# Mathematical Literacy as a Condition for Sustainable Development

Luís Soares Barbosa[1(✉)] and Maria Helena Martinho[2]

[1] HASLab - INESC TEC & Informatics Department, Universidade do Minho, Braga, Portugal
lsb@di.uminho.pt

[2] Cied - Institute of Education, Universidade do Minho, Braga, Portugal

**Abstract.** Argumentation and proof are two main ingredients in strategies for developing mathematical skills and structured reasoning. This paper reports on a research project aimed at 'refactoring' school Mathematics in other to achieve a higher degree of mathematical literacy. In a sense this builds on a number of 'lessons' learnt from the practice of Computing Science. We further argue that mathematical fluency, broadly understood as the ability to reason in terms of abstract models and the effective use of logical arguments and mathematical calculation, became a condition for democratic citizenship and sustainable development.

## 1 Introduction

*We must give industry not what it wants, but what it needs*
— E. W. DIJKSTRA, quoted in the program of his birthday symposium, Austin, Texas, 2000

Critical infrastructures in modern societies, including those related to finances, health services, education, energy and water supply, are critically based on information systems. Moreover, our way of living depends on software whose reliability is crucial for our own work, security, privacy, and quality of life. This places the quest for software whose correctness could be established by mathematical reasoning, which has been around for a long time as a research agenda, at the centre of a debate which is no longer a technical one. Actually, for IT industry correctness is not only emerging as a key concern: it is simply becoming part of the business.

Collaboration with João Ferreira, at Teeside University, and Roland Backhouse, at Nottingham University, in the context of the MATHIS project, is greatly acknowledged. The authors express their gratitude to Paulo Silva who developed the teaching scenario on logarithms from which some examples discussed in the paper were taken. This work is partially funded by ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT (Portuguese Foundation for Science and Technology) within projects MATHIS, FCOMP-01--0124-FEDER-007254, and Professional Practices of Teachers of Mathematics, PTDC/CPECED/098931/2008.

A. Cerone et al. (Eds.): SEFM 2012 Satellite Events, LNCS 7991, pp. 64–77, 2014.
DOI: 10.1007/978-3-642-54338-8_5, 

Companies are becoming aware of the essential role played by proofs and formal reasoning in this process. At present, at least in what concerns safety-critical systems, *proofs pay the rent*: they are no more an academic activity or an exotic detail.

This places serious challenges to higher education and training programmes for future software engineers and IT-professionals. On the one hand, there is a growing demand for highly skilled professionals who can successfully design complex systems at ever-increasing levels of reliability and security. On the other, and in general, IT-driven societies also require from people a higher degree of *mathematical literacy*, i.e., the ability to resort to logic to build models of problems and reason effectively within them. Such an ability is at the heart of what it means *to understand* and it may be considered a fundamental condition for democratic citizenship. Either directly, by supporting the implementation of high-assurance information infrastructures, or indirectly, by empowering citizenship, mathematical literacy became, in a broader perspective, a main ingredient for promoting sustainable development.

Actually, skills as basic as the ability to think and reason in terms of abstract models and the effective use of logical arguments and mathematical calculation in normal, daily business practice are on demand. This concerns not only highly skilled IT professionals, who are expected to successfully design complex systems at ever-increasing levels of reliability and security, but also specialised workers monitoring, for example, automated plants and computer aided manufacturing processes.

Even more it concerns, in general, everyone, who, surrounded by ubiquitous and interacting computing devices, has an unprecedented computational power at her fingers' tips to turn on effective power and self-control of her own life and work. Neologism *info-excluded* is often used to denote fundamental difficulties in the use of IT technologies. More fundamentally, from our perspective, it should encompass mathematical illiteracy and lack of precise reasoning skills rooted in formal logic.

