Searching for Barriers to Learning Iteration and Runtime in Computer Science

Philipp Shah
Technische Universität München
TUM School of Education
80333 München, Germany
+49 89 289 29114
Philipp.Shah@tum.de

Dino Capovilla
Technische Universität München
TUM School of Education
80333 München, Germany
+49 89 289 29114
Dino.Capovilla@tum.de

Peter Hubwieser
Technische Universität München
TUM School of Education
80333 München, Germany
+49 89 289 29110
Peter.Hubwieser@tum.de

ABSTRACT

The knowledge about misconceptions of programming beginners can help the instructors to improve their lessons and exercises and to eliminate barriers to learning. However, there is not much research about learning barriers, like misconceptions, in computer science education. This paper explains the goals and first results of our survey in this area. We interviewed 60 students in a pretest and 110 students in a test [8] to observe whether misconceptions about iterations and runtime are following underlying intuitive rules. Our results are verifying an underlying rule and unveiling two new misconceptions, which – to the best of the authors' knowledge - have not been mentioned in literature yet. The results could help teachers to prevent learners' misconceptions.

Categories and Subject Descriptors

• **Social and professional topics** → Professional topics → Computing education;

Keywords

Misconceptions; Intuitive Rule; Alternative Conceptions; Iterations; Runtime.

1. INTRODUCTION

Learners of computer science sometimes make a 'mistakes'. A few of these mistakes are possibly just problems with syntax or are due to inattention, but some of them have their source in so-called misconceptions or alternative conceptions (see section 2 for a definition). We are interested in a systematical view on the source of misconceptions in computer science. Categorizing misconceptions based on their underlying structure could help to explain mistakes and prevent learners from making mistakes, which are based on these alternative conceptions. Therefore, we constructed a pretest and a written interview [8] with six questions. With the first two questions we tried to validate the misconception 'two programs containing the same statements (even if in different order) are equally efficient' discovered by Gal-Ezer & Zur [3] by testing students with very simple Java programs – instead of rather complex C programs. Furthermore, we tried to generalize this misconception to the misconception 'two programs containing the same number of lines and similar (but not same) statements are equally efficient'. This would suggest a more generalized and stronger effect of the intuitive rule 'same A, same B' (see section 2) for students of computer science. This intuitive rule would emerge as a barrier to learning and a base for alternative conceptions. This paper presents the qualitative evaluation of the results of the first question, which is validating the misconception mentioned above. Moreover, we are unveiling two new misconceptions, which have not been mentioned in literature yet. The results could help instructors to prevent learners' misconceptions.

2. THEORETICAL BACKGROUND

Alternative conceptions or so-called misconceptions are conceptions which are not in line with accepted scientific notions [9], but some of them seem to follow intuitive rules. Stavy and Tirosh [9] investigated such intuitive rules and their corresponding misconceptions in science and mathematics.

Publications about misconceptions in computer science education can be considered as collections of these alternative conceptions; Ragonis and Ben-Ari [7] who listed about 58 misconceptions and difficulties, are providing a good example. Some of the first authors who are collecting misconceptions using an object oriented programming language (Smalltalk) are Holland, Griffiths and Woodman [4]. Referencing to them [4], Fleury [2] is searching for misconceptions using Java as a programming language. Ragonis and Ben-Ari [7] are investigating misconceptions based among others on Holland, Griffiths and Woodman [4] and Fleury [2]. Danielsiek, Paul and Vahrenhold [1] – referencing to Holland, Griffiths, Woodman [4], Ragonis and Ben-Ari [7] – are analyzing misconceptions on complex data structures like heaps and binary search trees.

Our major interest is to unveil underlying intuitive rules of misconceptions. Gal-Ezer and Zur [3] succeeded in connecting intuitive rules of science and mathematics to intuitive rules of computer science which lead to misconceptions in both domains. They discovered among others the basic alternative conceptions 'a shorter program (in terms of code lines) is more efficient' following the intuitive rule 'more of A, more of B' [9] and 'two programs containing the same statements (even if in different order) are equally efficient' following the intuitive rule 'same A, same B' [9]. However, they tested high school students using rather complex C programs. These relatively complex programs could cause students to fail the task of choosing the most efficient program out of two given programs with the same lines of code based on other unknown reasons.

