43

Teaching of Discrete Mathematics

Using Functional Languages in Computer Science Courses

Simone C. Mendes, Tiarajú A. Diverio and Dalcidio M. Claudio
Instituto de Informática e PPGC da UFRGS, C.P. 15064, 91501-970 Porto Alegre, Brazil
diverio@inf.ufrgs.br

Keywords: computing, algorithms, higher education, programming

Abstract: This paper presents an alternative methodology of development for the discipline of Discrete Mathematics offered to courses in the area of computation. It explores the availability of new teaching tools and programming languages. This way, we try to stimulate and provide students with an environment in which they can reach better results in order that it can contribute in the creative and constructive application of their knowledge.

1. INTRODUCTION

The constant development of technology has unleashed new challenges in education. Its use has been seen as a strong element for knowledge construction support. This makes it possible for the teacher to adopt a new role, as a mediator and facilitator of the student's learning.

Some parts of mathematics disciplines have found difficulties in using computational tools in the development of their concepts and the performance rate in these disciplines have not been satisfactory. One such case is Discrete Mathematics, which is essential to the formation of the courses in the area of Computer Science, as seen in universities of Rio Grande do Sul State (Menezes et al. 2000).

D. Watson et al. (eds.), *Networking the Learner*
© Springer Science+Business Media New York 2002

An alternative methodology is desired for the development of the Discrete Mathematics discipline, offered to courses in the area of computation, by exploring the availability of new teaching tools and programming languages. Thus, we try to stimulate and provide students with an environment in which they can achieve better results and contribute to a creative and constructive application of their knowledge.

2. MATHEMATICS AND COMPUTATION

The development of computation creates a propitious environment for the launching of new educational technologies. Such tools end up giving value to the process of the student's learning and renew the didactic methods used by the teacher.

The integration of technology with education has taken place in different periods in the most distinctive areas. The Mathematics teaching, specially the Discrete Mathematics for graduation courses of the computational area in Brazil, has not found yet a way to incorporate computational tools to the development of very important concepts for the formation of the student in this area. It is very common for students to study the disciplines in a forced way, as an obligation they must fulfill. They do not see direct or indirect application of the contents in their future professional life, and even less they do not see it as a necessary and essential knowledge for the rest of the course.

It is very common for them to study the disciplines of the mathematical area as if they were independent of the rest of the course, without having the opportunity to experience learning in which the intrinsic relation between mathematics and programming is explained.

Mathematics is very used in several areas of Computing Science, in applications that include the specification of programs, data structures, analysis and implementation of algorithms and correctness. The use of computers, however, can help making mathematics easier to be understood and used, by turning mathematical terms executable and abstract concepts more concrete, all through the use of computational tools such as mathematical proof verifier.

Mathematics provides fundamental tools to any scientific discipline, which is essential to programming because it helps the student to rationalise the implementation of the programs. And the programming, on the other hand, has been utilised since its beginning for the resolution of mathematical problems. However, nowadays the courses on mathematics are delivered in a separate way from the programming courses, and they make very little use of the second one to re-enforce and stimulate the learning of mathematical concepts.

The solutions for the mathematical problems can be drawn and implemented in the computer, as if it were the pencil and the paper traditionally used for this very purpose. This new way to reach the solution modifies the procedure as a whole, but without depriving it of its characteristics. The traditional process, according to Rosa and Cirigliano (1999), is one that creates a mathematical model, starting from the problem, and then creating a transformed model that will conduct it to the solution, upon which the proof will be carried out in order that the solved problem can be finally obtained. When integrating the programming to this model, we have a new process in which the mathematical model is implemented by reaching the solution through the computer. To assure the validity of this solution we must then demonstrate the equivalence between the models, which guarantees that the solution obtained by the program is correct. This new model, through which it is, illustrated the process of resolution of problems under the algorithm approach of the mathematical concepts and the approach on the functional characteristics of a programming language is represented in Figure 1.

