6 How informatics and discrete thinking return to school

Walter Oberschelp
Angewandte Mathematik und Informatik
RWTH Aachen
Germany

Abstract

We localise Computer Algebra Systems (CAS) among the technical innovations at school and emphasise the difference to artificial intelligence tools. There are two fundamental modes for CAS, the *graphic-numerical* and the *symbolic term* modes. Of these, the graphic-numerical mode is most spectacular. It supports the continuity paradigm of mathematics and induces the danger of an algorithmic demotivation of scholars. The symbolic term mode bears a strong connection to discrete thinking. It automatises term manipulation and seems to turn drill in this field into an obsolete art. We reveal its nature as a recognition process for context free languages and sketch the perspectives of term manipulation in secondary instruction.

Keywords

Algorithms, calculus, discrete mathematics, knowledge based, symbolic manipulation systems.

CHANGES IN MATHEMATICAL EDUCATION CAUSED BY NEW TECHNOLOGIES

Computer Algebra Systems among other new technologies for secondary mathematics teaching

Beyond chalk and pencil and beyond the now already classical pocket calculator, new techniques of the communication age invade the scenario of mathematics teaching. Mathematical software for geometry and many application oriented tools ranging from statistical data analysis to PROLOG software for artificial intelligence (AI) offer their services. Computer Algebra Systems (CAS) like

Information and Communications Technologies in School Mathematics
J.D. Tinsley & D.C. Johnson (Eds.)

MAPLE, DERIVE or MATHEMATICA promise to substitute most of the mathematical drill by creative activities. Furthermore, the Internet and e-mail support scientific communication world-wide; thus mathematics instruction in the classroom has a direct link to the whole community of teachers and pupils; the time of isolated work in a small group without the support through external resources is over—at least as far as the technical possibilities are concerned.

In the last few years, AI has suffered many drawbacks as the problem of efficiency has not yet been mastered. The search for efficient and deterministic deduction algorithms has only yielded inherently non-deterministic calculuses, which are controlled in most cases only by heuristic assumptions. In our opinion the most important changes occur as consequences of the new CAS facilities. We claim that their appearance alters the content and methods of mathematics teaching in a very specific way. In order to estimate this influence appropriately, we have to look at the details, and also some anthropological issues need to be discussed. In particular, the role of active and passive mastery of mathematics, the importance of intuition and of verificational abilities and the chances for a reliable assessment all appear in a new light.

New horizons for mathematical teaching

The classical tools of mathematical education are, of course, the ability to understand and, moreover, to find proofs for theorems and formulas in geometry, algebra, analysis and other fields, and to apply these results to problems which are in a close connection to these fundamentals. The drill on algorithms which was required, gave the basis for assessment—correct calculating and correct term manipulation seemed to be a reliable indicator. The arrival of the pocket-calculator was the first occasion to weaken this position. But the belief that competency in algorithmically finding the remarkable points of a function curve would be the evidence for a good mathematical understanding, was unbroken. And the ideal goal, to demonstrate mathematical strength by finding an independent way to solve an applied mathematical problem, seemed to be very often non-realistic for an average scholar.

As a consequence these classical objectives are now changing. While the understanding of the basic facts remains officially in its important position, the need for training on traditional algorithms is now no longer obvious. It would be wonderful if the time which is now saved could be used to further improve the capability for applied mathematical problem solving. But considering the doubts mentioned above, another chance is most welcome: the ubiquitous offer of complete solutions for whole problem classes with an applications interface, which does not really require an understanding of the very problem.

Actually, we get an extensive market of overview packages and of visualisation-tools, which help us to estimate or rather to get a feeling for the power of such a tool in an applicative context. Thus working within mathematics changes to management of mathematics or rather to being managed, only guided by cleverness to find appropriate packages in the Internet or elsewhere.

Thus there remain three fundamental mathematical abilities as goals of mathematical education:

- fundamental mathematical *understanding*, perhaps creativity;
- appropriate *use* and visualisation of mathematical tools, perhaps for solving applied problems; and
- competent *evaluation* (interpretation) and the ability to *discover* useful tools.

The goal of mastering algorithms and mathematical techniques has become lost, and in connection we have lost important chances for assessment. It is therefore not surprising that, even in the innermost circles of mathematical didactics, opinions arise like that of Heymann (1996) "seven years of mathematics instruction suffice". The speculation that politicians might subsume objectives like c) under the non-mathematical general sciences is therefore not absurd.