3. SURVEY

As a pretest we interviewed about 60 students who did not pass the exam to the first course (CS 1) of computer science at our CS department. This course is an introduction to programming (using Java as a programming language). We did this pretest to find out if our Java programs were easy enough to be understood by the students and to validate if the misconception 'two programs containing the same statements (even if in different order) are equally efficient' occurred. After positive results (section 5.1) of this pretest we interviewed about 110 students from a secondary school (10th grade; age range: 15-17). These students visited four different school classes and had been taught 40 hours object-oriented programming (with Java) and modelling at this school year and about 150 hours computer science (functional modelling, database systems, etc.) from grade 6 till grade 10 in total. They did not explicitly learned about runtime analysis.

The students had to answer two questions about runtime. For each question they had to decide which of two Java methods runs less statements or whether they both run the same number of statements. Students had to give written explanations for both possibilities. In this paper we will discuss the results of the first question of our interview.

The first question contained two methods with exactly the same lines, but in different order, which causes a different runtime.

```
void A( ){
   int n = 100;
   output("This program prints n lines.");
   for ( int i = 0; i < n; i = i + 1 ){
      output ("Line No." + i);
   }
   n = n + 500;
}
void B( ){
   int n = 100;
   n = n + 500;
   output("This program prints n lines.");
   for ( int i = 0; i < n; i = i + 1 ){
      output ("Line No." + i);
   }
}
```

output(..) is a shortcut for System.out.println(..) and was explained to the students.

We expected few students from university to fail the question and to argue with conclusions which suggest the existence of the intuitive rule 'Same A, Same B', in this case 'same lines (but) in different order, same runtime'. But we expected that much more students from school would argue with this intuitive rule.

4. METHODS

In this section we will briefly describe our methods.

All questions of the pretest and the interview had to be answered by the students in written form [8]. Both tests were qualitative tests; therefore students had to give (written) explanations for every decision.

The qualitative analyzing of the students' answers was done using the technique of inductive coding (paraphrasing and generalizing) by Mayring [6]. We used two independent coders to ensure good inter-coder reliability [5] after the first 20% of all answers. Differences in generalizing or categorizing were discussed afterwards and where appropriately corrected.

5. RESULTS

5.1 Pretest results

In this section we present the results of the evaluation of the first question of our pretest and test.

The pretest was quite successful; most students understood the Java programs. Just 7 students out of 60 (nearly 12%) answered that both methods (method A and method B) are executing the same number of computational steps. One student answered: "*Same number of steps. Just the order of the statements is different.*" This corresponds exactly to the alternative conception 'two programs containing the same statements (even if in different order) are equally efficient' [3].

5.2 Interview results

After our pretest we were sure that our Java programs were not too complex to be understood and we expected students from secondary school to fail the questions and to argue with conclusions which suggest the existence of the intuitive rule 'Same A, Same B'.

5.2.1 No answer

Indeed just 9 out of 110 (less than 10%) students did not answer the first question, so we conclude that the Java programs were not too complex.

5.2.2 Method A has a shorter runtime (correct answer)

53 out of 110 (nearly 50%) students answered correctly that methods B runs more computational steps than method A and most of them were arguing that the maximum of the counter variable in methods B was increased before executing the loop statement and therefore B was computing 500 more steps than A.

5.2.3 Method A has a longer runtime

16 students (nearly 15%) answered that method A runs more computational steps than method B. Most of these 16 students did not give a detailed reason for their answer. But 4 student answers were quite interesting: "*I believe that A is computing more steps than B, because n = n + 500 is at the end of the method.*" (No. 120), "*method A is executing more steps than B, because in method A the statement 'n=n+500;' comes at the end of the method.*" (No. 46), "*a is computing more steps, because the addition is after the loop.*" (No. 59) and "*Method A: the loop is repeated 100 times and after this another 500 times, whereas in B n is increased to 500 before the loop.*" (No. 30). These students might think that an increase of the maximum of the counter variable after the loop still effects the number of repetitions. In the next section (5.2.4) we give another example of a student answer, which is very similar to the student answer No. 30. This could be a hint to a possible alternative conception which we did not have in mind before and did not find in recent literature.

5.2.4 *Equal runtime*

That Method A and Method B are executing the same number of computational steps was given as an answer by 32 of 110 students (nearly 30%). 23 of these 32 students argued that both methods have the same statements but only in a different order. "*Both same; because they have the same statements, but they are executed in a different order.*" (No. 114), "*I believe that both methods are executing the same number of statements, because it does not matter where n=n+500 is written.*" (No. 119), "*It does not have an effect in which line a statement is written.*" (No. 85), "*Same number of steps! Both algorithms are equal, with the exception of order*" (No. 29) and "*Both are executing the same amount of statements, because both methods are nearly equal, with the exception of n = n + 500 which is done later in A.*" (No. 62). This was exactly the alternative conception "Same A, Same B", which Stavy and Tirosh [9] discovered in didactics of mathematics and science and which Gal-Ezer and Zur [3] tried to connect to didactics of computer science.

