Mathematics and programming problem solving

Anabela Gomes
Polytechnic Institute of Coimbra
Coimbra Superior Institute of Engineering
Dep. Informatics and Systems Engineering
3030-199 Coimbra - Portugal
+351 239 790 350
anabela@isec.pt

Lilian Carmo
University of Coimbra
Dep. Informatics Engineering
3030-290 Coimbra - Portugal
+351 239 790000
lilian@dei.uc.pt

Emília Bigotte
Polytechnic Institute of Coimbra
Coimbra Superior Institute of Engineering
Dep. Informatics and Systems Engineering
3030-199 Coimbra - Portugal
+351 239 790 350
ebigotte@isec.pt

António Mendes
University of Coimbra
Dep. Informatics Engineering
3030-290 Coimbra - Portugal
+351 239 790000
toze@dei.uc.pt

ABSTRACT

Programming is hard to learn to most novice students. Several reasons can be pointed out to this situation, going from the inherent difficulties of the subject to the lack of mathematical problem solving competences that students should have acquired before. This later factor is, in our view, extremely important and explains many student difficulties.

In this paper we present some experiences carried out during the second semester of 2005/2006, where we tried to explore the relationships between mathematical problem solving competences and the lack of programming abilities shown by a group of students who failed their initial programming course.

Keywords

Mathematical skills, programming competences, problem solving

1. INTRODUCTION

Programming learning is well-known as a difficult task to many students. Several authors have discussed possible causes for this problem [1], [2] but we think that a main factor is the lack of problem solving abilities that many students show, namely those that involve mathematical and logical knowledge. To verify this assumption we made some experiences, trying to determine how the development of mathematical and logical problem solving abilities would impact programming capacity. In a first phase we decided to evaluate mathematics and logic knowledge of a group of students who failed to get approved in the first programming course (first semester 2005/2006). Then, during the second semester, these students followed a special course on mathematics and logic problem solving, in order to improve their skills in this field and see if this improves their basic programming competences. Although we still don't have final results, during the semester it was possible to identify many students' main difficulties and relate them with their programming limitations. In this paper we describe our experience and focus on those preliminary conclusions.

2. STANDARDS

In this experience we followed a set of standards prepared by a workgroup of the Commission on Standards for School Mathematics from National Council of Teachers of Mathematics [3]. This document defines the standards that indicate the abilities that mathematics teaching should promote at different levels of education. These levels are divided in three groups, primary education, 5th to 8th grades and 9th to12th grades. After analysing the exercises proposed and the abilities defined for each level, we decided to concentrate our experience in some standards concerning the level between the 5th and the 8th grades. This decision was made based on our experience that this type of problems usually is used in introductory programming courses and also because many of our students show difficulties to solve them.

In this section we will describe the mathematical standards that in our opinion may have a stronger influence in programming abilities and that were used in our experience.

2.1 Mathematics as problem solving

In this experience our interest was to verify if the students have the necessary mathematical concepts to solve some types of problems. We wanted to know if they can apply strategies and mathematical concepts to solve a wide variety of everyday problems, with special interest in those with a programming solution. In addition, we wanted to find out if they can generalize solutions and strategies in order to use them in new situations. During the experiences this standard was object of intensive training, hoping that students would acquire a certain mental experience, using problem-solving approaches to develop reasoning and training a variety of abilities, such as exclusion of parts, logic of the proposals, among others.

2.2 Mathematics as means of communication

This standard indicates that the Mathematics curriculum should allow students to exercise communication skills, because in that way, they hopefully will acquire the capacity to interpret and to

evaluate mathematical ideas. It is very important to give students opportunities to reflect on and clarify their thinking about mathematical ideas. One of the obstacles they frequently cope with in solving a programming problem lies in transforming a textual solution into mathematical language. As the experiences went on this standard was applied in an intensive way, because we consider that the capacity to read, write, listen, discuss and interpret mathematics is a vital part of programming learning.

2.3 Mathematics as means of reasoning

In successive sessions we presented to students different types of logical problems which would train their abilities in inductive and deductive reasoning. To help them we used a large amount of models, known facts, properties and relationships. The presented exercises didn't always have a direct translation in programming terms, but they included fundamental aspects and concepts to develop programming skills. Another aspect of this standard, considered to be a powerful strategy in problem solving, consisted on the identification of patterns and relationships to analyze mathematical solutions and certain types of regularities which is the basic characteristic of inductive reasoning. The pupils were encouraged to validate their assumptions through the construction of arguments that justify them, as well as carrying through generalizations or applications to new problems.

