Helena Durnová

Change in progress: Notation and algorithms

In: Eduard Fuchs (editor): Mathematics throughout the ages. Contributions from the summer school and seminars on the history of mathematics and from the 10th and 11th Novembertagung on the history and philosophy of mathematics, Holbaek, Denmark, October 28-31, 1999, and Brno, the Czech Republic, November 2-5, 2000. (English). Praha: Prometheus, 2001. pp. 186–195.

Persistent URL: http://dml.cz/dmlcz/401249

CHANGE IN PROGRESS: NOTATION AND ALGORITHMS

Helena Durnová

Abstract

This is a by-product of my research project. The topic was stimulated by the 1999 Novembertagung theme. It is not exactly philosophy of mathematics and its influence on its history, it is more connected to language and culture. Being a "by-product", this article uses terminology a bit vaguely and the reader is asked for understanding and for reading it not as a "fact file", but rather as "fiction". The terms are not precisely defined, attention is brought to the "feeling", or perhaps "idea".

Introduction

When studying articles and books on discrete optimization and graph theory, I was amazed by the number of terminologies used - not really different sets of terms, but overlapping vocabularies of the mathematicians active in the field, different meaning for the same words. There even does not have to be a great time difference between the publication times of the pieces – one is not so much surprised to find out that what was called "analysis situs" a hundred years ago is now called "topology" or "graph theory". In this case, the changes take place in a life-time. What are the reasons for (making and deepening) these differences? A perhaps even more important question is: what does this lead to?

A definite shift towards the formalisation of the language of discrete optimization problems can be observed. It is not to be concluded that *all* branches of mathematics undergo this particular process. However, some changes would probably be observed also elsewhere. Further thoughts circle around the role of language for understanding mathematics and for the communication between mathematicians. What are the forces that rule out some modes of expression and establish new ones?[1]

[1]One could also speculate about the role that natural language (of a mathematician) plays. Is there a connection between our natural language and the ability to do mathematics? Is there a "national" mathematics? These interesting questions, however, are not to be answered here.

What is mathematics

Before trying to answer some of the above-mentioned questions, it should be explained what we will mean by mathematics and the role of language in mathematics and/or for mathematicians. The first thoughts are therefore devoted to the question: *What is mathematics?*

Mathematics might be seen as a MODEL, TOOL, GAME, SCIENCE, ABSTRACTION, etc.[2] It depends on the individual which definition he or she finds the most accurate. The two notions important in this context will be

- *Mathematics as Model or Tool:* Mathematics describes the reality and in a certain way helps us in solving "real-life" problems (such as finding the minimum spanning tree …).

- *Mathematics as Abstraction:* To employ mathematical methods, we need to take into account not concrete things, but rather abstract notions.

Language is connected to both mathematics as a model/tool and to the abstraction in mathematics. Language itself is a comparatively abstract notion, and the language of mathematics is no less abstract then the natural languages.[3] Just as natural language changes with reality, the language of mathematics changes with regard to the "mathematical reality", i.e. adapts itself to the changing needs of mathematics (and, of course, mathematicians). In the following, I will try to present a demonstration of what happens. The topic chosen for demonstration is discrete optimization.

The core of this paper is denoted by the following scheme. Again, the readers are invited to make their own conclusions and judgement as far as the accuracy of this scheme is concerned.

[2] The reader is invited to argue whether these terms express what mathematics really is, and is also invited to add his or her own "definitions".

[3] It can be argued which of the two is more abstract – I would be in favour of the view that mathematical language is the more abstract of the two.

WORDS	GRAMMAR
NOTATION	ALGORITHMS
SYMBOLS	PROCESS

LANGUAGE
MATHEMATICS

Discrete Optimization and Algorithms

Discrete optimization is a relatively new area of mathematics. Its problems are relatively well-known. Some of them are:

1. Minimum spanning tree
2. Shortest route
3. Travelling salesman
4. Chinese postman
5. Matching
6. Network flows

For these problems, some kind of automated procedure would be worthwhile. This procedure is called an *algorithm* – one should rather say, *nowadays* it is called so. As these procedures will be central to this paper, a short "detour" about algorithms and their general history follows.