Irrespective of its foundational role in all the technology on which modern life depends, Mathematics seems absent, or invisible, from the dominant cultural practices. Regarded as *difficult* or *boring*, its clear and ordered mental discipline seems to conflict with the superposition of images and multiple *rationales* of the post-modern way of living. Maybe just a minor symptom of this state of affairs, but *mathphobia*, which seems to be spreading everywhere, has become a hot spot for the media. Our societies, as noticed by E. W. Dijkstra a decade ago, are through an *ongoing process of becoming more and more "amathematical"* [12]. On the surface, at least.

Under it, however, Mathematics is playing the dominant role, and failing to recognize this and training oneself in its discipline, will most probably result in people impoverished in their interaction with the global *polis* and diminished citizenship.

In such a context, this paper aims at contributing to the debate on strategies for achieving a higher degree of *mathematical fluency*, which the authors strongly think to be a condition for sustainable development in the years to come. By this we do not have in mind the exclusive development of numerical, operative competences, but the ability to resort to the mathematical language and method to build models of problems, and reason effectively within them. Our claim is that such strategies should be directed towards *unveiling* Mathematics by rediscovering the relevance of both

– *argumentation* skills, understood as the ability to formulate and structure relationships, justifications and explanations to support an argument;
– and *proof*, as the formal certification of an argument, which encompasses the effective development of proof design and manipulation skills.

Although both aspects are often emphasized separately, the development of educational strategies to bind them together in learning contexts may have an impact in empowering people reasoning skills and, therefore, their ability to survive in a complex world.

Section 2 frames the paper in the context of the MathIS project [9,10], a Portuguese research initiative on reinvigorating mathematical education, coordinated by the first author. A main component of this project is concerned with refactoring school mathematics, which is illustrated in Sect. 4 through an example on the development of calculational proofs. Before that, however, Sect. 3 characterises our conceptual framework on argumentation and proof. Finally, Sect. 5 concludes and enumerates some current research concerns.

## 2 The MathIS Project

The need for policies capable of reinvigorating Mathematics education and its effective application at all problem-solving levels was the starting point of a research project lead by Universidade do Minho in Portugal: the MathIS project.

The project was launched in 2009, aiming at exploiting the dynamics of algorithmic problem solving and calculational reasoning in both maths education and the practice of software engineering in an integrated way. The project's overall approach stems from two decades of research on correct-by-construction program design which brought to scene a whole discipline of problem-solving and shed light on the underlying mathematical structures, modelling and reasoning principles. A most relevant consequence has been the systematization of a calculational style of reasoning which, proceeding in a formal, essentially syntactic way, can greatly improve on the traditional verbose proofs presented in natural language.

A main contribution of MathIS, at the educational level, was an effort to reframe a collection of themes in pre-university mathematics along these lines and assess its merits not only on the development of general calculational and algorithmic skills, but also as a tool for discovery (see [9,10] as well as João Ferreira's Ph.D dissertation [8]). Recall, for example, that it was the formal manipulation of Maxwell's equations that led to conjecturing the existence of electromagnetic waves, confirmed experimentally shortly afterwards.

On the technology side, MathIS capitalizes on recent developments and increased flexibility in Human-computer interaction technology, to provide an infra-structure for the envisaged methodological shift. In this context, a second axis in MathIS concerned the development of innovative computer-based tools exploiting Tablet PC technologies in order to provide learning environments oriented to calculational reasoning and algorithmic problem solving [18]. These principles, although consistent with traditional blackboard-style teaching, can benefit from the enhanced facilities provided by computers.

## 3 Argumentation and Proof

### Argumentation

Mathematical learning requires a stepwise construction of a reference framework through which students build their own personal account of mathematics in a dynamic tension between previous and newly acquired knowledge. This is achieved along the countless interaction processes taking place in the classroom. In particular, the nature of the questions posed by the teacher may facilitate, or inhibit, the development of argumentation and reasoning skills [4]. A student who is given the opportunity to share her intuitions, conjectures and previous knowledge, as well as to explain the way she thought about a problem, will develop higher levels of mathematical literacy in the broad sense proposed in Sect. 1. Team work, which entails the need for each participant to expose his views, argue and try to convince the others, is an excellent strategy to achieve this goal.