But it is erroneous to believe that mathematics must not work with algorithms, since informatics is now the home of algorithms. Surely, informatics had deprived mathematics of algorithmic responsibilities in the past but the present and the future aim of informatics is to create high level interfaces for information technology in an applicative and declarative manner and thus to make algorithmic competency obsolete. Therefore mathematics has to become again the—possibly non-popular—refuge for discrete and algorithmic competency. In fact, our students of computer science have not just forgotten how to program in an algorithmic language like C^{++}. While assembler programming is only practised by very few specialists, programming in a third generation language is done as a more or less annoying duty, if no public domain software is available. But the real merits for activities in informatics are high level achievements well above the algorithmic level. In the race for high level solutions computer scientists are well in front of mathematicians, and therefore the pendulum of algorithmic duties turns back to mathematics.

The governing paradigm of continuous models and the annoyance of the discrete

The classical content of mathematics education was based mainly on continuos models. We worked in the Euclidean plane or in 3D-space with real co-ordinates; optimisation was based on real valued functions, and the only unavoidable discrete objects seemed to be the natural numbers and the two truth values. But even number theory and Boolean logic only played a marginal role in mathematics education, not to speak about graph theory, combinatorics or other discrete theories. Of course, digital calculations and term manipulation could be considered as discrete activities, but they were never objects of mathematics instruction, only medium and tool. The appearance of the computer with its discretised technology promised a renaissance of discrete thinking at least in the mathematical foundations of computer science. But this period is coming to an end: mouse and light pen are being substituted for the keyboard as the usual input device. To press a button or to click the mouse (in the appropriate geometric

menu position) are (apart from writing) the only remainders of discrete activities of the average computer user, while the mouse seems to move in a continuous mode. Surely, this is a most natural activity from an anthropological point of view: the ability and the need of man to translate motion into action is the reason why the speech controlled or the programmed automobile has and will have no chance against the wheel-steering; and even though we are forced to discretise and to quantise information for computer processing, the trend is to make the user believe that information technology occurs in the continuous space of continuously moving pictures, where continuous motions produce actions.

It seems to me that mathematicians again get the responsibility to make man aware that there has to be a discrete world at the basis. We must not decide on speculations whether the 'real world' is discrete or continuous. Discretisation is more than a technical trick to make information processing possible (see Graham *et al.*, 1994, for excellent evidence for this claim). It is a pointer to mathematical realities which compete with continuous notions. And it seems most natural to get experience with a discrete dynamic system working with linear and non-linear recurrences rather than analysing existence theorems for solutions of differential equations; in experiencing the security of a public key cryptosystem based on number theory. Thus discrete objects turn out to be important for applications within a natural discrete model setting.

New tools to support working with discrete objects are not yet completely developed. Yet the technique of automatic graph drawing or more generally of visualising discrete objects with high flexibility is—even though in progress—not perfect compared with the facilities to visualise continuous functional information.

THE FUNDAMENTAL MODES OF COMPUTER ALGEBRA SYSTEMS

The graphic numerical mode of CAS and its revolution of calculation-practice at school

What is the role of CAS in this context? We have to distinguish two fundamental modes, in which these systems can be used (see Oberschelp (1996) for more details):

- the graphic-numerical (GN) mode;
- the symbolic term (ST) mode.

The difference between these two modes is not defined by the operating system of the computer, but only via the semantics of input and output.

CAS in the GN-mode owe their most spectacular success to their ability to represent on the screen graphs of functions, sets of trajectories or visualisations of gradient fields. In many cases they are able to show the dynamic development of those pictures according to changing parameters in the manner of a mathematical

movie. In this way we get a complete overview of a functional situation and we are dispensed of the nowadays obsolete curve discussion. There is much charm in visualising the maxima of a function *f(x)* in the GN-mode of a CAS, more than in the tedious calculations with the derivatives of *f(x)*; and also in visualising geometric configurations or curves in a way which goes beyond the somewhat anaemic ritualism of vector calculus or the theory of conics.