Algorithms: A Very Short History

Algorithm is nowadays a widely used word in mathematics. It is especially used with computers nowadays, but even EUCLID (ca. 365–300 B.C.) devised a kind of algorithm for determining the greatest common divisor. The name itself is said to be derived from the name of the Arabian mathematician AL-KHWARIZMI (ca. 783–850 AD). According to SCHREIBER [3], the word "algorist" was used as the opposite to "abacist" for the new way of counting. In 1849, a German orientalist suggested that the name comes from the name of an Arabian learned man, AL-KHWARIZMI. Since the 1950s, the word algorithm has been widely spread in mathematical papers.

With algorithms, we can speak about several aspects:

- design
- proof of correctness
- search for machines performing algorithms
- transformation of problems for algorithms
- NP-completeness

In different times, different notions of algorithm were used. For example, the proof of correctness would not be appropriate for ancient Egyptians. NP-completeness lures out only because of a repeated failure to find a suitable algorithm. However, there is one thing that is central to any algorithm: the necessity of working with *abstract* notions. Without them, it would be impossible to design an algorithm for more than one specific problem.[4]

Change in Progress: New Structures

New approach to problems, new technigues, higher level of abstraction – all that requires new structures, in which we can think. More precisely, these structures can be divided into Words, Grammar, and Language. How have these developed in the 20th century mathematics? Examples from discrete optimization should throw some light onto this.

Words

Changing the vocabulary of any language is performed very easily: there is no need to point out "real-life" examples. In mathematics, as most probably also elsewhere, new terms are introduced either for new objects, or for objects already existing. The former is quite understandable: new objects need new names. The latter is a bit more complicated: were the old names insufficient? If not, why do mathematicians devise new terms? Perhaps it is a matter of recognition of *same things that arose in different contexts*. Perhaps people love naming things . . . . As a result, however, parallel terminologies could be developed, which may cause communication error. Also, language "dislikes" absolute synonyms, and therefore the two terms adopt slightly different meanings.[5]

[4] A further step of this is the transformation of the problems, one into another, which especially arises in the form of dual problems.

[5] For example, minimum spanning tree formulations appear in algebraic, geometrical, or graph theoretical languages. In graph theory in general, the same notion can

Parallel development of terminologies could eventually lead to parallel development of theories. In mathematics perhaps more than in any other science, the definitions are crucial – therefore if mathematicians do not recognise the terminology, they would not read the articles or books. "Re-inventing the wheel" is a direct consequence of this.[6]

Grammar

The development of a grammatical system is not so straightforward, which is true also for natural languages, where the grammatical changes come up slowly (structures cease to be used and new ones come up in order to express new facts or situations). In natural languages, interference with foreign languages can often be traced (either conscious or subconscious). In mathematics, the structural part of grammar is of utmost importance. What did the Egyptians do? They *described* their solutions, word by word. How did the Romans multiply, using the Roman numerals? This would be a hard task even today. Examples are multiple. Perhaps using symbols for unknowns could be considred one of the crucial changes.

When it comes to discrete optimization, the terms and methods have already been fairly formalized – we are talking about the 20th century here, some time after the Hilbert programme. In spite of that – more precisely, because of that – this example can give us the feeling of *how* the changes happen. The time-span is "only" a lifetime of one person, and therefore could help us understand the speed with which things happen in the structural field.

At the beginning of the 20th century, the language of "algorithmic" papers was still a bit story-like, descriptive, written in the "write-it-as-you-talk" style. This is especially demanding (and time-consuming) for the reader. It is not systematic, in the sense that the paper starts with if on the first page, then a long row of *then*'s follows, until eventually on the tenth page, you find the *else* for the first if.

---

easily have three different names, seemingly signifying different things: branch, arc, edge.

[6] As has been pointed out, theories could definitely have "intersections", where they work in the same way in practice. However, it is not certain whether the theories are context-dependant; by this I mean the fact that some assumptions and corollaries are easy in one theory, but difficult in another theory. As an example, the minimum spanning tree will do: what Borůvka did in matrix terminology, PRIM did in graph-theoretical terminology. However, could Borůvka have formulated the theorems of graph theory in matrix theory, if graph theory is a "subset" of matrix arithmetics? Probably no — he would not think of such applications.

