On Programming and Mathematical Reasoning

(talk to be given for Computer Sciences freshmen of the University of Texas at Austin on 13 October 1987, 5:00 - 6:00 PM.)

Ladies and Gentlemen,

let me begin with stating that it is a great pleasure to me of having the opportunity to give a talk to you. The pleasure consists in the combination of the selected theme, Programming and Mathematical Reasoning, and you being freshmen. This might become clear as the talk proceeds. Before I start, I want to apologize to those among you who are eager to hear anything useful about particulars of the undergraduate study here at UT. Me being a Department's guest who just arrived, I am not able to provide any information of that sort. Therefore I can do what I like much more and discuss something more general.

Most of you are at the beginning of a professional career and I thought this were a good reason to give you an overview of my own professional life hitherto. Some among you might pick up something useful from it. It serves me in that it provides a stepping stone for what I really want to talk about, namely the unprecedented challenge Mathematics is faced with by the emergence of Computing Science

I was born, grew up, and received my high-school education in what was called the dark south of The Netherlands. That part of the country was almost exclusively Roman Catholic, and so was I. During my young years the Vatican had still a considerable influence on the organization of every-day life and on the way people would think, or rather: not think. At primary school, more time was spent on learning to behave like a good Roman Catholic than on learning arithmetic. I mention this because, as I realized much later, it has had an influence on my mathematical life.

My mathematical life really started at the age of 12 when I entered high school. The high school in my town was run by, and under firm control of, the Order of the Friars Minor. For the intellect at school, that Order had to resort to laity. And indeed, all teachers of mathematics were laymen. High school offered me a considerable amount of mathematics during five consecutive years. It included plane geometry, stereometry, analytical geometry, trigonometry, algebra, calculus, and mechanics. And I loved it! And I also loved mr. Pasmans, my regular teacher for mathematics. He was a gentle man who taught mathematics with passion and with austerity. I always felt I have been lucky with such a teacher.

In those days, in the late fifties, it also dawned upon the dark south, that miraculous machines, called computers, had been invented and that they were to change the face of society. (For instance would they cause a massive unemployment, because one single computer could eventually do the work of all clerks in the country. As a young boy I was intrigued by the phenomenon and by all the rumours and fairy-tales attending it. However, nobody in my environment knew something fundamental or substantial about computers. There was even no place in the country where you could study them. One year before I graduated from high school I received a golden tip from dr. Bijlsma. He was an old man, gray, tall, stately, having a croacking voice, and feared by all pupils. He was regarded as the embodiment of wisdom. His advice to me was short, clear, and without any hesitation:

"Then, study mathematics."

What an insight in those days! How much more obvious would have been a banishment towards Electrical Engineering. Many years went by before I realized that this man had his intellectual roots in the much more secular western part of our country.

So, there I went, mathematics being my cup of tea anyhow. I went to the

Department of Mathematics at the Technological University of Eindhoven (, of which I later became an employee and still am). When I arrived there, I quickly became deeply impressed by what I saw and heard, and I remained so for awfully many years. In particular the full professors would fill me with a holy respect; each of them knew much more than I would ever be able to grasp. They represented Mathematics. And mathematics being the Queen of the Sciences, they were the Kings. (In retrospect, some of them behaved like a king indeed.) At formal occasions they would show up collectively, wearing gowns. For me as a novice, all the devotion, dignity, and decorum was in perfect harmony with my Roman Catholic heart. So much for this.

The Department would offer a curriculum which would reasonably well cover what was generally felt to be Applied Mathematics. There was a main emphasis on Analysis, probably for the benefit of Numerical Mathematics, Mathematical Physics, and Theoretical Mechanics. There was a minor rôle for Discrete Mathematics, and there was Statistics and Operational Research. There was only one chair for Computing Science, occupied by prof. dr. E.W. Dijkstra, and it was a bit of a disappointment to me that Computing Science was generally regarded as not really belonging to genuine mathematics. (This did not agree with the Golden Tip I received at high school.)

Presumably the chair had been founded because -- with the advent of computers -- numerical analysis, operational research, statistics, discrete mathematics, and perhaps number theory were expected to boom. I think that in those days of spectacular advancements in electronics, Computing Science -- Computer Science, I should say -- was really believed to belong to Electrical Engineering. At my university, EE indeed had more professors of computer science than the Department of Mathematics. I must admit that I was too young then to be able to judge.

I accepted the things as they came, entered the mathematical world, and enjoyed studying very much. I also started to share -- by osmosis -- the mathematicians's view of Computing Science. It took me many years before I started to understand how big a misconception this was. The misconception has not alleviated till this very day. It is still reflected in the name of many a "Department of Mathematics and Computing Science", at least in the Old World.