Strategies which call students to analyze their arguments and identify its strengths and weaknesses are also instrumental to this aim [15]. Reference [17] singles out a number of basic issues in the development of what is called a *reflexive* mathematical discourse: the ability to go back (either to recover previous arguments in a discussion or to introduce new view points) and the ability to share different sorts of images supporting argumentation (e.g., sketches, tables, etc.).

Training argumentation skills is not easy, but certainly an essential task if one cares about mathematical literacy in modern societies. The teacher's role can not be neglected. She/he is responsible for stimulating a friendly, open discussion environment [1], avoiding rejection and helping students to recognize implications and eventual contradictions in their arguments to go ahead [11,21]. Her role is also to make explicit what is implicit in the students' formulations [6], helping them to build up intuitions, asking for generalizations or confronting them with specific particular cases.

The following opening statement of Paul Halmos' autobiography [14] is particularly elucidative, written as it was by a mathematician, who in the 1950's, at the University of Chicago, was director of doctoral studies in what was then one of the top Mathematics Departments of the world: *I like words more than numbers, and I always did (...) This implies, for instance that in Mathematics I like the conceptual more than the computational. To me the definition of a group is far clearer and more important and more beautiful than the Cauchy integral formula.*

Often in school practice conceptual disagreements are avoided (let alone encouraged!), with negative effects in the development of suitable argumentation skills. On the contrary, such skills benefit from exposition to diverse arguments, their attentive consideration and elicitation, as empirically documented in, e.g., [23]. Actually, classroom interactions can shape the mathematical universe of students. School mathematics is an iceberg, of which students often only sees what emerges at surface (typically, definitions and procedures). Rendering explicit what is hidden under the water is the role of effective mathematical training in argumentation.

## Proof

If the development of suitable *argumentation skills* is a first step to a Mathematics-aware citizenship, mastering *proof technology* is essential in a context where, as explained above, *proofs pay the rent*. Such is the context of software industry and the increasing demand for quality certified software, namely in safety-critical applications. But what contributions may Computing Science bring to such a discipline? And how could they improve current standards in mathematical education?

As a contribution to a wider debate, we would like to single out in this paper the emphasis on the central role of *formal logic* and the development of a *calculational* style of reasoning.

Clearly, Computer Science fostered a wider interest in applied logic. A simple indicator is the almost universal presence of a course on formal logic in every computing undergraduate curriculum. Proficiency in mathematics, however, would benefit from an earlier introduction and explicit use of logic in middle and high school. Note this is usually not the case in most European countries; the justification for such an omission is that *logic is implicit in Mathematics and therefore does not need to be taught as an independent issue*. Such an argument was used in Portugal to eliminate logic from the high-school curriculum in the nineties. The damage it caused is still to be assessed, but it is certainly not alien to the appalling indicators in what concerns the country overall ranking in mathematics education [20].

High-valued programmers are heavy users of logic. At another scale, this is also true of whoever tries to use and master information in modern IT societies: the explicit use of logic enables critical and secure reasoning and decision making. On the other hand, a heavy use of logic entails the need for more concise ways of expression and notations amenable to formal, systematic manipulation.

The so-called *calculational style* [3,22] for structuring mathematical reasoning and proof emerged from two decades of research on *correct-by-construction* program design, starting with the pioneering work of Dijkstra and Gries [7,13], and in particular, through the development of the so-called *algebra of programming* [5]. This style emphasizes the use of systematic mathematical calculation in the design of algorithms. This was not new, but routinely done in algebra and analysis, albeit subconsciously and not always in a systematic way. The realization that such a style is equally applicable to logical arguments [7,13] and that it can greatly improve on traditional verbose proofs in natural language has led to a systematization that can, in return, also improve exposition in the more classical branches of Mathematics. In particular, lengthy and verbose proofs (full of *dot-dot* notation, case analyses, and natural language explanations for "obvious" steps) are replaced by easy-to-follow calculations presented in a standard layout which replaces classical implication-first logic by variable-free algebraic reasoning [12,22].