Of course, there are limits to the numerical power and (as consequence) to the graphical perfection in the GN-mode. According to reasons connected to numerical instability it will not be possible in the near future to plot the 10 zeroes of a polynomial of degree 10 in the complex plane and to pursue in real time a quasi-continuous change of coefficients with a mouse-like 10-finger-input device. But those limitations have no practical influence on the consequences, which we draw from the nevertheless impressive performance of CAS in the GN-mode. We say good bye without regret to the zeroes-finding-activities in maximisation. They were only based on the more or less accidental fact that the derivatives of a polynomial have a smaller degree, and it is therefore easier to calculate their zeroes. But from a numerical aspect the calculation of zeroes and of maxima has for *arbitrary* functions a similar complexity. The GN-mode now leaves to the computer the somewhat monotonous task to evaluate a function at sufficiently many arguments, such that a curve can be drawn or is simply visible in accordance with the pixel granularity of the screen.

Since the points on the screen have to be calculated with suitable precision, the GN-mode relies on the common paradigm of continuous co-ordinates. Thus a typical intermediate step to display the function $f(x) = \sin 2\pi x$ at the argument $x_0 = \frac{1}{8}$ is to calculate $f(x_0) = 0.7071068$ not $f(x_0) = \frac{1}{2}\sqrt{2}$ (the concept of an 'exact solution' is not a genuine notion in the GN-mode). It should be clear that CAS in the CN-mode have facilities for very high precision calculations as a necessary prerequisite for fast and precise graphic performance.

CAS in the GN-mode give a strong de-motivation to exercise numerical calculations by oneself. Nearly all calculation exercises for assessment tests can be done very quickly by these systems. Considering the built-in facilities for different solve-concepts it would seem to be no longer necessary to learn how to solve a quadratic equation $ax^2 + bx + c = 0$ or a (3x3)-system of linear equations. Given the parameters *a,b,c* or the coefficient matrix together with the right side as input, the appropriate solve-command will yield the result. As long as the teacher could prevent the use of the CAS-computer in the classroom (this is a non-realistic assumption), the old forms of mathematics instruction would stay valid for teaching and for assessment purposes—but no longer for homework. Most of our teachers agree that the alternative for assessment of only solving applied mathematical problems with the help of CAS would end in a disastrous failure of the average students, who rely on perfection with traditional algorithms in order to conceal their mathematical incompetence. Therefore, we have to establish

officially more general goals for education and for assessment (as described previously).

The symbolic term mode of CAS and its impact on mathematical education

The tremendous success of CAS in the GN-mode can be understood from the well known phrase

one picture says more than 1000 words

or in an appropriate adaptation

one plot tells more than 1000 table entries.

This thesis obviously favours the continuity paradigm. But for mathematics it is not the ultimate argument. Even a good and intuitive picture does not create reality. But the mathematician has to create mathematical reality from a skeleton of facts and decisions (e.g. axioms).

Therefore, there is, in contrast, a symbolic term (ST) mode which supports discrete thinking for mathematics in a certain way. Here we are working with non-trivial terms and equations (or more generally with inequalities).

That is, we don't restrict ourselves to single terms and equations such as

$$3.14159 \quad \text{or} \quad \sin 0.1 = 0.09983$$

Instead, now typical objects are term equations such as

$$\sin^2 x + \cos^2 x + 1$$

$$\sum_{i=1}^{100} i = 500 \quad \text{or even} \quad \sum_{i=1}^{N} = \frac{N(N=1)}{2}.$$

In mathematics instruction we are used to manipulating the terms according to rules and conventions which come from algebra. But most of these identities have a nontrivial legitimisation or an application-oriented meaning; term manipulation is then usually only an auxiliary step between nontrivial argumentations.

Now CAS in the ST-mode offer to perform these auxiliary steps automatically. Commands like

```
simplify, expand, factor, derive, integrate
```

in the context of a term input indicate essential services of the ST-mode. But there is more about these systems. We can expect from a system like MAPLE that it yields as an output to

$$\sum_{n=1}^{N} n^2 \quad \text{or} \quad \sum_{k=0}^{n} \binom{a}{k}\binom{n}{k}$$

the simplified results

$$\frac{1}{6}N(N=1)(2N+1) \quad \text{or} \quad \binom{a+n}{n}$$

In other cases a confirmation or a denial of a conjectured identity has to be given. We also expect the answer

$$f(x) = \frac{1}{\sqrt{1-2x-3x^2}}$$

for the input

$$\text{solve } (1-2x-3x^2)f'-(1+3x)f = 0 \text{ with } f(0) = 1.$$

Moreover, if no closed-form-solution is known or does provably not exist, we expect to receive the appropriate information. Thus a user of a CAS in the ST-mode cannot only expect help in doing intermediate term manipulation steps; the user can expect any reasonable information about the result, which he or she sets out to find or to prove. A good CAS in the ST-mode would give you the information that

$$\sum_{n=1}^{\infty} \frac{1}{n} \text{ is divergent,}$$

while a system working only in the GN-mode might stop with a certain value, e.g., with

$$21.30048\ 1502,$$

(thus indicating convergence) because the built-in-precision has told the evaluating program that summands which are less than 10^{-9} can be neglected.

