ELSEVIER

Computers & Education □ (□□□□) □–□

COMPUTERS & EDUCATION

www.elsevier.com/locate/compedu

The efficiency of algorithms—misconceptions

Judith Gal-Ezer, Ela Zur*

Computer Science Department, The Open University of Israel, Tel-Aviv, Israel

Received 28 January 2003; accepted 17 July 2003

Abstract

The implementation of a new computer science (CS) curriculum in high schools which includes all the basic elements of traditional CS programs, motivated a research to determine how students conceive the very fundamental notion of efficiency. Since this was the first time that algorithm efficiency was integrated into a high school curriculum, our study was crucial for further implementation of the program. This paper describes a study that revealed misconceptions in perceiving the efficiency of algorithms by high school students. We discuss the results, provide some indication of the roots of these misconceptions, suggest ways to prevent them, and recommend further research.

Keywords: Programming and programming languages; Secondary education; Teaching/learning strategies; Improving classroom teaching

1. Introduction

A new computer science (CS) curriculum has been designed and is now being implemented in high schools (Gal-Ezer, Beeri, Harel, & Yehudai, 1995). While there has been considerable activity regarding CS curricula on the college and university levels over the years, on the high school level, development has been somewhat slower, due in part to the lack of adequate separation between CS and computer literacy or computer applications. The new curriculum currently being implemented is in a sense a break-through since it combines conceptual and practical computing issues in a zipper-like fashion, with an emphasis on the basics of algorithmics, and introducing, for the first time, notions of algorithmic correctness and efficiency.

The design of efficient algorithms to solve algorithmic problems is one of the most important research fields in CS. Good algorithm design is crucial to the performance of all software systems and therefore essential in every CS program of study. The study of algorithms gives the learner

* Corresponding author. Fax: +972-3-6460744.
E-mail address: ela@openu.ac.il (E. Zur).

0360-1315/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compedu.2003.07.004

insight into the problems involved in providing techniques for solutions that are independent of programming languages or other implementational aspects.

Efficiency and complexity are pervasive themes throughout the study of algorithms, however they are considered advanced issues and therefore were not included in high school CS curricula in the past. Even in many university programs, they were not usually included in the introductory course, but only when teaching data structures. Thus, it was important to design a study to examine whether the curriculum suits high school students and to suggest ways to revise the materials if that was found to be necessary, depending on the outcomes of the research.

Recognizing the important contribution that research on misconceptions in relation to mathematics and science education made to the teaching of these fields (Eylon & Lynn, 1988; Fischbein, 1987; Hart, 1981; Osborne & Freyberg, 1985; Perkins & Simmons, 1988; Stavy & Tirosh, 1996a, 1996b), we believed that exposing misconceptions in relation to the concept of efficiency was crucial to the examination of the CS high school curriculum.

Initial observations of the implementation of the program, interviews with teachers and students, and examination of students' achievements, confirmed our assumption that efficiency is a difficult concept to grasp. The goal of our study was to find out what misconceptions students hold regarding the notion of efficiency. We did not attempt to uncover the source of the misconceptions, nor did we systematically examine ways to prevent them. We will, however, provide some assumptions and clues regarding reasons for the misconceptions together with some ideas on how to prevent them; however, these need to be further investigated.

2. Background

Algorithmic thinking is a special and powerful way of thinking, which lies at the heart of computer science and represents the "spirit of computing" (Harel, 1992). As a consequence, the correctness and efficiency of algorithms—their computational complexity—are basic concepts in every CS curriculum (see, for example, ACM, 1968; IEEE, 2001; Merritt, Buen, East, Grantham, Rice, & Proilx, 1993; Tucker, Lidtke, Mulder, Rogers, Spafford, & Turner, 1991).