Let us illustrate with a very simple example what we mean by a *calculational* proof. Suppose we are given the task to find out *whether* $\log_a(2) + \log_a(7)$ *is greater than, or lesser than* $\log_a(3)+\log_a(5)$. The 'classical' response consists of first formulating the hypothesis $\log_a(2)+\log_a(7) \ \leq\ \log_a(3)+\log_a(5)$ and then verifying it as follows:

| | | |
|---|---|---|
| (1) | function $\log_a$ is strictly increasing | |
| (2) | $\log_a(x \times y) = \log_a(x) + \log_a(y)$ | |
| (3) | $14 < 15$ | |
| (4) | $14 = 2 \times 7$ and $15 = 3 \times 5$ | |
| (5) | $\log_a(14) < \log_a 15)$ | by (1) e (3) |
| (6) | $\log_a(2) + \log_a(7) < \log_a(3) + \log_a(5)$ | by (2), (4) e (5) |

The proof is easy to follow, but, in the end, the intuition it provides on the problem is quite poor. Moreover, it is hard memorize or reproduce. Most probably it was not made, originally, by the order in which it is presented. This may explain why, in general, this sort of proofs, although dominant in the current mathematical discourse, fails to attract students' enthusiasm.

Consider, now, a *calculational* approach to the same problem. The main, initial difference is easy to spot and has an enormous impact: its starting point is not an hypothesis to verify, formulated in a more or less diligent way, but the original problem itself. The proof starts by identifying an unknown $\square$ which stands, not for a number as students are used to in school mathematics, but for an order relation. Then it proceeds by the identification and application of whatever known properties are useful in its determination. The whole proof, being essentially syntax driven, builds intuition and meaning.

$$
\begin{array}{ll}
& \log_a(2)+\log_a(7) \ \square\ \log_a(3)+\log_a(5) \\
= & \quad \{\ \text{function } \log_a \text{ distributes over multiplication.}\ \} \\
& \log_a(2\times 7) \ \square\ \log_a(3\times 5) \\
= & \quad \{\ \text{routine arithmetic.}\ \} \\
& \log_a(14) \ \square\ \log_a(15) \\
= & \quad \{\ 14 < 15 \text{ and function } \log_a \text{ is strictly increasing.}\ \} \\
& \ \square \quad \text{is} \quad <
\end{array}
$$

Empirical evidence gathered within MATHIS suggests the systematization of such a calculational style of reasoning can greatly improve on the way proofs are presented. In particular, it may help to overcome the typical justification for omitting proofs in school mathematics: that they are difficult to follow for all but exceptional students.

## 4 Refactoring School Mathematics

A main objective set for the MathIS project was the 'refactoring' of several pieces of school mathematics, systematically introducing the sort of *proofs by calculation* illustrated in the previous section. Although it is too early to draw general conclusions (preliminary results, however, appeared in [8,10]), this effort shows how the formalization of topics arising in different contexts results in formulae with the same *flavour*, which can be manipulated thereafter by the same rules of the predicate calculus, without reference to a 'domain specific' interpretation in their original area of discourse. This is the essence of formal manipulation, and yields proofs that are shorter, explicit, independent of hidden assumptions, easy to re-construct, check and generalize.

### An Illustration

To illustrate the direction of such a 'refactoring' let us consider a few examples related to the use in school mathematics of definitions by universal properties, as one is used to in program calculus (see, for example, [5]).