There were two happy circumstances that prevented me from sharing the mathematicians's view of Computing Science completely. The one circumstance was that one summer I volunteered at the Computing Centre of The Dutch State Mines. There I witnessed dozens of people fighting a Univac III (, a mainframe in those days). There I was face-in-face with the inability of so many people to produce software in a systematic manner. Later I learned that this was typical of nearly all Computing Centres and that what I saw was the Software Crisis.

The other happy circumstance was that on the university we had to program quite a lot, mainly for the courses on numerical analysis. There I experienced the recurring inconvenience of being unable to come to grips with an evidently existing correspondence between the programs I made and the mathematical formulae from which these programs emerged. No teacher of numerical analysis and no professor at EE for that matter had given me the slightest indicati whatsoever towards resolving that inconvenience.

It got resolved ultimately, by that one professor of Computing Science within our department, in his course "A short introduction into the art of programming". But prof. Dijkstra would show and argue much more in that course. He showed how programs could be constructed in a highly systematic manner (without resorting to "a ballet of symbols"). And he would argue why computer programming was intrinsically difficult and why --by its very nature-- had to be considered as a completely new branch of applied mathematics. The ideas attracted me very much and were in accordance with what I had experienced. When I graduated from the university, in 1970, I had the extreme privilege of becoming an assistant of prof. Dijkstra. For me, this was the beginning of a period of many years of apprenticeship, collaboration, and friendship. And that period seems not to have ended yet. (Here is the place to say that to him I owe most of my understanding of Computing Science.)

The whole period in principle has been marked by just one theme: "keep it simple, please". Technically, the period falls apart in two halves. In the first half we were mainly concerned with the mathematization of the art of programming, i.e. bending the art into a firm mathematical discipline. In the second half the emphasis moved --as an inevitable consequence of our previous experiences-- towards "streamlining the mathematical argument". And that is where we are now. (The transition from the first to the second period coincides more or less --and not by accident so-- with Netty van Gasteren joining us.)

Let me briefly describe what happened. When I joined the Department of Mathematics in 1970 a few remarkable results had been achieved on the international battle field of Computer Programming. First, prof. N. Wirth had designed the programming language Pascal. Second, prof. Dijkstra had written his "Notes on Structured Programming". Third, prof. C.A.R. Hoare had just published "An axiomatic basis of computer programming".

All three people had the same goal: coming to grips with programming. Their contributions spread over the world like wildfire; they had an enormous impact and still have.

Aside I am here in Austin for seven weeks now, and I have seen all these three people on your CS Department. Isn't that a nice indication for how intellectually vivid a department you are on?
(End of Aside.)

The contributions set the scenery for the years to come and in Eindhoven they served us as a starting point towards mathematizing the art of programming. During at least seven years we made rapid progress, I think. We learned to derive programs from their functional specifications; simple ones first and then more complicated ones. We explained about programming, about our experiments, and about our experiences to all sorts of people. There was always excitement when a next beautiful little program emerged. We had a regular course on programming which was attended by third-year students of mathematics. That course was a definite source of inspiration. There we could test in detail the explainability of our findings. In general, third-year students of mathematics formed a sufficiently critical and enlightening audience. For me, it was the first time in my life that I understood the relevance of the intimate interweaving of academic research and academic educati

Now, let me dwell on these courses in some more detail. The students that would visit it had been exposed to programming courses before. There they had learned that a program had to be executed on a machine, and there they had learned to think about a program by trying to think of what happens during execution. (The jargon refers to this habit as "operational thinking".) The outcome of our research was that a program is best understood as a mathematical formulae to be derived from the program's a priori given functional specification. Also, it so happened that the mathematics of program design urged us to design, conceive, and read a program text backwards. Now imagine the students visiting our course. They had to make a U-turn in their thinking. While learning about an objectively superior method of program design, they had to unlearn at the same time what they had made themselves more or less familiar with before. And we discovered that the process of unlearning, forgetting about a past, was their major stumbling block. (Here I have to add that this past was inflicted upon these students. When I come to think of it this way, deciding on what to teach and how to teach becomes a highly delicate and risky matter.) For me, this experience was my first firm confrontation with the consequences of having acquired inadequate thinking habits. It would not take long before the second confrontation came, which was so impressive that it would change my scientific life.