When given a problem, it is useful to compare several algorithms that solve it, and then select the "best" in terms of space and time. Space—memory space—is measured by elements such as the number and size of the data structures used, or the number of variables included. Time—run-time—can be measured in different ways. We can use an empirical approach, by programming and running the algorithms on a computer. The results of such a measure will be different depending on the computer used and, in some cases, may encourage belief in the myth that computers are so incredibly fast that there is no real problem of time, which is totally groundless. We would therefore recommend a universal measure for determining run-time. Such a measure can be calculated by computing the number of elementary actions carried out by the processor when executing the algorithm. This typically differs from input to input, but determines mathematically the quantity of execution time needed. The advantage of this approach is that it depends neither on the computer being used nor on the programming language, and allows us to study the efficiency of an algorithm when used on input of any size.

The notion of efficiency is deeply rooted in the theory of computational complexity, one of the three complexities with which computer scientists try to cope (Harel, 1992). Concepts like big-O,

decidability and undecidability, and questions like "P = NP?" relate to this, and are known to be difficult to conceive and difficult to teach and learn. However, in our study we do not relate to these concepts. Since we introduce efficiency at very early stages of the program, the study material relates only to efficiency measured by the number of elementary actions, as an introduction to future study of computational complexity, a field of study and research which considers the class of all algorithms that are able to solve a given problem.

In the assignments given to the students, as well as in the course material, they are asked to examine the efficiency of a given program in terms of execution time, and not in terms of big-O.

For the past 25 years or so, scholars have studied students' misconceptions regarding mathematics and science. Studies have shown that students' conceptions of scientific issues are often not in line with accepted scientific thinking; that is, they have misconceptions regarding various notions (Eylon & Lynn, 1988; Fischbein, 1987; Hart 1981; Osborne & Freyberg, 1985; Perkins & Simmons, 1988; Stavy & Tirosh, 1996a, 1996b, 2000). In their study of mathematics and science education, Stavy and Tirosh (1996a, 1996b, 2000) examined common thinking patterns underlying misconceptions in different content areas. They found that many misconceptions in mathematics and science stemmed from a small number of intuitive rules. They identified three different intuitive rules: 'more of A, more of B'; 'everything can be divided by two' and 'same A, same B'. Students consider these rules to be self-evident, and apply them with great confidence. By creating logical environmental relations, they appear to enable students to understand a given situation to a certain extent without having a real understanding of the concept.

In what concerns misconceptions of CS concepts, studies have mainly related to programming languages (Du Boulay, 1986; Putman, Sleeman, Baxter, & Kuspa, 1986; Saj-Nicole & Soloway, 1986; Stemler, 1989). Our study takes the research of misconceptions in CS education further, and investigates the perception of the notion of efficiency through an examination of the implementation of the new CS high school curriculum mentioned above.

3. The Study

3.1. Research hypotheses

Our assumption was that students' intuitive perception of the concept of algorithm efficiency is not compatible with the definition of efficiency in computer science. We assumed that the students' perceptions were the following:

- A shorter program (in terms of number of lines) is more efficient.
- Two programs containing the same statements are equally efficient (even if the order of the statements is different).
- The fewer variables there are, the more efficient the program.

We also examined to what extent 10th and 11th grade students succeed in internalizing efficiency and to what extent significant changes in their perceptions occur after they have studied efficiency.

However, in this paper we concentrate on revealing the misconceptions; the other questions are discussed elsewhere, and we will mention them here only briefly.

3.2. Research population

The study was carried out among 10th and 11th graders in five high schools where the new CS curriculum was being implemented. The curriculum has two versions: a 3-unit, 270 h version, and a 5-unit 450 h version (Gal-Ezer et al., 1995). The first two units (180 h), that provide the foundation for the entire program, are *CS Fundamentals 1* and *2* (Gal-Ezer & Harel, 1999). These units are studied 3 h per week in the 10th or 11th grades. Some schools begin teaching the program in the 10th grade, in which case the course is mandatory for all students, while other schools begin in the 11th grade, in which case only students who wish to major in CS study the material. We will not expand here on the reasons for this situation, but it is important to bear in mind that as a result, the 10th graders described in this study are not necessarily science-oriented, while the 11th graders probably are. The participants in this study were:

- 174 students in the 10th grade after they had studied seven chapters of the first unit—*CS Fundamentals 1*, including repeated execution, but before studying the chapter on efficiency.
- 141 of the same students at the end of the 10th grade, after completing the first unit.
- 145 students in the 11th grade, after completing *CS Fundamentals 1* and *2*.