Now this very feature is of revolutionary importance for all mathematicians. What is left for the person, who is working at the fundamental level, i.e., that of understanding and creatively developing ideas, who wants to find new results and proofs, if the CAS in the ST-mode gives any information which is wanted automatically?

One might give a preliminary answer in the same way, in which jogging answers the question. Why learn to run fast, if technical crafts can exceed the largest speed which is possible for man to run!

There are analogous situations. The various facilities for automatic word processing obviously do not dispense with the need for teaching orthographically correct writing. We feel that the comfort to correct every type-fault easily and immediately does not strengthen our ability to concentrate on good and correct writing. A certain kind of orthograpical jogging is asked for! Our idea is therefore to understand mathematical education as an institution to learn and to practice mathematical jogging. If we agree that secondary education needs such a jogging to a certain extent, we have to develop (and to argue for) a reasonable and indispensable training program. We have to start a general discussion about the cultural basics of our civilisation, about the aims of what the German word 'Bildung' (education) means.

DISCRETE THINKING AND THE ST-MODE

What is behind the ST-mode?

There are two obvious questions now, which should be answered in order to analyse the situation correctly:

- How is all this possible for CAS in the ST-mode? Why are all the other computer-based tools unable to do a similar job?
- What remains of mathematics teaching, if all results are delivered to us automatically by the CAS? Is there really nothing left for the humans? It seems implausible that CAS are doing everything and more of that—what happens in human mathematical thinking?

We are going to discuss the first question. How do CAS in the ST-mode work? The basic observation is that everything is strictly based on processing techniques for formal languages with the main impact on syntax-analysis (parsing) of the input. The relevant language class is that of context-free-languages (CFL) in the sense of the well known Chomsky-Hierarchy. In this framework mathematical terms can be handled as linguistic objects. The main problem is to recognise the input problem first syntactically and then in its semantic context. Then a CAS has to find an appropriate entry in a data-method-base, which contains specific mathematical knowledge. Afterwards the system generates the output by reporting the facts and applying the algorithms, which have been found. The difference from methods of artificial intelligence (AI) is important. AI tries to deduce with the general rules of logic from given assumptions. Since Gödel we know that this is a complete technique (for first order languages), but it is in general not decidable. As an indirect consequence AI-techniques are usually not efficient.

In contrast, a CAS in the ST-mode simply uses the facts and applies the results (e.g., algorithms), which have been found in the method-data-base and which usually have been discovered by eminent mathematicians. In this way, a CAS will neither find, e.g., the well known proof of divergence for the harmonic series nor the well known trick of the little Gauß to sum the numbers from 1 up to 100. Instead, a CAS has to recognise the harmonic series or an arithmetic series with certain parameters as the input, and then look for the right answer within the stored information. The CAS can only print out the proof idea, if it is stored as a text in the database.

These examples explain the enormous success of CAS compared to AI: The whole mathematical knowledge of mankind, more exactly all those results and algorithms, which have been stored and implemented in the CAS, are available, if the input problem can be identified appropriately by the system.

The basic technique, which is used here, is the so called *unification*, which tries to fit terms together, if they "only differ by a substitution". Unification is the main technique of the theory of term rewriting systems, which is a well developed branch of informatics. AI systems also work with unification in their basic

resolution technique. But the terms, which are handled by AI systems, are of a very elementary kind, since AI does not (yet) possess the power to incorporate arbitrary auxiliary theories.

We give some more characteristic application examples. If the input is

$$\text{solve} \sum_{n=1}^{\infty} \frac{1}{n^2},$$

then a CAS in the GN-mode will output 1.64493407, while in the ST-mode the system has to recognise that this is an instance of Riemann's Zeta-function, and the output has to be

$$\zeta(2) = \frac{\pi^2}{6}$$

If we sum in the same way the inverse cubes instead of squares, then we can expect in the ST-mode an answer like

$$\zeta(3) = 1.202056903 \quad \text{approximate value, result is irrational}$$

The latter remark relates to Apéry's famous proof from 1981; his result that $\zeta(3)$ is irrational, has to be stored as a fact in the data base.