2.4 Mathematical connections

This standard was extensively explored during the experiences. We made a constant effort to make students establish connections between concrete situations in their daily lives and mathematics as well as programming. We consider it is very important that students can recognize relationships among different topics in various domains, especially in mathematics and programming. As the sessions went on, we tried to point out analogies demonstrating the relations between certain concepts and to link conceptual and procedural knowledge.

2.5 Numbers and relations between numbers

Although there are different but equivalent numerical representations to characterize the same amount (for example, real numbers, fractions, decimals, percentages, powers, scientific notation, and so on), this standard intends to demonstrate that it is very important to develop a deeper understanding of various numerical representations and to recognize that in some situations there are representations that are more suitable than others. We think that this standard is not very important if the objective is to improve programming skills. However, we included it in some exercises to show that there are different forms to represent the same thing, but that in some situations some of them make more sense than others.

2.6 Numerical systems and number theory

The number theory offers many possibilities for interesting explorations that may have positive results to student's problem solving capacity and to the understanding and development of other mathematical concepts. Thus, many of the exercises we used in our experience had to do with number theory, with frequent applications in programming, such as abundant numbers, deficient numbers, perfect numbers, triangular numbers, squares, cubes, palindromes, factorials, Fibonacci numbers, maximum and minimum common divider of two numbers, among others.

2.7 Calculation and estimation

Our main interest in this area was that the students could use estimation techniques in calculations and ratios, not especially to solve problems, but mainly to evaluate results. The evaluation and analysis of wrong results can constitute meaningful learning opportunities in programming. So it is important that students can be aware of them as soon as possible. While students try to describe what they are doing and verify their answers, they are simultaneously using conjectures to formulate new problems and planning new strategies to answer them.

2.8 Patterns and functions

An important aspect of this standard deals with the capacity to represent a real problem expressing its solution using a function. This is very important to develop abstraction skills and programming competences. Two types of important programming problems can be connected with these competences. The first deals with the concept of variable. The second concerns the capacity to observe simple situations and recognize that they can lead to a general method which can be applied to a variety of situations.

2.9 Algebra

One aspect of this standard with implications in programming is the limited understanding that many students have about variables, expressions and equations. Sometimes, in programming learning the variable concept is the first problem students have to face. For example, many students associate a numerical value to a letter (variable name) since the beginning, some ignore the letter, and others still deal with the letter as if it was a name of an object (for example, y means youngster instead of number of youngsters [4]). Another aspect regards to the definition of expressions and equations used in the construction of cycles. During the sessions this standard was intensely used in the exercises presented to the students because we considered essential to dominate these concepts to algorithm understanding and development.

2.10 Geometry

We believe that the capacities involved in this standard are important to the development of programming capacities. Directly, because there are numerous programming problems that imply that the students know how to identify, describe, compare, and classify geometric figures. Also there are other problems that can only be solved through the application of geometric models. Indirectly, for the inherent abstraction capacities that geometry helps to develop. That's why several geometric problems were presented to the students during our research.

"Statistics", "Probabilities" and "Measure" which also belong to the standards have not been described here, because they were not used during the experience. Although the capacities they develop are important, we considered that the described standards were enough to our experience goals.

3. EXPERIENCE

Our experience was conducted during the second semester of the 2005/2006 academic year. It included several sessions, alternating sessions where the students were asked to solve mathematical exercises, logical challenges and problem solving in diverse domains, focusing especially in those with a closer relation with programming. In the next session Mathematics and Computer Science teachers corrected and explained previous session problems, as well as introduced basic mathematical concepts when necessary. This experience involved a group of 33 Informatics students from the Superior Institute of Engineering of the Polytechnic Institute of Coimbra who volunteered to participate. The involved students had failed the course of Algorithms and Programming in the first semester of 2005/2006 academic year. They all presented severe difficulties in basic programming.

The set of selected exercises was chosen taking in consideration the results of two diagnosis tests made in the experience first sessions. One of them focused on programming competences while the other looked for students' mathematical concepts, especially those more connected with programming. As expected, these tests generally showed very low programming skills, but also many students did not show basic mathematical skills that should be expected when entering higher education.