We begin with the simple definition of the pairing function. Its *explicit* definition looks rather obvious

$$\langle f, g\rangle\,(c) \;=\; (f\,c, g\,c)$$

but is not so easy to handle in calculations. Suppose students are asked to show that a function which builds a pair is a pairing function, i.e.

$$\langle \pi_1 \cdot h, \pi_2 \cdot h\rangle \;=\; h$$

where $\pi_1, \pi_2$ are, respectively, the first and second projection associated to the Cartesian product. A typical proof is as follows. Suppose $ha = \langle b, c\rangle$. Then,

$$\begin{array}{ll}
 & \langle \pi_1 \cdot h, \pi_2 \cdot h\rangle a \\
= & \quad \{ \text{ pairing definition, composition } \} \\
 & \langle \pi_1(ha), \pi_2(ha)\rangle \\
= & \quad \{ \text{ definition of } h \} \\
 & \langle \pi_1\langle b, c\rangle, \pi_2\langle b, c\rangle\rangle \\
= & \quad \{ \text{ definition of projection functions } \pi_1 \text{ and } \pi_2 \} \\
 & \langle b, c\rangle \\
= & \quad \{ \text{ definition of } h \text{ again } \} \\
 & ha
\end{array}$$

Refactoring this proof involves replacing the explicit definition of a pairing function given above, by a *property* which characterises its behaviour completely. Therefore, define

$$k = \langle f, g \rangle \;\equiv\; \pi_1 \cdot k = f \;\wedge\; \pi_2 \cdot k = g$$

Notice that in this property $\Rightarrow$ gives *existence* and $\Leftarrow$ ensures *uniqueness*[1].

With this definition the envisaged proof becomes trivial:

$$\begin{aligned} & h = \langle \pi_1 \cdot h, \pi_2 \cdot h \rangle \\ \equiv \quad & \{ \text{ universal property with } f := \pi_1 \cdot h,\ g := \pi_1 \cdot h \ \} \\ & \pi_1 \cdot h = \pi_1 \cdot h \;\wedge\; \pi_2 \cdot h = \pi_2 \cdot h \end{aligned}$$

This shift from *explicit* to *implicit* definitions lead, usually, to simpler and smaller proofs, rid of unnecessary variables and more general, in the sense that they can be replicated in different situations and corners of the mathematical experience.

Let us now come back to logarithms and investigate what can be proved directly from the very basic property which records the primitive fact that the logarithm is the inverse of the exponential function. Formally,

$$\log_a(x) = c \;\equiv\; a^c = x \tag{1}$$

To prove *cancellation*, i.e. that $a^{\log_a(x)} = x$, it is enough to instantiate variable $c$ with $\log_a(x)$, therefore making the left side of equivalence (1) trivially true. Note the similarity with the pairing proof above. Formally,

$$\begin{aligned} & \log_a(x) = c \;\equiv\; a^c = x \\ \equiv \quad & \{ \text{ instantiate } c := \log_a(x) \} \\ & \log_a(x) = \log_a(x) \;\equiv\; a^{\log_a(x)} = x \\ \equiv \quad & \{ \text{ reflexivity} \} \\ & \text{True} \;\equiv\; a^{\log_a(x)} = x \end{aligned}$$

[1] The attentive reader will recognise this property as the categorial definition of the universal arrow associated to a product construction [2], but such a formal setting is unnecessary for our purposes here. Reference [16] provides, however, an introduction to categorial arguments most suitable for didactical practice and research.

Consider now a slightly more difficult result, which students learn (often by heart) as the *product rule* for logarithms:

$$
\begin{array}{ll}
 & c = \log_a(x \times y) \\
\equiv & \quad \{ \text{logarithm definition} \} \\
 & a^c = x \times y \\
\equiv & \quad \{ \text{cancellation (proved above)} \} \\
 & a^c = a^{\log_a(x)} \times a^{\log_a(y)} \\
\equiv & \quad \{ \text{product of exponentials} \} \\
 & a^c = a^{\log_a(x)+\log_a(y)} \\
\equiv & \quad \{ (\Rightarrow) \text{ the exponential function is injective; } (\Leftarrow) \text{ Leibniz rule} \} \\
 & c = \log_a(x) + \log_a(y) \\
\therefore & \quad \{ \text{indirect equality} \} \\
 & \log_a(x \times y) = \log_a(x) + \log_a(y)
\end{array}
$$