Later on, in the 1950s, mathematicians started using step-by-step descriptions of algorithms. It is much easier to understand the structure of the algorithm. The small step towards some kind of "pseudo-code" was made in roughly twenty years. The expressions in pseudo-code are standard, which results in faster reading – if one understands the expressions and is familiar with the structure.[7]

New syntax of mathematics tends to be more that of an analytic language (as opposed to syntactic language) – i.e. to a language where the word order is of crucial importance.

Language

The new terminology and structure needs some superstructure – this structure can then become a new branch of theory. As an example, we can take the algorithm and complexity theories. They also start to have their own problems, their own results – both applicable to more theories, perhaps also more abstract then the theories just *using* algorithms.

Polynomial algorithm and the search for it thus became central to thoughts of the algorithm theorists (and practitioners). The luring creatures – NP-hard or NP-complex problems – are more widely known results of the research. In a certain way, they can be seen as a kind of new alternative of *squaring the circle*, *doubling the cube*, or *trisection of an angle*.

Game, or a Science of Its Own?

These two words, game and science, reflect the crucial quality of discrete mathematics. Many problems in discrete mathematics seem to have their origin in puzzles or curious questions. Here are some of them:

1. Euler, 1735: The Seven Bridges of Königsberg
2. Hamilton: Round-the-World Trip
3. Euler: 36 Officers
4. Kirkman: Fifteen Schoolgirls

The starting point is fun – however, the results interpreted well need not be so unimportant. In the following, I will try to show how simple the formulation was for some discrete optimization problems, and what the results were.

[7] By saying this, I would like to suggest that there are two sides to each coin.

Abstraction in Discrete Mathematics

In the following, more precise examples of abstraction are given.

Minimum Spanning Tree

In his solution to MST, BORŮVKA uses more abstract way than those who attempted to solve it after him. He uses a matrix of real numbers - and in his mathematical paper, he does not even explain why! He just states the problem: *In each row, a number must be chosen. The sum of such numbers should be minimum.*[8]

What is the meaning of numbers in BORŮVKA's table? It is exactly the same as the distance between points in a plane for JARNÍK, or the edge weights for KRUSKAL, PRIM, or DIJKSTRA. The missing word(s) – weight, edge – were substituted for by greater abstraction in BORŮVKA's solution. What is worth noticing, however, is the fact that Prim also uses some form of matrix. In his paper, however, this seems to be motivated by the use of computers (and not by a missing term).

By the same token, it can also be claimed that the *maximum* spanning tree can be constructed using the same, only a little adapted, procedure.

Shortest Paths

Finding the shortest route is a task solved by people – consciously or subcosciously – almost every day. What we want to spare are our feet, time, fuel, ... This simple task can be used in some more complex procedures. This is another typical feature of mathematics, by the way: always trying to solve subproblems of the given problem, or by using analogies.

Travelling Salesman Problem

The travelling salesman problem is interesting from the point of view of abstraction, developing new fields, but also mysterious: it is one of those problems that cannot be solved in polynomial time. Problems related to the TSP were solved also without direct relation to hamiltonian circles – whence there exist two distinct definitions of the problem, one for hamiltonian circuits and one without.

[8] The reason I am using his paper is that it is the oldest one I have on the subject.

Mathematics: Changing Picture

Theorem-Proof

A mathematical theorem, when correctly proved, is true forever. [9] Theorems about Eulerian and Hamiltonian graphs are true: they need not always give us the solution, they, however, always decide – somehow.

Algorithm-Analysis

If we have an algorithm, we have to say: first, whether it solves our problem correctly, and second, in what time we can expect the solution. The second task gave rise to a whole new field of complexity and computability theory. It seems that this direction of mathematics can be separated from the Theorem-Proof one. The only thing I am not sure of is whether it will separate or whether there already exist two kinds of mathematics, in some way. In the following, some features of the two parts, demonstrated on "pure" and "applied" graph theory, are suggested.