4. RESULTS ANALYSIS

The experience available partial results are presented in this section. When possible we give examples of problems that were used to test and develop student's problem solving skills. The results show that the students have many limitations, namely:

- Students do not have enough basic mathematical concepts concerning the number theory - We think that students aren't able to solve many of the programming exercises, because they do not know basic mathematical concepts. Consequently there is a group of common programming exercises (for example, prime numbers, dividing and multiple numbers) that they can't solve. To confirm this assumption and before going into programming exercises that included that type of concepts (for instance a number divider), they were asked the following question "Given the following numbers 1, 2, 3, 4, 8, 11, 12, 20, 32, 44, 66, 70, 88 indicate the dividers of 22". We verified that many students confused the concepts of divider and multiple of a number. A reduced number did not answer at all, and when asked why sometime later, they said that they did not have any idea whatsoever about this concept.
- Students have difficulties to transform a textual problem into a mathematical formula that solves it - In some exercises students had to do calculations. If the involved quantities were easily calculated (for example by "counting with the fingers" or through a simple mental operation) most of them obtained a correct answer. The following problem presented in one session can be mentioned in this context. "A man is on a step in the stairs. He goes up 5 steps, then he goes down 7, then he climbs again 4 steps and later another 9 to arrive on the last one. How many steps are there altogether?" We verified that most students, who got a correct answer, came up on it through literal or graphical descriptions. The exception was a reduced number of students who presented the solution using a function. However, when sometime later they were asked to solve the same problem using a formula, the majority was unable to reach the right answer. Thus, the difficulties in some programming problems can be associated with the difficulty to obtain a formula to solve the problem. That is to transform the problem mathematically or to go from a concrete to a more abstract and generic level.
- Students don't recognize geometric figures - It was verified that many students don't recognize geometric figures or at least they do not have a clear comprehension about their definitions and associated concepts. This difficults the resolution of many programming problems based on that knowledge. In the initial phase of the experiences we proposed the following problem: "Make a program that given 2 values that represent the length of two adjacent sides of a 4 side polygon and the value in degrees of the angle formed by them, indicate the type of polygon. It is known that the polygon has two pairs of equal sides and also two pairs of equal angles". The majority of students were unable to solve it. In a later session, the same exercise was proposed to them, but this time we previously presented them the necessary geometric concepts. So we had explained them what a polygon, a parallelogram, a square, a rectangle and a rhomb are. This time many more students were able to solve this problem;
- Students have difficulties to understand the problem description - We think that many times students fail to solve a problem simply because they don't understand it clearly. Although the exercises had been written in a clear and simple way, students frequently showed doubts in their interpretations. During the sessions, they kept wondering what they had been asked to do. From time to time, the teachers asked students to read the problem calmly and aloud. We confirmed that they did not know how to read correctly, therefore they made pauses or omitted them in incorrect places, altering completely the meaning of the text. But when the teacher read it aloud, students said that they had understood it, without having been given any additional explanation. Other times pupils claimed that they did not know where to initiate the resolution process, because they didn't understand clearly what they were asked to do. However, when we presented them an equivalent problem but divided in successive stages, many of them could solve it. For example, in the specific case of programming problems these stages included the main solution components, such as input data, expected results, necessary auxiliary variables and initialization values (if required). In this case, when we asked students for a textual description or a graphical illustration of the general process, the majority could, at least, understand what was asked for. So we think that a subdivision of the resolution in different steps could help students to understand the problem in the initial

learning stages. However we verified that when the problems were presented in this way, the students didn't always realize that each stage is a component of the whole solution and many can't see the relation between the different steps.

- Students are unable to define comparison criteria - The following question was asked to the students: "Assume that there is a set of 4 square shaped boxes of different dimensions. Indicate the procedure to place them inside each other". All the students were able to determine a procedure to solve this problem, but the majority referred to the boxes as the smaller box and the bigger box. They didn't explain how they had come to the conclusion that a box was bigger or smaller than the other. When asked how they would know if a box was bigger or smaller than the others, they answered that they would look at the boxes. However they were not able to define a general criterion to reach to this conclusion. Later, in another session, we asked them the same question, but in the statement each box was indicated to have the sides L1, L2, L3 and L4 and the following relation between them L1<L2<L3<L4. In this situation they were able to solve the problem correctly.

- Students were unable to guide themselves in the Cartesian Plan, have lack of trigonometric concepts or are unable to apply them to solve exercises - We think that many times the students do not know how to solve programming problems, because they can't identify the geometric models that are necessary to solve them. In an initial session we asked students to solve the following problem: "Describe the procedure to draw the figure shown below (Figure 1). The centre of that figure is in the point (100, 100) and the circumferences ray is 30". In this session, not a single student was able to solve it, not even to propose a minimum sketch towards the solution. However, in a later session the question was reformulated, and had some extra questions inserted in the same group. The goal was to guide the students to the correct solution by dividing the problem in successive stages. It was clear that many students didn't understand some basic concepts of the Cartesian Plan, were unable to locate the circumferences in some coordinates and had no knowledge of some basic trigonometry concepts, such as sine and co-sine. However, many others were familiar with these concepts, but were not able to apply them to solve the problem. The same question was repeated in the final session on a programming test and this time the code that implemented the figure was given to the students. Two questions were made; one asking them to indicate the value assumed for each variable in each cycle iteration and another asking them to indicate the final result of the program. Surprisingly, there were only a few students that were able to answer correctly.