There are, of course, serious technical problems with a successful syntax analysis. The system must, for example, be able to try different substitutions in order to recognise $\sum_{n=1}^{\infty} \frac{1}{n^2}$ as the same problem as $\sum_{i=1}^{\infty} \frac{1}{n^2}$.

The success of the system usually depends on the abilities of the user. If the user is able to inform the system that some kind of Dirichlet series is connected with the input, the system has to extend its unification test only to a part of its data base and no tests for membership of subclasses is required in advance.

There may arise serious theoretical problems for successfully working with term-systems. Only in the simplest cases there exist normal formulas for terms, to which each input can be uniquely reduced. For polynomials and for rational functions in one variable those normal forms are available (using the degree-concept and the technique of partial fractions). Here commands like `expand` or `factor` yield unique results. But the factors can in general not be written using the symbolic radical—it is well known that this is only possible if the degree of a polynomial is less than 5.

A normal form for *nested* terms of root-radicals is unfortunately not available (see Davenport *et al.*, 1988, for the subsequent discussion). Similar problems arise with trigonometric terms. Here the non-existence of the so called Church-Rosser-normalisation-property raises problems. Therefore we lack a technique to verify systematically the famous Ramanugan-identity:

$$\sqrt[3]{\sqrt[5]{\frac{32}{5}} - \sqrt[5]{\frac{27}{5}}} = \sqrt[5]{\frac{1}{25}} + \sqrt[5]{\frac{3}{25}} - \sqrt[5]{\frac{9}{25}}.$$

And even the systematic deduction of the identity $\sin 2x = 2 \sin x \cos x$ from a database, which contains only $\sin(x+y) = \sin x \cos y + \sin y \cos x$.

This is a serious problem for a CAS, since the system has not only to find the substitution $x = y$ —which is easy—but also to use the identity $x + x = 2x$ which is not obvious in the present context.

In the context of secondary mathematics instruction it is important that symbolic differentiation of function terms does not give rise to difficulties. And the problem of indefinite integration is completely settled by Risch's theorem. It is decidable which integrations lead to elementary results, and here the results are found by an algorithm.

Of course, CAS are by no means absolutely fool proof. If the user is unable to formulate his task precisely (which very often happens in connection with the commands `simplify` and `solve`), the CAS usually try heuristic default techniques. But often the user is disappointed by the output, because it is not the type of answer which was expected. Thus the ambitious user of the ST-mode has to possess a serious background in mathematics, and very often the lay person fails, particularly if it is only a case of simply "throwing the problem into MAPLE".

ST-Mode automatisation and the mathematical mind

The foregoing discussion has revealed some answers concerning the human aspects which might be missing in the ST-mode. These answers are necessary, since otherwise we would enter a similar motivation crisis concerning our own thinking in categories of term manipulation, as we have experienced for the GN-mode, where we scarcely escaped with our jogging philosophy.

For the ST-mode we state its inability to create and to use mathematical concepts in an intuitive way. These systems essentially imitate and copy mathematical knowledge, admittedly with a remarkable intelligence to find out what the issue is.

Therefore, we usually don't get help from the system for *finding* proofs. Instead, the system betrays final results and very often it spoils the mathematical clout of an argument.

But we have to be cautious and should not fall into the trap of anthropocentric self-overestimation. Is there an invariant notion of an authentic and genuine human mathematical argumentation?

Since mathematics as a deductive and living science develops further and further, surprising new methodological inventions may happen. One of the most spectacular methodological inventions of the last years was, in my opinion, the Wilf-Zeilberger (WZ) technique, which finds proofs and new identities for 'hypergeometric formulas' (see Petkovsk *et al.*, 1996, for the present state of the discussion). An example is the famous Dixon identity

$$\sum_{k=-\infty}^{+\infty} (-1)^k \binom{n+b}{n+k}\binom{n+c}{c+k}\binom{k+n}{b+k} = \frac{(n+b+c)!}{n!b!c!}$$