3.3. Research instruments

The research instrument was a questionnaire including 14 multiple-choice questions, which dealt with various aspects of the concept of efficiency. Each question left room for the explanation of the choice of the response.

The 10th graders responded to the questionnaire before studying the chapter devoted to efficiency (pre-test) and again after completing the entire unit, *CS Fundamentals 1* (post-test). The 11th graders responded to the questionnaire after completing both *CS Fundamentals* units. In this paper, we will analyze only three questions which were relevant to the misconceptions we encountered.

Item analysis was used to enable us to differentiate among students' responses. Discrimination analysis is considered the key indicator of an item's value. We used point-biserial correlation (r_{pbi}) as an index of item discrimination. An item with $0.3 < r_{pbi} < 0.6$ is considered to be a good discriminator for differentiating between advanced and weak students (McNemar, 1955; Linn, 1989). In addition, we used McNemar's test of significance of changes, which is particularly applicable to pre/post designs (McNemar, 1955; Linn, 1989). We used the test to examine the significance of changes between the first administration of the questionnaire and the second for 10th grade students.

4. Results

The findings confirmed our assumption that students indeed have misconceptions regarding the notion of time-efficiency. As expected, students relate the efficiency of an algorithm to its length.

Also, we found that students thought that if two programs accomplish the same task they are equally efficient, and that if the algorithm uses fewer variables, it is more time efficient. We also found evidence of other misconceptions regarding efficiency on which we will not elaborate here, such as the misconception that a program is more efficient if its maintenance is simple, or if it contains more procedures, and the like.

Here we will concentrate on evidence for the more basic misconceptions: (1) a shorter program (in terms of code lines) is more efficient; (2) the fewer variables, the more time-efficient the program; (3) two programs containing the same statements (even if in different order) are equally efficient; (4) two programs that perform the same task are equally efficient. The first two misconceptions can be interpreted as being in line with Stavy and Tirosh's (1996a, 1996b) intuitive rule 'more of A, more of B' (a longer program, in terms of code lines, needs more execution time, or, in other words, is less efficient; and the more variables a program includes, the greater the execution time), while the other two are in line with 'same A, same B' (same statements or same task, same efficiency).

We will now present the responses to the three questions in the questionnaire that related to these misconceptions (the correct answer appears in bold face), and analyze the results. In our description of the results, we will also give the point-biserial correlation (in parentheses), to emphasize that in most cases the questions discriminated well among students, and thus the outcomes are indeed meaningful.

Question 1: The two programs below calculate the product of two given integers and print it, using only addition. Examine the efficiency of the programs in terms of execution time. Which of the three statements is correct? Explain your response.

A. **Program 1 is more efficient than Program 2.**
B. Program 2 is more efficient than Program 1.
C. Program 1 and Program 2 are equally efficient.

Program1

```
#include <iostream.h>
void main()
{
 Int a, b, mult, i, limit, unit;
 cout << "enter 2 numbers";
 cin >> a >> b;
 if (a < b)
 {
      limit = a;
      unit = b;
 }
 else
 {
```

Program2

```
#include <iostream.h>
void main()
{
 int a, b, mult, i;
 cout << "enter 2 numbers";
 cin >> a >> b;
 mult = 0;
 for (i = 1; i <= a; i++)
        mult = mult + b;
 cout << "the multiply is: "
        << mult;
 }
```

```
 mult = 0;
 for (i = l; i < = limit; i + + )
       mult = mult + unit;
 cout  < <  "the multiply is: "
       < <  mult;
}
```

Analysis of the findings revealed that only 19.44% of the 10th graders gave the correct answer without having learned about efficiency ($r_{pbi} = 0.267$). At the end of the 10th grade, some improvement was observed, but still only 27.5% of the students gave the correct answer ($r_{pbi} = 0.416$). More than half (55.9%) of the 11th grade students gave the correct answer at the end of the year ($r_{pbi} = 0.366$) (see Fig. 1).