8 of these 32 students explained that the runtime should be equal, because both methods have the same number of lines: "*both same; because they have the same number of statements.*" (No. 96) and "*both execute the same number of steps, because the number of the statements in A and B is the same.*" (No. 58). This alternative conception is also mentioned by Gal-Ezer and Zur [3] ('same number of statements, same efficiency') and is also fulfilling the intuitive rule 'Same A, Same B'.

Quite interesting explanations were given by 2 of these 32 students: "*Method A and method B are executing the same number of statements, because both are using the same values.*" (No. 50) and "*I believe that both methods are executing the same number of statements, because method A is printing 100 lines first and later another 500 lines; whereas method B is printing directly 600 lines.*" (No. 107). The alternative conception of the first student fulfills also the intuitive rule "Same A, Same B", but with values of variables instead of number of lines and the second student is arguing very similar to the 4 students from above (5.2.3), who could have thought that method A is computing more steps, because the later increase of the counter variable still effects the loop.

3 of these 32 students answered that both methods are executing the same number of computational steps, because both are using iterations with a fixed number of iterations. They recognize that both iterations do not have a termination condition – like a while(..) – but a fixed number of iterations. Unfortunately, a fixed number of iterations does not mean that both 'fixed numbers' are equal. "*because both methods contain iterations with fixed number of iterations.*" (No. 116) and "*because both are using iterations with fixed number of iterations.*" (No. 115). This alternative conception could also be an indicator for the underlying intuitive rule 'Same A, Same B', because if both methods are using the same kind of iterations (loop/for instead of while) they have the same number of iteration steps.

6. DISCUSSION

The evaluation of the first question of our test verified the existence of alternative conceptions or so-called misconceptions, which are following the intuitive rule 'same A, same B' [9]. Despite the very simple and not complex algorithm the alternative conceptions 'two programs containing the same statements (even if in different order) are equally efficient' [3], 'same number of statements, same efficiency' [3] and 'the same kind of iteration is equally efficient' occur. The last alternative conception is completely new to us and was not described by Gal-Ezer and Zur [3].

Therefore, the intuitive rule 'same A, same B' should be taken seriously as a possible barrier to learning.

Furthermore, another alternative conception emerges: 'The increase of a counter variable after a loop still effects the loop'. This alternative conception may not follow an intuitive rule, but could also explain problems with loops and runtime.

7. FUTURE WORK

As a next step we are going to evaluate the student answers of the remaining questions and compare them to the above mentioned results of this paper. On the one hand we hope to generalize the alternative conception 'two programs containing the same statements (even if in different order) are equally efficient' to 'two programs containing the same number of lines and similar (but not same) statements are equally efficient' and on the other hand we want to construct a system of categories of misconceptions and their underlying intuitive rules including the misconceptions from this paper. Assistance and tools for teachers and instructors to prevent learners from alternative conceptions will be developed with the help of this system.

8. REFERENCES

[1] Danielsiek, H., Paul, W., Vahrenhold, J. 2012. Detecting and understanding students' misconceptions related to algorithms and data structures. In *Proc. 43rd SIGSCE Comp. Sci. Ed.*, pp. 21-26.

[2] Fleury, A. E. 2000. Programming in Java: Student-constructed rules. *SIGCSE Bulletin*, 32(1), 197 – 201.

[3] Gal-Ezer, J., Zur, E. 2003. The efficiency of algorithms–misconceptions. *Computers & Education, 42*(3), 215-226.

[4] Holland, S., Griffiths, R., Woodman, M. 1997. Avoiding Object Misconceptions. *SIGCSE Bulletin, 29*(1), 131-134.

[5] Mayring, P. 2000. Qualitative Content Analysis, *Forum Qualitative Sozialforschung*, 1 (2), Art. 20.

[6] Mayring, P. Qualitative Inhaltsanalyse: Grundlagen und Techniken 2008, Beltz, 2008.

[7] Ragonis, N., Ben-Ari, M., 2005. A long-term investigation of the comprehension of OOP concepts by novices. *Computer Science Education, 15*(3), 203-221. DOI= http://dx.doi.org/10.1080/08993400500224310

[8] Schiek, D. 2014. The Written Interview in Qualitative Social Research. *Zeitschrift für Soziologie*, 43(5), 379-395.

[9] Stavy, R., Tirosh, D. 1996. Intuitive rules in science and mathematics: the case of 'more of A-more of B'. *International Journal of Science Education, 18*(6), 653-6.