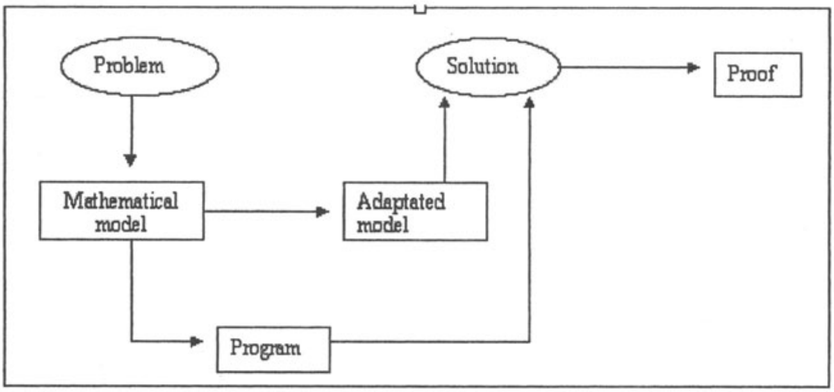

Figure 1. Process of resolution of a problem through the implementering in a programming language

It is important for the student to acquire, from the beginning of the course, a habit of specifying, solving problems and try to solve them by using the computer and a mathematical method to reason about them. Students can have a vision of the integrated process represented in Figure 1 through the methodology of the application of a language as formalism, in which the solutions for certain problems are programmed.

3. STATE OF ART OF THE DISCRETE MATHEMATICS DISCIPLINE

The discipline of Discrete Mathematics is present in the courses of the Computer Science area according to the Policy Laws and Bases of the Education and Culture Ministry of Brazil (MEC 2000). According to this document, the discipline is a fundamental pre-requisite for several other disciplines that follow in the posterior semesters, such as programming, computation and algorithms, computer architecture, compilers, database and artificial intelligence. Such evidence re-enforces the fact that their contents must be worked out in a general and ample way, in order to develop the pertinent reasoning, so that students can have a strong base for the moment they start studying the other disciplines of the course.

Although the discipline is essential for the rest of the course, nowadays it is not obtaining the desired success. The approval rates verified in the two principal universities of Rio Grande do Sul, during the between 1998 and 2000, are considered unsatisfactory due to its importance. The data below demonstrate the performance of the students that attended this discipline in the mentioned universities.

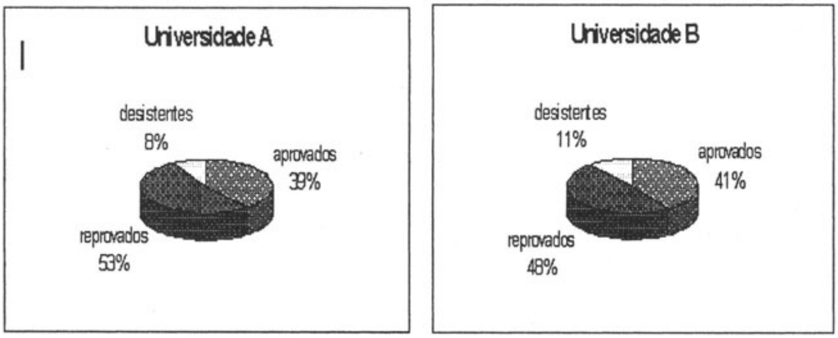

Figure 2. Performance of students in discrete Mathematics, period 1998-2000

This discipline is delivered today in a traditional manner, through theoretical-expositive classes and through realisation of works and written exams. We can notice, thus, the necessity to search in the programming paradigms an alternative, through the utilisation of a functional language.

4. UTILISATION OF A FUNCTIONAL LANGUAGE

The utilisation of a functional language as a supportive tool to the development of discrete mathematical contents makes it possible to naturally express the mathematical language, to easily prove the properties about the programs and to prove the equivalence between the model and the implementation.

The functional language holds reasoning similar to the one necessary for the development of discrete mathematical contents. That is, it is very similar to the one used in the mathematical formalisms. This fact allows the utilised language to be introduced in the course in a natural way, without requiring extra efforts from the students to learn other programming syntax.

For example, let be the theorem $P \wedge Q \rightarrow P$. its proof can be carried out in a traditional manner by using a functional language. The traditional manner to prove it is through the utilisation of the inference rules of the propositional logic, by structuring it in a tree form, as it can be observed in Figure 3.