As I said earlier, we made rapid progress in mastering a mathematical discipline of programming, but after seven years or so the progress started to slow down considerably. The somewhat more intricate designs we experimented with did no longer satisfy our cautiously built up standard of beauty and mathematical simplicity. In studying the phenomenon we eventually arrived at the inevitable -- and for me unthought of -- conclusion: the standard mathematical reasoning patterns -- of which I had been so proud -- were the source of the trouble. This was the beginning of a new period. We started to investigate very elementary mathematics and the way mathematicians would typically deal with it. (I remember that the use of the mathematical implication was the first to investigate.) The outcome was beyond doubt: mathematicians hardly manipulate their formulae. They interpret them instead. They interpret them in terms of a model, which in one way or another is better than the mathematical formulae. As a result they are as operational in their reasoning as the oldfashioned programmer was in programming. During our investigations we also found that on many an occasion mathematics hardly can manipulate its formulae because the notation in which they have been rendered is sufficiently clumsy to prevent any manipulation whatsoever. (Here we have the question which of the evils comes first.)

These observations evoked a drastic change in my mathematical life. Aided by an ever lasting ignorance of the mathematics of computer programming by the mathematical community as a whole, and aided by a number of unfortunate and emotional confrontations with "genuine" mathematicians, I broke with my unconditional allegiance for the Kings. Now I am learning to judge them on their merits.

Time has come to show some slides. On high school most of you have become familiar with mathematical objects called sets. Here is one

Slide 0

An arbitrary set

Usually, the egg-shaped figure (which ought to be a circle) is embedded within a rectangular frame, which represents the universe. Here we take the slide as our universe: all elements are inside the slide. Likewise, all elements of the set we consider are inside the egg. It was Venn, who invented this representation. (Who was Venn? I even don't know his first initials.)

In set theory there are two highly important sets, the most important ones in fact. One of them is the empty set.

Slide 1

The empty set.

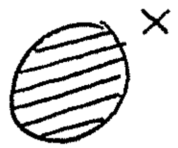
The other one is the universe.

Slide 2

The universe.

They are equal! So, here something must be wrong [sic]. The next invention of Venn is that the two sides of the egg should have different colours. In practice, this is realized by shading the egg's interior.

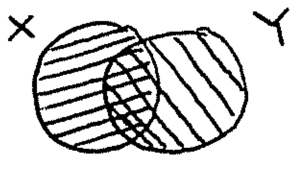
Slide 3



Venn diagram for the arbitrary set X.

With this convention, the universe is the fully shaded slide and the empty set is the empty slide. Now we are ready to consider diagrams with more than one set on it.

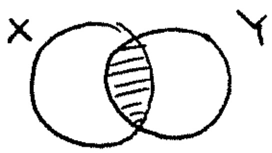
Slide 4



Two sets X and Y

We were very careful in drawing it. Had we shaded the two sets into the same direction, we would have missed the opportunity to recognize the important area which is now shaded in two distinct directions. That area is the intersection of the two sets. Here we arrive at what possibly is Venn's third invention. The contours of the set being depicted anyhow, even if they are absent in cases like the empty set or the universe, we just shade the part of the diagram which we are interested in. Because, after all, all those different shadings makes the formalism heavy going and that defeats our purposes: reasoning about sets. With this new convention the intersection of two sets can be rendered as

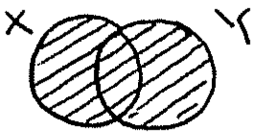
Slide 5



The intersection of two sets

Likewise we can introduce the union of two sets.

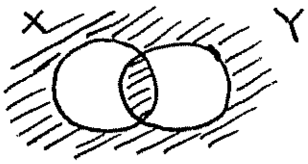
Slide 6



The union of two sets.

By now we have introduced our formalism and we have shown it worked by trying two rather familiar examples. Next, let us introduce a brandnew operation, namely the query of two sets. It is defined by

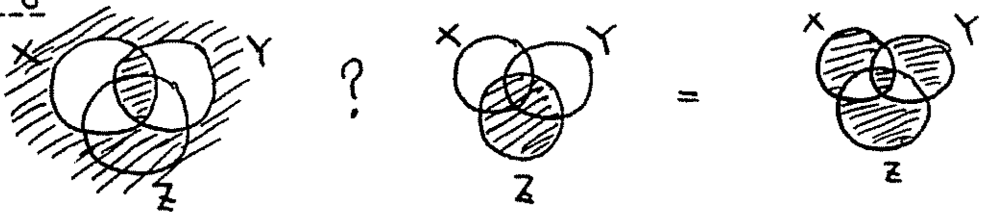
Slide 7



A Venn-definition of X ? Y

Now we can ask to prove the following theorem.