Of the 10th grade students who answered correctly, few gave the correct explanation. This indicates that even when they answered correctly, it is likely that they did not fully understand the concept. It is interesting to examine the explanations given. Fig. 2 shows that in the pre-test, 75% of the students thought that Program 2 was more efficient and 55% explained that this was because it was shorter. In the post-test, 69.2% of the students thought Program 2 was more efficient and 56.7% gave the same explanation. At the end of the 11th grade, the situation was better: only 40.6% of the students thought that Program 2 was more efficient and only 34.3% explained this in terms of the length of the program. Here some learning has taken place. Analysis of this question reveals a misconception and provides some evidence that this misconception follows the intuitive rule 'more of A, more of B'. In addition, the results show that in the 10th grade, little learning has taken place, while a somewhat more optimistic picture is found in the 11th grade.

Question 2: Examine the efficiency of the two programs below in terms of execution time. Which of the three statements is correct? Explain your response.

A. **Program 1 is more efficient than Program 2**.
B. Program 2 is more efficient than Program 1.
C. Program 1 and Program 2 are equally efficient.

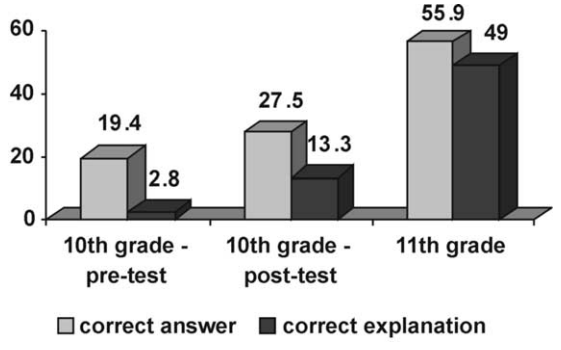

Fig. 1. Comparing the efficiency of two programs (Program 1 is longer than Program 2, but more efficient): Correct answer–correct explanation (in percentages).

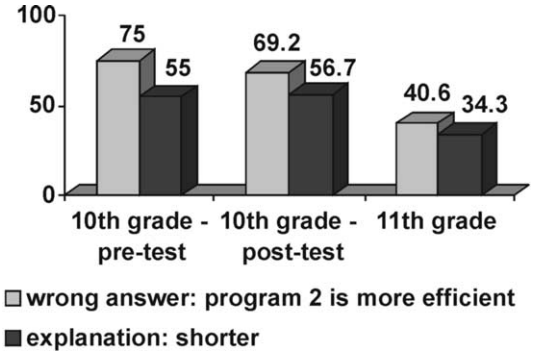

Fig. 2. Comparing the efficiency of two programs (Program1 is longer than Program 2, but more efficient): Wrong answer–wrong explanation (in percentages).

Program1

```
#include <iostream.h>
void main()
{
 int n, sum, number, i;
 sum = 0;
 cout << "enter number";
 cin >> n;
 for (i = 1; i <= n; i++)
 {
    cout << "enter number";
    cin >> number;
    sum = sum + number;
 }
 sum = sum * 100;
 cout << "the sum is: "
      << sum;
}
```

Program2

```
#include <iostream.h>
void main()
{
 int n, sum, number, i;
 sum = 0;
 cout << "enter number";
 cin >> n;
 for (i = 1; i <= n; i++)
 {
    cout << "enter number";
    cin >> number;
    number = number * 100;
 sum = sum + number;
 }
 cout << "the sum is: "
      << sum;
}
```

According to the findings, only 41.4% of the 10th graders gave the correct answer on the pre-test ($r_{pbi} = 0.534$), while at the end of the 10th grade, 53.7% of the students answered correctly ($r_{pbi} = 0.525$). Among the 11th grade students, 84% answered correctly at the end of the year ($r_{pbi} = 0.241$) (see Fig. 3).