The WZ-theory has been completely integrated into, for example, MAPLE, and it serves as a decision procedure for conjectured hypergeometric identities. Here the

results are not stored in the knowledge database of the system, only the *method*; this situation can be interpreted as a passage to a very high level of AI. In addition, WZ offers verification tools (certificates) for correct formulas. Therefore a handmade control of correctness is possible, and nobody has to trust the honesty of the system blindly. But even though we understand the theory, usually these certificates don't support our 'original proof intuition' for hypergeometric formulas. We are used to accepting proofs, which are based on induction or on certain bijection arguments. But the appearance of a 'strange and less transparent' verification technique will not keep these negative attributes forever. It is just the trademark of great mathematics, that it starts with a flare of mystery and finally works as a well accepted, elegant and steady source of creative ideas. A good example here is the residual calculus of complex analysis. Initially under the odium of an extremely non-elementary theory, now most mathematicians classify proofs with the residual technique as easy, elegant and obvious.

Nevertheless, the big complexity gap between *verification* of a proof and *finding* a proof seems to be an intrinsic mathematical fact. The fundamental conjecture of complexity theory that $P \neq NP$ gives a somewhat precise formulation for our experience that it is much easier to teach students a bulk of knowledge using verification techniques—this is the usual procedure of university mathematics teaching—than to make them find a single theorem. We can teach in one semester more mathematics to an average student, than centuries of ingenious mathematical research could find. And the main achievements of the ST-mode concern verification! Keeping therefore in mind that the creative production of mathematics has to be rewarded more so than professional verification, a substantial portion of human motivation should survive the CAS area. As teachers we will have to decide in the future again and again about the appropriate distribution in the trade-off between the professional use of mathematical verification tools and the mathematical fitness in the mind of a student.

A final word concerning discrete thinking

We have argued over and over that working with *discrete objects* has to be(come again) an important part of school mathematics, even though these objects are often hard to distinguish in a very-high-level application interface; that they can be grasped constructively, add to a certain scientific realism and are last but not least essential features of our mathematical jogging program. And for the future of CAS we hope for even better and more comfortable practice with discrete objects.

But discrete *thinking* is more than working with discrete *objects*. It also means thinking in and working with distinct linguistic units, which amounts essentially to argue in general term systems. The language of mathematical reasoning is a system of discrete utterances (quite different, for example, from the continuity potential, which is hidden in the language of music). The differentiation and integration of function terms is a purely discrete (term-

oriented) activity! This way of discrete thinking has been implemented partly in CAS in the ST-mode. It seems to be clear now, that a minimum of discrete-thinking-competency is required in order to use such a system appropriately. With respect to this qualification, mathematics education has the task of teaching basic mathematical reasoning, which is a well controlled and an intuitively guided calculus of discrete linguistic operations. This kind of discrete thinking is indispensable for wide areas of mathematics instruction and a counterpart to functional and continuous thinking, which is like steering a continuous device to an appropriate position.

REFERENCES

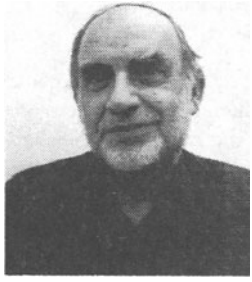
Davenport, J., Siret, Y. and Tournier, E. (1988). *Computer Algebra*. London: Academic Press.

Graham, R., Knuth, D. and Patashnik, O. (1994). *Concrete Mathematics*, 2nd Edn. Reading (Mass.): Addison-Wesley.

Heymann, W. (1996). *Allgemeinbildung und Mathematik*. Weinheim, Basel: Beltz-Verlag.

Oberschelp, W. (1996). Computer-Algebra Systeme als Implementierung symbolischer Termalgorithmen. In H. Hischer and M. Weiß (eds.) *Rechenfertigkeit und Begriffsbildung*. Proceedings of a workshop in Wolfenbüttel, 1995. Bad Salzdetfurth b. Hildesheim: Franzbecker, 31—37.

Petkovsek, M., Wilf, H. and Zeilberger, D. (1996). *A=B*. Wellesley, Massachussets: A. K. Peters.



Walter Oberschelp was born in Herford, Germany, studying in Göttingen, Tübingen and Münster, gaining his PhD in Mathematical Logik in 1958. After holding professorships at Münster, Hannover and Illinois, he was appointed to his present post as Professor of Applied Mathematics and Informatics in Aachen where he has developed the Informatics Department and is engaged in scientific work. He has delivered courses on informatics, discrete structures and algorithms. Walter is also an advisor for many student teachers in mathematics and informatics, giving courses in high school instruction for both students and teachers.