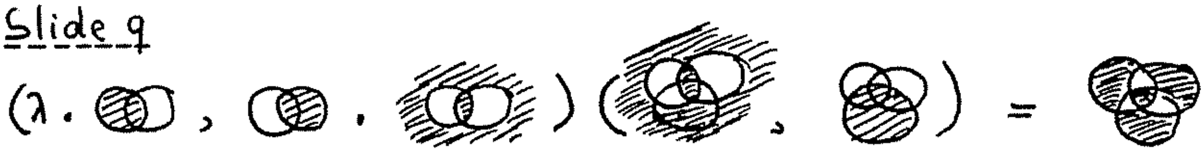
Slide 8



Venn's formalism at work.

Here I must apologize for having chosen a somewhat hybrid notation. In the pure Venn-formalism, the theorem would have looked like

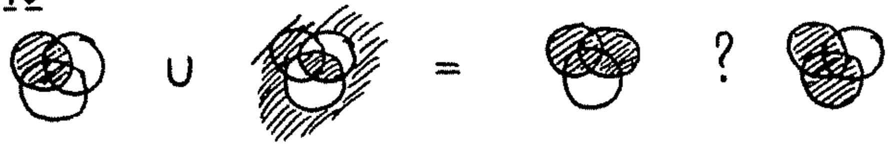
Slide 9

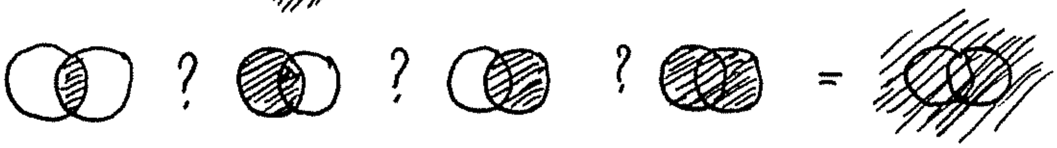


Courtesy, the Handbook on the Foundations of Illustrative Mathematics™

And what about the following theorems?

Slide 10

(a) 

(b) 

(a) U distributes over ?
(b) The Golden Rule.

I beg your pardon. Are you out of the picture already? I thought that also in English a picture was worth more than a thousand words?

Well, let me tell you this. In mathematics, pictures are not worth a penny. I could easily tell you what the query of two sets precisely is and with this off-picture information you could dig yourself a road through pictorial demonstrations of the stated theorems. But the point is that I don't want you to do that: it is almost criminal. The criminality has at least three grounds.

First, and minor, the manipulations on the pictures are presumably not well-defined. They are mechanically tedious and error-prone. Moreover, the possibilities for using them are very limited.

Second, and worse, each picture represents a specific instance of the arbitrary case. After having demonstrated each of these theorems, you might very well feel obliged to consider the cases that are not covered by the pictures you used. There are plenty of them such as two sets being disjoint, one set being contained in the other, etc..

Third, and worst, the pictures are just a model -- one particular model even -- for the subject matter, which in our example was operations on sets. Dealing with it we find ourselves traveling back and forth between the model and the mathematical formulae. And we are in imminent danger of identifying them with each other, or mixing them up, or --worse-- of regarding the pictures as reality and the formulae as their model. (Peano tried to "capture" the natural numbers by a number of axioms !)

Here you see in a nutshell what is at the order of the day among many mathematicians: they unnecessarily complicate their reasoning by interpreting their formulae. This causes mathematical reasoning to be less efficient, less precise, and more mysterious than it would have been otherwise. How much easier is it to manipulate in an uninterpreted fashion formulae from a well-designed formalism. The way we have learned to master arithmetic on primary school offers an example par excellence.

Computing Science has started to investigate Mathematical Methodology. It had to, because the discipline of programming required it and mathematicians didn't care. Programming from an operational viewpoint has already been replaced with a firm calculus for the derivation of programs. Predicate Logic has been bent into a well-oiled Predicate Calculus. The redoing of Lattice Theory has begun. The train has started moving, and I sincerely hope that some Kings on earth join the journey. If not, their dethronement might become inescapable.

And what about you, freshmen? Should you avoid courses on mathematics? For

Heaven's Sake, don't! I did not want to saddle you with a dilemma. On the contrary, I pointed out a challenge. Therefore, do take classes on mathematics. Many, and tough ones. But forewarned is forearmed. Before I leave I provide you with three concrete arms:

- if your teacher argues that something is "easily seen", then ask him HOW.
- if your teacher argues that something is "obvious" or "trivial", then ask him WHY.
- if your teacher refers to naturalness or to intuitition, then OBJECT.

Thank you for having listened to me.

Austin, 12 October 1987

W.H.J. Feijen,
Department of Computer Sciences,
The University of Texas at Austin,
Austin, TX 78712 - 1188