Roots of Graph Theory

It apears that graph theory has grown up from two distinct roots. We can divide it basically into two parts: *pure*, solving problems related to topology, and *applied*, solving problems with economical background.

As for the first, "pure" part, it can be said that problems are usually formulated on non-weighted graphs. When we consider problems such as eulerian and hamiltonian graphs or the four-colour problem, it might surprise us that the origins of the problem are in curious questions.

This, however, is not true for the other graph theory, the "applied" one. The motivation to the problems here is often very practical: let us name just the shortest electrification network for the minimum spanning tree and the school-bus routing for the travelling salesman problem. The travelling salesman problem as well as the shortest-paths problems show the motivation even in their names.

Sometimes, a *naive approach* could work when designing an algorithm. This kind of approach tells us to choose the "locally best" solution at every step. This works for example for the MST algorithms, and also for matching algorithms.

Later on, especially with the discoveries of NP-completeness of problems and after having "reasonable" algorithms for the easier problems,

[9] However, proofs have not always be as valid as this. [1]

heuristics were devised. A *heuristics* is a procedure that gives a result which may or may not be optimal. After each step, we ask whether we got a better solution, and after a certain criterion is fulfilled, the procedure is ended. What is important when judging a heuristics is *how near* to the optimum solution we get and *how fast* we get there.

Differences between the "pure" and "applied" graph theory

Graph theory, perhaps as any other branch of mathematics, has its more theoretical and more practical parts. It would probably be more precise to talk about the aspects of "purity" and "application" in graph theory, because for some problems, it is not easy to state which category they belong to. The following scheme tries to show the differences between the two.

speed	precision
possibility of solution	finding the best solution
somehow	this way
computability	elegance
reasonably good	optimal
greedy approach	paradoxes
applied	pure
approximate	precise
obscure	clear

… and a question: can we decide which is good and which is bad?

The Beauty of Discrete Mathematics

The focus on more specific problems gives rise to new areas of research. Thus, in connection with discrete optimization, we can see the emergence of complexity theory. These new theories offer space for parallels with popular interpretation of the problems. Thus, in confusion of "not solvable in polynomial time" with "not solvable at all" we can see a parallel with people still trying to square the circle, trisect an anlge, or double the cube.

The problems in discrete optimization – and in the discrete mathematics in general – are usually nicely put. The task seems to be so easy; yet the solution can be *extremely* difficult. Such a minor change that we have to pass through each *vertex*, not *edge* exactly once – and from an easy game, here is a problem without a simple solution. Similar change in the MST problem leads to TSP, which is not solvable in polynomial time – which might be, by some "over-educated" people, interpreted as *insolvable*, as the following extract shows: [2]

> *The traveling salesman problem recently achieved national prominence when a soap company used it as the basis of a promotional contest. Prizes up to $ 10,000 were offered for identifying the most correct links in a particular 33-city problem. Quite a few people found the best tour.* [...] A *number of people, perhaps a little over-educated, wrote the company that the problem was impossible–an interesting interpretation of the state of the art.*

Conclusion

I feel there is no real conclusion to this article. Instead of a conclusion, I will try to pose three questions: Do we need to put the isolated solutions together, or is there something good in keeping things separate? Discrete and continuous in mathematics: are they different notions, or just two aspects of the same thing? Does a new language enable us to do more?

References

[1] Henrik Kragh Sørensen, "Theorems with exceptions and the use of proof." In: D. Beckers, K. Peters, C. Vollmers (eds.), *Proceedings of the 9^{th} Novembertagung*, Nijmegen 1999.

[2] Little, J. D. C., Murty, K. G., Sweeney, D. W., and Karel, C., "An algorithm for the travelling salesman problem." *Operations Research* **11** (1963) pp. 972–989.

[3] Schreiber, P., Algorithmic thinking through ages. In: Ivor Grattan-Guinness, ed., *Companion Encyclopedia of the History and Philosophy of the Mathematical Science*, 2 volumes, Routledge 1994, pp. 686–693.

Helena Durnová
Department of Mathematics
University of Technology
Technická 8
616 00 BRNO
Czech Republic
e-mail: durnova@dmat.fee.vutbr.cz