1. Rule 1: $\wedge EL$: elimination of "e" ($\wedge$) to the left
1. Rule 2: $\rightarrow I$: Introduction of the implication ($\rightarrow$)
    1. $\underline{P \wedge Q}$ $\{\wedge E_L\}$
    1. $\underline{\quad P \quad}$ $\{\rightarrow I\}$
    1. $P \wedge Q \rightarrow P$

Figure 3. Theorem proof through the traditional manner

The proof carried out by using the functional programming language (for example, the Haskell language), allows the student to apply the same necessary rules for the proof demonstrated in the previous figure at the same time it works on the functional programming concepts. This way, it accomplishes the integration between the concepts developed in classroom and the programming. The proof of the same Theorem using a functional language for this purpose is represented in Figure 4.

```
prova Regra :: Prova
provaRegra =
ImpI
        (AndEL
              (Assume (And P Q))
               P),
        (Imp (And P Q) P)
```

Figure 4. Theorem proof by using the alternative methodology

Based on these facts, and through a master's thesis, an alternative methodology for the teaching of Discrete Mathematics that aims at utilising a functional language in the development of the discipline's concepts has been developed and proposed. The work is based on a prototype in which libraries using the Haskell language were worked out, which implement some theorem proofs, mathematical recursion and induction, having as a model the proposal of Hall and O'Donnell (2000). The work set results in the proposition of a new methodology that culminates in the restructuring of the discipline as a whole and its reflects on the discipline itself and the other disciplines of the graduation course.

Through the utilisation of theorem prover, for example, the student develops several parallel skills. Besides needing knowledge of the propositional logic rules, students must know how to apply them in order to carry out the correct implementation of the proof, while they develop programming skills in a language very similar to the one used in the proof. Figure 5 illustrates some of the skills developed simultaneously when the programming is utilised as a tool in the assistance to the learning of mathematical concepts.

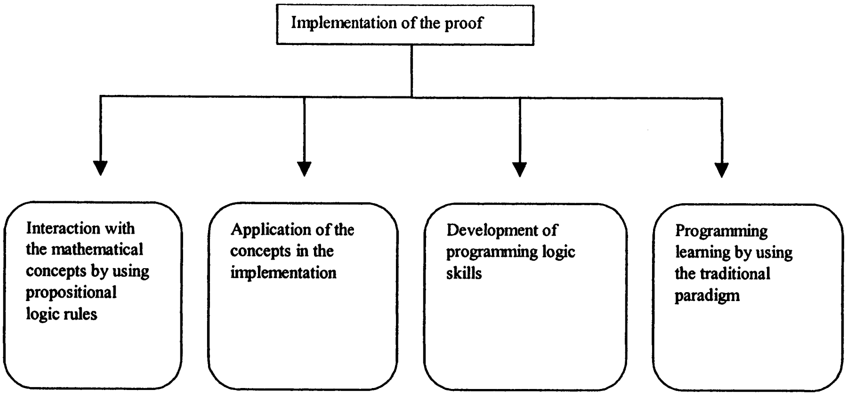

Figure 5. Skills developed during the implementering of proof

When programming and the model of resolution of a problem are integrated, instead of just one process being used, no matter the theorem proof, several parallel processes are developed, important and necessary skills to the students of the computational courses. Without the integration, a basic process takes place, which is the interaction of the student with the mathematical concepts, as demonstrated in the examples of Figures 3 and 4, through the utilisation of the propositional logic rules in order to carry out the traditional proof of the theorem.

The integration of the programming the proof of the Theorem develops three new skills, which added to the one described in the traditional process, are the following:

Interaction with the concepts of the discipline: the student performs learning of fundamental concepts of the discipline.

Application of logic concepts in the implementation: to implement the solution of the proof, the student must have the knowledge of the concept (1) and, then, apply them in the implementation of the proof;

Development of programming logic skills: at the same time it is thought about a solution to carry out an implementation of the proof, it performs logical reasoning activities necessary to the programming;

Learning of the programming by using the functional paradigm: due to the fact that the syntax of the functional language is similar to the mathematical language, the student learns to program by using the functional paradigm without too many difficulties.

The skills created when using the alternative methodology are illustrated in Figure 5.

5. FUNCTIONAL LANGUAGE AS FIRST LANGUAGE

The first language taught in the courses of the Computer Science area in the main universities of Rio Grande do Sul that is a procedimental language. Its characteristics do not allow logic concepts, theory of the sets and demonstrations to be worked out with the same facility that a functional language, for example, does.

The functional languages, like the Haskell language, have as a principal characteristic the reproduction of a mathematical language because in this paradigm the programs are formed exclusively by functions, which justify its denomination. The principal program is a function that receives entry parameters and produces a result, in the same way the student reasons mathematically when he is solving a problem on the paper.