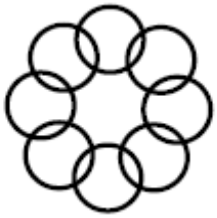

Figure 1 – Figure given to the students

- Students have difficulty in calculus - We think that some students don't know how to program because the types of exercises they work with imply calculation. It is necessary that they develop procedures to determinate certain amounts, in order to solve problems. The following is an example that confirms this difficulty. "The note delivery in an ATM (Automatic Teller Machine) can be programmed in different ways. One of them is giving the user the least possible number of notes. Describe the process to determine this number. Assume that the machine uses notes of 5, 10, 20 and 50 euros and that the desired amount is a multiple of 5 euros". We confirmed that the students who succeed to solve the problem had done it by assuming concrete amounts. They were unable to derive expressions for the calculation from these amounts, being given both a general and an abstract value.

- Students have weak abstraction levels – We believe that some problems pointed out before in this text can also be related to the enormous difficulties students feel to reach higher abstraction levels. In the experiences carried through, we verified that when the students were put against similar problems they had different results depending whether the involved amounts were concrete or abstract. An example stating that problem: "Which even number is bigger than 20, minor than 30 and the sum of its digits is 8". As we supposed all students answered correctly with no difficulty. However, in a later session they needed to answered the following question: "Develop an algorithm that finds out the even number that meets the following criteria: it is bigger than a certain number named *small* and smaller than another number named *big* and the sum of its digits is sum." This reformulation could raise other problems related with programming, such as the process to determine if a number is even or to calculate the sum of the digits of a number. However, the key issue that the students struggled with had to do with the difficulty to understand what they were asked to do. They had difficulties to identify the necessary variables, they couldn't see what to do and they couldn't see that the problem was similar to the previous one.

- Students have lack of logical reasoning – During the experiences, we launched some logical challenges and charades that involved diverse concepts, such as exclusion of parts, logic of proposals, and discovery of ambushes or wrong reasoning, among others, with the sole purpose of training their logical reasoning. We confirmed that initially the students had big difficulties and the majority of them were unable to solve that kind

of problems. However, we observed that this difficulty was significantly reduced as they kept on solving that type of problems along the sessions. We considered that to be able to program the students should both apply the inductive and deductive reasoning, and validate their own thinking which should be trained in an intensive manner.

5. CONCLUSION AND FUTURE WORK

The experiences carried through during the second semester of 2005/2006 in the Superior Institute of Engineering of the Polytechnic of Coimbra were described in this paper. Throughout the research the students belonging to the studied sample were submitted to a series of problem solving exercises that were directly or indirectly related to common beginners programming problems. The essential intention was to identify the students main gaps in mathematics and to analyse their influence on programming capacity. The analyses of the obtained results indicate that the majority of the students don't have basic mathematical concepts, which reflects in their problem solving ability and, therefore, in their low programming skills. With our observations we could establish clearly that the lack of programming skills was accompanied by a deep lack of mathematical knowledge and skills and that the later many times was the main cause of the former. The capacity to program consists of a hierarchy of multiple competences [4]. However we believe that many of them can be trained and improved, with persistence, determination and perseverance.

Our research also includes a comparative study between the results the students obtained in the programming and mathematical tests done in the beginning of the semester with similar tests to be done after the semester ends. The idea was to evaluate the impact the problem solving activities had on the students' abilities. Also, next academic year, when the students follow the initial programming course again, it will be possible to study eventual improvements in these students programming performance. Those will be the next steps of our work.

6. ACKNOWLEDGMENTS

The authors would like to thank all students that participated in the experience and especially to professor Francisco Pereira.

7. REFERENCES

[1] Gomes, A. and Mendes, A. J. A animação na aprendizagem de conceitos básicos de programação, *Revista de Enseñanza y Tecnología*, 13 (1999), 22-32.

[2] Jenkins, T, On the difficulty of learning to program, In Proceedings of the 3rd Annual LTSN- IC Conference (2002), Loughborough University.

[3] Normas para a avaliação em matemática escolar. *Associação de Professores de Matemática*. (Out. de 1999).

[4] Lister, R., Adams, E., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J., Sanders, K., Seppälä, O., Simon, B., Thomas, L. A Multi-National Study of Reading and Tracing Skills in Novice Programmers, *SIGSCE Bulletin*, 36, 4 (Dec. 2004), 119-150.