The *same* proof structure, i.e.,

$$
\begin{array}{ll}
 & \dots \\
\equiv & \quad \{ \text{logarithm definition} \} \\
 & \dots \\
\equiv & \quad \{ \text{cancellation} \} \\
 & \dots \\
\equiv & \quad \{ \text{property of the dual structure} \} \\
 & \dots \\
\equiv & \quad \{ (\Rightarrow) \text{ the dual function is injective; } (\Leftarrow) \text{ Leibniz rule} \} \\
 & \dots \\
\therefore & \quad \{ \text{indirect equality} \} \\
 & \dots
\end{array}
$$

applies to find out (or to compute the prove of) the power logarithm rule:

$$
\begin{array}{ll}
 & c = \log_a(x^p) \\
\equiv & \quad \{ \text{logarithm definition} \} \\
 & a^c = x^p \\
\equiv & \quad \{ \text{cancellation} \} \\
 & a^c = (a^{\log_a(x)})^p
\end{array}
$$

$\equiv$ { product of exponentials}

$a^c = a^{p \times \log_a(x)}$

$\equiv$ { ($\Rightarrow$) the exponential function is injective; ($\Leftarrow$) Leibniz rule}

$c = p \times \log_a(x)$

$\therefore$ { indirect equality}

$\log_a(x^p) = p \times \log_a(x)$

The reader may check that the same proof structure is still valid for computing the rule for the logarithm of a quotient. Actually, the common pattern underlying the three proofs comes from the adoption in all cases of the same proof strategy: *the introduction of the corresponding property of the dual function*.

Identifying this strategy, and the proof pattern it leads to, enriches students' reasoning skills: as a rule one may attempt to establish properties of a structure (the logarithm, in this case) by resorting to properties of its dual (the exponential). Moreover, in the long term, this process helps students to build and dynamically enrich a personal *classification* of proofs, which is a basic ability to master Mathematics.

At this point the teacher may challenge students with more complex properties: for example the ones involving change of basis,

$$\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$$

Discovering that again a very similar proof structure applies, a conclusion students arrive quite quickly, builds insight on the subject and empowers their mathematical skills. Actually,

$\log_a(x) = y$

$\equiv$ { logarithm definition}

$a^y = x$

$\equiv$ { ($\Rightarrow$) the exponential function is injective; ($\Leftarrow$) Leibniz rule}

$\log_b(a^y) = \log_b(x)$

$\equiv$ { power logarithm rule (proved above)}

$y \times \log_b(a) = \log_b(x)$

$\equiv$ { routine arithmetic}

$y = \dfrac{\log_b(x)}{\log_b(a)}$

$$\therefore \quad \{ \text{ indirect equality} \}$$

$$\log_a(x) = \frac{\log_b(x)}{\log_b(a)}$$

Note, finally, that this calculational approach allows students to be more constructive because the requirements emerge from the calculations themselves.

## Teaching Scenarios

The examples above are part of what we call in the MathIS project a *teaching scenario* [9]. Actually, a main component of this refactoring programme is the development of specific educational material supporting the use of a calculational approach and algorithmic problem solving strategies in the *practice of mathematics*. This material, in the form of example-driven *teaching scenarios* is designed for use with teams of up to 20 volunteer high school students in the context of extra-curricular "Maths' Clubs". The latter are aimed at students between 15 and 17 years old and do not require any extra-curricular prerequisite knowledge.

A scenario is a fully worked out solution to a problem in a domain integrated in school curricular topics, together with a "method sheet" [9]. The latter provides detailed guidelines on the principles embodied in the problem, on how it can be tackled and solved. Although they can be used directly by the student, they are primarily written for the teacher. In general, each scenario is divided into the following sections:

– **Brief description and goals:** This section provides a summary of the scenario, allowing the teacher to determine if it is adequate for the students.
– **Problem statement:** This section states the problem (or problems) discussed in the scenario.
– **Students should know:** This section lists pre-requisites that should be met by the students. The teacher can use it to determine if the scenario is adequate for the students.
– **Resolution:** This section presents a possible solution for the problem in the style advocated here.
– **Notes for the teacher:** In this section the solution presented above is decomposed into its main parts and each part is discussed in detail. To maintain the balance mentioned in the first paragraph, we also recommend how the teacher should present the material, including questions that the teacher should or should not ask and important concepts that should be introduced.
– **Extensions and exercises:** This section can be used for homework or project assignments. All the exercises are accompanied by their solutions.
– **Further reading:** Recommended reading for the teacher and the students. It may include discussions and comparisons between conventional solutions and the one presented in the scenario.

The success of teaching depends on the amount of discovery that is left to students: if the teacher discloses all the information needed to solve a problem, students act only as spectators and become discouraged; if the teacher leaves all the work to the students, they may find the problem too difficult and become discouraged too. It is thus important to find a balance between these two extremes. Self-discovery is also promoted by the sections *Extensions and exercises* and *Further reading*, which are both designed to encourage further work by the students.

Finally, a word on the role of the 'teacher'. Our experience, however limited it is, suggests she/he is more likely to be expected to act as *coacher* than as repository of pre-framed knowledge. The adoption of new educational practices, would not be effective without an assessment of how teachers feel about that and how this interacts with their own images of their profession. Also at this level, further research is certainly needed.

## 5 Conclusions and Future Work

Understood, more and more, as a condition for democratic citizenship in modern Information Societies, mathematical literacy has to be taken as a serious concern for the years to come. From our perspective this entails the need for a systematic (and, given *l'esprit du temps*, courageous) *unveiling* of Mathematics. That is, to make mathematical reasoning explicit at all levels of human argumentation and develop, through adequate teaching strategies, the skills suitable to promote correct reasoning in all sorts of social, cultural or professional contexts.

This paper focused on two main issues in this process: empowering *mathematical argumentation*, by developing adequate teaching strategies, and *proof*, made simpler, easier to produce and more systematic through a new calculation style which has proved successful in reasoning about complex software. The study of *mathematical arguments* is still an issue in Mathematics Education (see, e.g., [1,19]). On the other hand, the rediscovery of the essential role played by *proofs* (and the associated relevance given to formal logic), has been raised, for the last 3 decades, in a very particular context: that of Computing Science. It may be, so we believe, a contribution of Computing Science to reinvigorating mathematical education.

A final word is in order on the above mentioned relationship of Mathematics and Computing. Actually, the latter is probably the paradigm of an area of knowledge from which a popular and effective technology emerged long before a solid, specific, scientific methodology, let alone formal foundations, has been put forward. Often, as our readers may notice, in software industry the whole software production seems to be totally biased to specific technologies, encircling, as a long term effect, the company's culture in quite strict limits. For example, mastering of particular, often ephemeral, technologies appears as a decisive requirement for recruitment policies.

This state of affairs is, however, only the surface of the iceberg. Companies involved in the development of safety-critical or mission-critical software have already recognized that mathematical rigorous reasoning is, not only the key to success in market, but also the warrantee of their own survival. With a long experience in training software engineers and collaborating with software industry, the authors can only claim the need for a double change:

– in the Mathematics *middle school curriculum*, in which the notion of *proof* and the development of argumentation skills are virtually absent;
– in a popular, but pernicious, technology-driven computing education which fails to provide effective training in tackling rigorously the overwhelming complex problems software is supposed to solve.

Future research goes exactly in this direction. In particular, we are currently working on strategies for developing argumentation and calculational proof skills in probabilistic reasoning. As researchers in Computing Science and Education, respectively, the authors see their job as E. W. Dijkstra once put it, *We must give industry not what it wants, but what it needs.* Mathematics should, definitively, be in the package.

## References

1. Alro, H., Skovsmose, O.: Dialogue and Learning in Mathematics Education: Intention, Reflection, Critique. Kluwer Academic Publishers, Dordrecht (2002)
2. Awodey, S.: Category Theory. Oxford Logic Guides. Oxford University Press, Oxford (2006)
3. Backhouse, R.C.: Mathematics and programming. A revolution in the art of effective reasoning. Inaugural Lecture, School of Computer Science and IT, University of Nottingham (2001)
4. Barrody, A.: Problem Solving, Reasoning, and Communicating, k-8: Helping Children Think Mathematically. Macmillan, New York (1993)
5. Bird, R., Moor, O.: The Algebra of Programming. Series in Computer Science. Prentice-Hall International, Englewood Cliffs (1997)
6. Buschman, L.: Communicating in the language of mathematics. Teach. Child. Math. **1**(6), 324–329 (1995)
7. Dijkstra, E.W., Scholten, C.S.: Predicate Calculus and Program Semantics. Springer, New York (1990)
8. Ferreira, J.F.: Principles and applications of algorithmic problem solving. Ph.D. thesis, University of Nottingham (2010)
9. Ferreira, J.F., Mendes, A., Backhouse, R., Barbosa, L.S.: Which mathematics for the information society? In: Gibbons, J., Oliveira, J.N. (eds.) TFM 2009. LNCS, vol. 5846, pp. 39–56. Springer, Heidelberg (2009)
10. Ferreira, J.F., Mendes, A., Cunha, A., Baquero, C., Silva, P., Barbosa, L.S., Oliveira, J.N.: Logic training through algorithmic problem solving. In: Blackburn, P., van Ditmarsch, H., Manzano, M., Soler-Toscano, F. (eds.) TICTTL 2011. LNCS(LNAI), vol. 6680, pp. 62–69. Springer, Heidelberg (2011)
11. Forman, E., Ansell, E.: The multiple voices of a mathematics classroom community. In: Kieran, C., Forman, E., Sfard, A. (eds.) Learning Discourse: Discursive Approaches to Research in Mathematics Education, pp. 115–142. Kluwer Academic Publishers, Dordrecht (2002)

12. Gries, D., Feijen, W.H.J., van Gasteren, A.J.M., Misra, J.: Beauty is our Business. Springer, New York (1990)
13. Gries, D., Schneider, F.: A Logical Approach to Discrete Mathematics. Springer, New York (1993)
14. Halmos, P.R.: I Want to Be a Mathematician. Springer, Berlin (1985)
15. Hiebert, J.: Reflection and communication: cognitive considerations in school mathematics reform. Int. J. Educ. Res. **17**, 439–456 (1992)
16. Lawvere, F.W., Schanuel, S.H.: Conceptual Mathematics. Cambridge University Press, Cambridge (1997)
17. McClain, K., Cobb, P.: The role of imagery and discourse in supporting students' mathematical development. In: Lampert, M., Blunk, M.L. (eds.) Talking Mathematics in School: Studies of Teaching and Learning, pp. 17–55. Cambridge University Press, Cambridge (1998)
18. Mendes, A.: Structured editing of handwritten mathematics. Ph.D. thesis, University of Nottingham (2011)
19. O'Connor, M.C.: Can any fraction be turned into a decimal?: a case study of a mathematical group discussion. In: Kieran, C., Forman, E., Sfard, A. (eds.) Learning Discourse: Discursive Approaches to Research in Mathematics Education, pp. 143–185. Kluwer Academic Publishers, Dordrecht (2002)
20. OCDE Report: Education at a clance: OCDE indicators 2006. OCDE Publishing, Paris (2006)
21. Rittenhouse, P.S.: The teacher's role in mathematical conversation: stepping in and stepping out. In: Lampert, M., Blunk, M.L. (eds.) Talking Mathematics in School: Studies of Teaching and Learning, pp. 163–189. Cambridge University Press, Cambridge (1998)
22. van Gasteren, A.J.M. (ed.): On the Shape of Mathematical Arguments. LNCS, vol. 445. Springer, Heidelberg (1990)
23. Wood, T.: Creating a context for argument in mathematics class. J. Res. Math. Educ. **30**(2), 171–191 (1999)