The teaching of a functional language in the first semester of the course helps the learning process of the concepts of discrete mathematics, mainly because these two languages have the same syntax used in Mathematics.

This way, we could have the teaching situation represented by Figure 6. The figure illustrates the skills developed in each one of the methodologies, the utilised languages as first language of the course and de performance obtained and/or expected from the students.

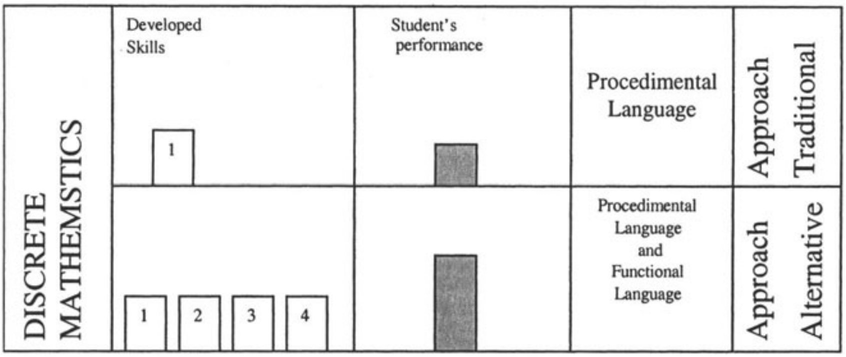

Figure 6. Utilisation of the alternative methodology: functional language also as first language

6. FINAL COMMENTS

The utilisation of a functional programming language in the discipline of Discrete Mathematics, offered in the first semester, is an alternative methodology of teaching. This methodology allows the student to apply and construct in a creative way, interaction skills with the concepts of the discipline, their application in the implementation, development of programming logic skills and learning of the programming with focus on the functional paradigm, as demonstrated in Figure 6.

The mathematical union with the functional programming paradigm also makes possible for students to develop concepts of discrete mathematics, which are very important for their development, without them noticing it. At the same time, they are developing functional programming concepts, with are only worked in more advanced semesters of the course.

The utilisation of the functional programming as a tool for the constructive and applied development of the theoretical concepts, does not require extra efforts for the comprehension of its programming syntax, as its programs are formed exclusively by functions. Thus, the necessary reasoning for both activities are similar, which provides a friendly integration of the discrete mathematics with the functional programming.

Finally, all the educational technology available to provide students with a learning environment that keeps them away from the hard classes, and apparently with no application for their professional future, must be used. It is necessary, then to allow students to make use of all their potential and enthusiasm in the activities that have as a principal objective the students' learning.

REFERENCES

MEC (2000) *Diretrizes curriculares das áreas de computação e informática*. Brasília. [www.mec.gov.br/Sesu/diretriz.shtm] 12.06.2000.

Rosa, S. and Cirigliano, G. (1999) Ensayo sobre Matemática aplicada a Computación. In *VII Congresso Iberoamericano de Educación Superior en Computación*, 1999, Asuncíon, Paraguay, Anais, Asunción, p. 141.

Hall, C. and O'donnell, J. (2000) *Discrete mathematics using a computer*. 1ª ed., Springer.

Menezes, P.B. et al. (2000) Proposta de plano pedagógico para matéria Matemática. In *Congresso da Sociedade Brasileira de Computação*, Curitiba, PR.

BIOGRAPHY

Simone C. Mendes has a Masters in Science Computer and Mathematics at PPGC-UFRGS. Her areas of interests are Informatics in Education and Computational Mathematics.

Tiarajú A. Diverio graduated in Mathematics for the UFRGS, and has Masters and Doctorate in Computer Science to the CPGCC (1986 and 1995). He is Professor and Researcher of the Institute Computer Science at UFRGS. His areas of interest are Processing of High Performance (Processing Vectorial, Parallel and Distributed and Parallel Mathematical Algorithms) and Informatics in Education (Educational Development of Software for Mathematics).

Dalcidio M. Claudio has a Doctorate in Computer Science from Karlsruhe. He is a researcher of the Institute Computer Science at UFRGS and Professor at CPGCC in PUCRS. His areas of interest are Interval Mathematics, Scientific Computation and Informatics in Education (Educational Development of Software for Mathematics).