those who have never tried to visualize how lectures in pondering could look like and, therefore, doubt their feasibility, I can only give one advice. Suppose that you stop teaching results and solutions, but start to solve problems in the lecture room and that you try to be as explicit as possible about your own pondering. What will happen? The need to get some sort of verbal grip on your own pondering will by sheer necessity present your ponderings as something in which, as time progresses, patterns will become distinguishable. But once you have established a language in which to do your own pondering, in which to plan and to supervise your reasoning, you have presented a tool that your students could use as well, for the planning and supervision of their reasoning. In all probability it will have your own personal flavour --I hope that it will, I am tempted to add-- but this does by no means exclude that it won't help some of your students: you would not be the first to found a school of thought! They will learn in your way never to embark unnoticed on an extensive new line of reasoning; then they will learn in your way never to do so without a prior evaluation of the chance of simplification versus the chance of further complication, etc. And, eventually, when they grow up, they will learn to ponder in their own way, although the traces of your teaching will probably remain visible all through their lives.

* * *

In the above I have presented "pondering" as an optimization problem, viz. how to get the maximum result out of the minimum amount of reasoning. In doing so I followed a quite respectable tradition that presents certain forms of "laziness" as an indispensable mathematical virtue. This may surprise all those who know my profound distrust regarding the simplifications that underly the "Homo Economicus" as a picture of Man, a picture in which the self-sacrificing person (if admitted as a human being at all!) is classified as a non-interesting fool. To correct a wrong impression I may have made, let me quantify my considerations. If you would like to think effectively (because you like doing so), then you had better try to become an expert at

it! But, if you would prefer to dream (because you like having beautiful dreams), then you have better become an expert at dreaming! (Remarks to which we can add, that the two facilities do not exclude each other.)

There is a third, indispensable mental activity which, for lack of a better name, I indicate with "revolting". This is what has to take place when, no matter what we try, our maximized output remains nil, i.e. when we are stuck in an impossible task. The only effective reactions are either to change the problem until it becomes manageable, or to throw it away and to turn to something else. Obvious as this may seem, I must mention it explicitly, because experience has shown that these are difficult decisions to consider, and that, therefore, we must teach the courage to revolt.

The decision is always difficult because it has the connotation of failure --usually without justification, for most tasks are impossible anyhow--, it is, however, particularly difficult if the impossibility of the task is politically impalatable to the society of which one is a member. In serious cases the victim who has embarked on a popular, but impossible task, can hardly afford to admit even to himself its impossibility and the resulting inner conflicts may form a serious threat to one's integrity, one's health and one's sanity. For those who doubt their courage to revolt, it seems a safe precaution to stay away from popular projects, particularly if they work in an intolerant environment. For those who have learned to revolt well, the act is always one of liberation: it is an ability shared by most innovators of science.

13th February 1975
NUENEN

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow