THE APPEARANCE OF ALGORITHMS IN CURRICULA A NEW OPPORTUNITY TO DEAL WITH PROOF?

Modeste Simon, Institut Fourier, Université Joseph Fourier, Grenoble.
Ouvrier-Buffet Cécile, UPEC & Laboratoire André Revuz, Université Paris 7.

We deal with the concept of algorithm which is taking importance in curricula in many countries. In particular, we develop an epistemological analysis of this concept and discuss its place in the mathematical science and the link it has with proof. This analysis is enriched by the study of "how researchers know the algorithm". We conclude with implications of the changes in curricula on proof learning.

Discrete mathematics represents a mathematical field which takes a growing importance in our society. In particular, the accessibility of the concepts of this field brings new tracks to teach and learn proof (e.g. Grenier & Payan, 1999; ZDM, 2004). This paper will deal with one of the concept of this field: the algorithm. Indeed, with the omnipresence of computers and technologies in our society, it seems that algorithm will take more and more importance in curricula and it raises questions that impacts on the teaching of mathematics. Moreover, new types of proofs involving computations appear, and with them, algorithmic proofs for instance and the philosophical and epistemological questions of the use of computers to create and/or validate proofs (Hanna, 2007). The 1998 Yearbook of the National Council of Teachers of Mathematics was entirely dedicated to the questions of algorithms and was "an attempt to answer many of [the questions provoked about the place of algorithms in today curricula] and to stimulate other questions that all of us in mathematical education need to consider as we continually adapt school mathematics for the twenty-first century" (NCTM, 1998, p. vii). Recently, the studying of algorithms got into the class of mathematics in the French curriculum of the secondary. The appearance of the concept "algorithm" in the mathematical curriculum questions the role in mathematics of an object which seems, at first look, to belong more to computer science. Actually, algorithm is first of all, from historical and epistemological points of view, a mathematical concept.

The links between algorithm and proof are not easy to describe and have not been much studied epistemologically. The goals of our study (started in Modeste, Ouvrier-Buffet & Gravier (2010)) are here twofold: from a mathematically-centered perspective, we want to bring an epistemological analysis of the algorithm which emphasizes its interplay with proof. The way researchers in mathematics and computer science know the concept of algorithm allows a validation of the epistemological model we develop in this paper. From a didactical perspective, we ask how the concept of algorithm can enhance the curriculum, focusing on links between algorithm and proof (avoiding the ubiquitous computer aspect). We ultimately want to build Research Situations for the Classroom (RSC) (Grenier & Payan, 1998, 1999; Godot & Grenier, 2004) involving algorithm and proof processes. In particular, the design of such situations implies to rely on an epistemological analysis and a study of the practises of researchers (see Knoll & Ouvrier-Buffet, 2006). Then the concluding session highlights the links between the concept of algorithm and the proving process and brings new research tracks in order to design and to analyze situations for the classroom.

EPISTEMOLOGICAL ASPECTS OF THE CONCEPT OF ALGORITHM

A common definition

A usual definition of algorithm is presented by Knuth who takes algorithm as an object of study (and also questions the differences between mathematical thinking and algorithmic thinking):

> "(...) an algorithm is a set of rules or directions for getting a **specific output** [1] from a **specific input**. The distinguishing feature of an algorithm is that **all vagueness must be eliminated**; the rules must describe operations that are so simple and so well defined that they can be **executed by a machine**. Furthermore, an algorithm must always terminate after a **finite number of steps**." (Knuth, 1996, p. 59)

This implies that an algorithm solves a specific *problem* P by returning in the *output* the answer corresponding to the instance of P given in the *input*. It is important to remember that the input and output information are coded and "algorithms deal primarily with the manipulation of symbols that need not represent numbers." (Knuth, 1996, p. 61)

The previous definition also shows the *effective aspect* of algorithm: no ambiguity must exist in the instructions so that any *operator* – most of the time a computer, and in this case the algorithm can be described by a *program* – gets the same output with the same steps. The *finiteness* cannot be dissociated from the notion of algorithm, as Chabert notes it in his book on the history of the algorithm:

> "Today, principally because of the influence of computing, the idea of **finiteness** [1] has entered into the meaning of algorithm as an essential element, distinguishing it from vaguer notions such as process, method or technique. […] Here we have a finite number of operations, a finite number of input values, but also a finite number of solution procedures, that is that each step should be able to be carried out by a finite process – something which is not possible, for example, in determining the quotient of two incommensurable real numbers. We also refer to an **effective** procedure, that is one that will effectively achieve a result (in a finite time)." (Chabert et al., 1999, p. 2-3)

This finiteness raises questions regarding *complexity*: given an input, how many steps does the algorithm take to answer? How much space does it need to store the involved information? These questions respectively deal with *time complexity* and *space complexity*. This complexity depends on the size of the input and can be studied from two points of view: the *worst-case complexity*, the maximum complexity of the algorithm for an input of size *n* and the *average-case complexity*, the average of the complexity for an input of size *n* (usually, the inputs of size *n* are considered to have the same probability). Since Knuth (1996), complexity is a specific and fundamental aspect of the algorithm which is absent from the other fields of mathematics. He says (about Bishop's mathematics [2]) that “[it] is constructive, but it does not have all the ingredients of an algorithm because it ignores the “cost” of the constructions” (Knuth, 1996, p. 110).

Here, we have detailed three important aspects of algorithm: the link with **problems**, the **effectivity** and the **complexity**.

Ambiguity of this definition

As examples of algorithms, authors often give Euclid's algorithm for gcd, arithmetic operations on integers, algorithms for sorting or algorithms for shortest paths in a graph... Among all these examples, one strikes us: the method to find the roots of the quadratic equation $ax^2 + bx + c = 0$ using the discriminant $b^2 - 4ac$.

With more details, the algorithm is the following:

Input: a, b, c
$\Delta = b^2 - 4ac$
if $\Delta > 0$ **then** return $\frac{-b+\sqrt{\Delta}}{2a}$, $\frac{-b-\sqrt{\Delta}}{2a}$
else if $\Delta < 0$ **then** return $\frac{-b+i\sqrt{-\Delta}}{2a}$, $\frac{-b-i\sqrt{-\Delta}}{2a}$
else return $\frac{-b}{2a}$

Judging by the definition we gave above, we could say that it is an algorithm. But we find it surprising that such a method was chosen as an illustration of the concept “algorithm”. Indeed, in each case, the algorithm is just a formula. Moreover, from the point of view of the “complexity”, such an algorithm is not interesting, as the complexity is independent of the size of the input (we are not speaking here about the complexity of the arithmetic operations involved in the discriminant, which are for us better examples of algorithms). This example raises the question of the border between algorithmic and non-algorithmic areas and the “usual” definition is ambiguous about this. For our study, for a didactical purpose, it would be useful to distinguish this kind of formula with “real” algorithms.

A more theoretical definition

In the beginning of the 20th century, the quest of foundations for mathematics caused mathematicians to give a more theoretical definition of algorithm.

> “The works of Gödel inspired the research of Alonso Church, Stephen Kleene, Alan Turing and Emil Post. These mathematicians attacked Hilbert's Entscheidungsproblem and showed that there were, indeed, undecidable problems, that is mathematical statements for which no procedure exists by which it can be decided if the statement is true or false. To do this, each of them defined a concept of computability, that is a concept of algorithm.” (Chabert et al., 1999, p. 457)

Two of these concepts should be quoted: the *Turing machine* and the *recursive functions*. This theoretical work leads to a classification of problems depending on whether there exists an algorithm to solve them or not (*undecidable* and *decidable problems*) and if they are "easy" or "hard" problems, which means if they can be solved by a polynomial algorithm or not (we refer here to *P* and *NP-hard problems* [3]). This point of view will constitute a fundamental aspect of algorithm, the **theoretical models**.

Algorithm and proof

Algorithm and proof interplay in many ways, it will be another important aspect for us. First, an algorithm has to be proved; more precisely, it is necessary to prove its *correctness* (i.e. it gives the expected answer) and its *termination* (i.e. it always stops after a finite number of steps). And, once it is proved, an algorithm can be used as a *step in another proof*. Actually, all the aspects raised previously have a link with proof. In particular, correctness and termination correspond respectively with *problem solving* and *effectivity*. The *complexity aspect* involves proof too, and studying the complexity of an algorithm often needs substantial mathematics. The same is true of *theoretical models*, which only make sense in a proof process. Just like any mathematical object, the algorithm raises questions involving proof. But some of them are specific and only the algorithm raise the mathematical questions previously mentioned. Moreover, the algorithm is not only linked with proof on that way. An algorithm can also be a *tool for proving* a property, and for a given problem, an algorithm will give a *constructive proof* of its resolution (e.g. Euclid's algorithm provides a proof of the existence of the gcd of two integers and an effective way to compute it). Conversely, an algorithm often lies under a constructive proof and it can be interesting to formulate this algorithm clearly. For instance, from any *proof by induction* follow a recursive algorithm.

Algorithms, seen as proofs, allow to deal with two kinds of problems: *existence problems* and *testing a property*.

Recently, a link has been pointed out between proof and algorithm, with the computer-assisted proofs, that is the use of algorithms to build proofs which are much too long to be verified by a human being. For instance the four-color theorem has been proved this way. However, the algorithm has to be proved in order to validate the mathematical result. This new kind of demonstration asks philosophical and epistemological questions about the nature of proof.

Tool-Object

The aspects of algorithms underlined above can be divided into two parts since they refer to algorithm as a tool or as an object. Looking at the algorithm as an object means studying questions of validity, of complexity and description of algorithms. Looking at the algorithm as a tool is focusing on the use of algorithms to solve problems. Among the aspects discussed here, the **effective aspect** and **problem aspect** refer to the algorithm as a **tool** whereas the **complexity aspect**, the **theoretical models** and the **link with proof** refer to the algorithm as an **object**.

MATHEMATICIANS' POINT OF VIEW ON ALGORITHM

This analysis of the algorithm concept is mainly theoretical and it would be interesting to compare it with the reality of mathematics, that is the ongoing research.

- Are mathematicians' representations in accordance with our epistemological study of algorithm?
- How do algorithms interplay in their practice of research?
- Which aspects of algorithm are involved in mathematical research and which ones are not?
- Do researchers refer mostly to the algorithm as a tool or as an object?
- Do these questions depend on their field of research?

Actually, the main point which interests us is the following: validating our epistemological analysis when comparing the descriptions of mathematicians of coming to use and to know algorithm and our epistemological model (in the same way that Burton did). Here, the form of our interviews does not permit to describe the whole conceptions of the researchers in a specific theoretical model. Right now, the trends in our results are enough to use this analysis as a preliminary work in order to build Research Situations for the Classroom (RSC).

Interviewing researchers

To answer the previous questions, we chose to interview researchers both in applied and fundamental mathematics. We also interviewed researchers from fields at the intersection of mathematics and computer science, like operational research, combinatorics, computational geometry... These researchers have a mathematical activity too, that is a proof activity, but should have a rich and different vision on the algorithm, provided by the links they have with computer science. We interviewed 22 researchers. From their point of view, they belong to the following fields:

	Fundamental	Applied
Mathematics	10	7
Computer Science	5	5

Table 1: Distribution of the researchers [4]

In fact, in order to study the practice and the representations of mathematicians, the tool of interview seemed to be the most convenient possibility (in the same way as Burton, 2004). We have met the researchers face-to-face and the interviews were audio taped. The researchers received a brief questionnaire to provide personal information (name, gender, function, discipline (mathematics, computer science; fundamental and applied mathematics [5]), research subjects, teaching level at university), and an excerpt of the new French curriculum with the appearance of algorithms [6]. We followed a list of questions (see above) in conversational style. These interviews last between 20 and 30 minutes and the part which interest us here makes up around half of this time.

The interviews

In the interviews, we chose to bring up three points:

1) What is an algorithm, how can one define it and recognize it, and what examples can be given?
2) How is it used in mathematics and in the field of the researcher?
3) The place and the role of algorithms in the researchers' work.

The questions were the following:

Q1: How would you define what an algorithm is?
Q2: How can one recognize an algorithm?
Q3: Give examples of algorithms, of non-algorithms. Is the discriminant (presented like above) an algorithm? Where is the border between algorithmic and non-algorithmic areas?
Q4: What are algorithms useful for? What are their roles in your field?
Q5: Are there algorithms in your personal research? Where?

The first three questions aim at making the researcher talk about the definition(s) of algorithm. Giving examples, counterexamples and thinking about the discriminant should make the researchers question their own definition of "algorithm". The last two questions aim at making them evoke the role and place of algorithms and more particularly in their field. All the questions were reformulated if necessary, so that there was no misunderstanding. And as it was an open discussion, these points were not been necessarily mentioned in this order.

Analysis of the interviews

In order to study the transcriptions of the interviews, we built an analysis grid based on our epistemological study. The goal was to find in the interviews which aspects were present. The difficulty was to give the grid a good granularity: the purpose was to associate each idea of the researcher to one or more precise aspect of algorithm, but many aspects were often vaguely mentioned. For example, as far as complexity is concerned, most of the time, the researchers just spoke about its importance but did not give details about the different kinds of complexities or about the question of optimality. After different draft versions of an analysis grid, we finally decided on the following:

TOOL					OBJECT								
Problem		Effectivity			Complexity		Theoretical models			Proof			
Solve a problem	Input/Output	Any operator and non-ambiguity	Computer or Program	Finiteness	Complexity (Space/Time)	Optimality	Turing Machine and other models	(Un)decidables problems	P/NP	Correctness	Termination	Tool for proving	Induction

Table2: The analysis grid

Results

Validation of the epistemological analysis

In the interviews, all the aspects we expected researchers to speak about were mentioned by them. Obviously the researchers did not all talk about every aspect and they did not always give many details. But each aspect has been mentioned enough to confirm our study. We will discuss in details how each aspect has been brought up.

Domination of the algorithm as a tool

The "effective" aspect and the "problem" aspect were mentioned by all the researchers. That means that the "tool" aspect is very important in their representation of the algorithm. The "effectivity" and "problem" aspects mainly appear in the definitions of algorithm researchers gave. The "effectivity" aspect is often linked with the use of computer, and many researchers pointed out the importance of computers for algorithms (and some of them made the confusion between algorithms and programs).

Definition of the algorithm

As the definitions researchers gave very often involve the effective aspect of the algorithm, we can say that their definitions are very close to the "usual definition" we mentioned at the beginning. To illustrate this, here are some definitions of "algorithm" given by the researchers:

> "A sequence of instructions which enable from an input to produce an output."
>
> "A finite chain of steps which can be described, and which allows to compute or find the solution of a given problem."
>
> "An effective process which allows to achieve a calculus or an automatic deductive task."
>
> "An automatic method to solve a problem which does not need any human intervention, and which is workable for a machine."

About the discriminant, most of the researchers answered that, according to their definition, it was an algorithm. Some others felt embarrassed that their definition encompassed the discriminant and explained that it was an algorithm but had no interest:

> "It is not a very rich algorithm, not a good example."
>
> "We can compute it explicitly. It is just applying a formula."
>
> "We are between the "method" and the "algorithm". There is no real process. If any time we have a formula we consider it as an algorithm, it reduces the meaning of algorithm."
>
> "The problem has a set size... for me, in the idea of algorithm, there is an aspect of variable size, there is the complexity behind..."

The problem of the ambiguity of the definition we underlined seems to be shared by some researchers. The "complexity" aspect seems to be closely related to this problem. Most of the researchers must not have noticed this because, as we will see below, the notion of complexity is not of a big importance for them.

Presence of the "proof" aspect

The "proof" aspect, which is for us the most important when looking at the algorithm from a mathematical point of view, has been brought up by about half of the researchers (12 among 22). There is a link with their field of research: indeed, most of the researchers who brought up the proof consider themselves as fundamental researchers (in maths or computer science) whereas the majority of the others consider themselves as applied mathematicians or computer scientists. We can say that the importance of proof activity has a link with fundamental questions. Moreover, among all the parts of the "proof" aspect, the most quoted is that an algorithm is a tool of proof, that is to say that the algorithm is associated with the notion of constructive proof. The notion of proof of an algorithm (correctness or termination) has not been mentioned much, and more precisely it is always the correctness which was quoted.

The "complexity" aspect

The "complexity" aspect was not mentioned by all the researchers, only 11 of them spoke about it. In this case, it seems to be linked with the computer science field of research: among the 10 (self declared) computer scientists, 8 raised the questions regarding the complexity involved by the algorithm. We can infer that the complexity is not really important from the mathematicians' point of view (as the quotation of Knuth about Bishop's mathematics let us think).

The theoretical models

Theoretical models were mentioned by only 8 researchers, not only from fundamental research but mainly from fields at the intersection of mathematics and computer science (computational geometry and topology, operational research, combinatorial optimization, graph theory or cryptology). In fact, theoretical models for algorithms have a very important role in these fields. That must be the reason why those researchers mentioned them. However, it should be noted that very few researchers from other fields (only two) brought up those theoretical models. It seems that this "recent" aspect of the algorithm (but older than the link with computers!) is not known by the researchers or does not seem important to them. The first possibility is, according to us, the most plausible. As an interviewed mathematician underscored:

> "As far as I'm concerned, I'm aware of these questions [about algorithm and theoretical models] because my husband is a computer scientist. But this is not in mathematicians' culture..."

CONCLUSION AND PERSPECTIVES

Our epistemological analysis has been validated by the interviews. Hence, we can ask ourselves what this study implies for the teaching of algorithm and the teaching of proof. These questions have their importance at the moment in France, but also widely impact on mathematical education. We saw that if one does not want to teach algorithm as a tool only, but also as an object, it cannot be separated from the “proof” or “complexity” aspects. Learning algorithm seems to be a good way to learn proof, judging by the connections there exists between these two concepts.

Learning from these interviews, it seems that researchers in mathematics and in fields which link mathematics and computer science, do not have a wide view of the algorithm concept. We can say that little is known about this concept. We could explain this by the recent development of the study of algorithm, but this still seems pretty worrying. We can assume that this lack of knowledge about algorithm is shared by teachers of mathematics (at least in France) and their training curriculum has to be questioned.

This study of the algorithm in mathematics should allow us to study curricula and textbooks of mathematics of the secondary in order to know if the “tool” and “object” aspects are involved.

We would also like to study how the algorithm can be handled as an object by pupils and how it can make them enter in a proof process. We already made experimentations about this, at the beginning of university and in the training of primary teachers, and we obtained promising results (the students and pre-service teacher training were able to build several algorithms and their proof, see Modeste, Ouvrier-Buffet, Gravier, 2010). Our goal is to carry on these kinds of experiments in the secondary level. Schuster (2004) has studied combinatorial optimization problems in the secondary and has obtained very positive results about pupils' skills in manipulating algorithms and proving. We could work on the “Konigsberg's bridges problem” studied by (Cartier, Moncel, 2008) or on other problems studied by the “Maths à Modeler” team, but from an algorithmic point of view. The analysis of the way one can deal with algorithms and proof in the classroom and the results of such experiments will be the object of a new article, based upon the epistemological model developed in this paper.

NOTES

1. Bold types added.

2. In this article, Knuth studied books from many mathematicians and notice that, for him, Bishop's mathematics where the most close to the algorithmic thinking. However, even in Bishop's mathematics, Knuth noted that the notion of complexity was absent.

3. The definitions of P and NP-hard problems are not exactly these ones. For more details about theoretical models, one can read: Hopcroft, J.E., Motwani, R., & Ulmann, J.D. (2007). *Introduction to Automata Theory, Languages, and Computation*. Pearson Education.

4. The total number is not 22 because some researchers consider that they have 2 fields of research.

5. The choice of the researcher is not necessary dichotomous. Indeed, the presentation for their choice of the discipline was the following:

Mathematics □□□□ Computer science

Fundamental Research □□□□ Applied research

6. The analysis of this excerpt by the researchers and their answers to questions about teaching the algorithm will not be discussed here.

REFERENCES

Burton, L. (2004). *Mathematicians as enquirers. Learning about learning mathematics*. Kluwer Academic Publishers.

Cartier, L., & Moncel, J. (2008). Learners' conceptions in different class situations around Königsberg's bridges problem. *Proceedings of ICME 11*, Mexico. Web available: http://tsg.icme11.org/document/get/758

Chabert, J.-L. et al. (1999). *A History of Algorithms From the Pebble to the Microchip.* Springer.

Godot, K., & Grenier, D. (2004). Research situations for teaching : a modelization proposal and examples. *Proceedings of ICME 10*, Copenhague.

Grenier, D., & Payan, C. (1998). Spécificités de la preuve et de la modélisation en mathématiques discrètes. *RDM* 18/1, 59-100.

Grenier, D., & Payan, C. (1999). Discrete mathematics in relation to learning and teaching proof and modelling. In Schwank, I. (Ed.), *Proceedings of CERME 1*, Vol. 1, 143-155. Osnabrück.

Hanna, G. (2007). The ongoing value of proof. In Boero, P. (Ed.), *Theorems in school: from history, epistemology and cognition to classroom practice*, 3-16. Sense Publishers.

Knoll, E., & Ouvrier-Buffet, C. (2006). Discussing the Challenge of categorising mathematical knowledge in mathematics research situations. *Proceedings of CERME 4*, 332-341.

Knuth, D.E. (1996). *Selected Papers on Computer Science*. Center for the Study of Language and Information - Lecture Notes.

Modeste, S., Ouvrier-Buffet, C., & Gravier, S. (2010). Algorithmique et apprentissage de la preuve. *Repères IREM*, n°79, 51-72.

National Council of Teachers of Mathematics. (1998). *The Teaching and Learning of Algorithms in School Mathematics* (1998 Yearbook of the NCTM).

Schuster, A. (2004). About traveling salesmen and telephone networks combinatorial optimization problems at high school. *ZDM*, 36(2), 77-81.

Discrete mathematics and Proof in the High School (2004). *ZDM*, 36(2), 44-84 and 36(3), 82-116.