At the beginning of the 10th grade, before studying the material related to efficiency, students seem to have followed the notion of 'same A, same B' (same number of statements, same efficiency). After studying the efficiency chapter in the 10th grade, there was some improvement (see Fig. 4). In the 11th grade, the few students who chose the incorrect answer followed the same notion.

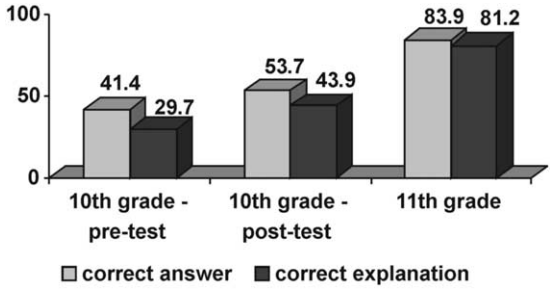

Fig. 3. Comparing the efficiency of two programs (of the same length), where a statement contained in the loop in Program 2 does not appear in the loop in Program 1: Correct answer–correct explanation (in percentages).

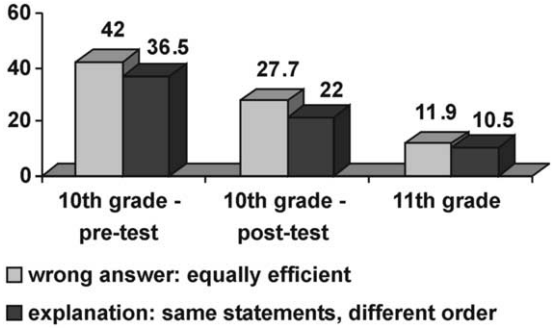

Fig. 4. Comparing the efficiency of two programs (of the same length), where a statement contained in the loop in Program 2 is excluded from the loop in Program 1: wrong answer–wrong explanation (in percentages).

Question 3: The two programs below print the input integers x,y within a given range (greater than x + y and smaller than x*y). Examine the efficiency of the programs in terms of execution time. Which of the three statements is correct? Explain your response.

A. Program 1 is more efficient than Program 2.
B. **Program 2 is more efficient than Program 1**.
C. Program 1 and Program 2 are equally efficient.

Program1

```
#include <iostream.h>
void main()
 {
   int num, x, y;
   cout << "enter 3 numbers";
   cin >> x >> y >> num;
   while (num != 0)
   {
     if ((num > x + y) &&
         (num < x* y))
```

Program2

```
#include <iostream.h>
void main()
 {
   int num, x, y, high, low;
   cout << "enter 3 numbers";
   cin >> x >> y >> num;
   low = x + y;
   high = x*y;
   while (num != 0)
   {
```

```
            cout << num;
    cin >> num;
  }
}
```

```
    if ((num > low) &&
        (num < high))
      cout << num;
    cin >> num;
  }
}
```

An analysis of the findings shows that only 16.5% of the 10th graders gave the correct answer in the pre-test ($r_{pbi} = 0.391$), while at the end of the 10th grade, 25% of the students gave the correct answer ($r_{pbi} = 0.453$). Surprisingly, even at the end of the 11th grade, only 36.4% of the students gave the correct answer ($r_{pbi} = 0.520$) (see Fig. 5). These are the most disappointing results we found. Less than half of the 10th graders who gave the correct answer in the pre-test also gave the correct explanation, indicating that the number of students who misunderstood the concept may be even greater.

We found that on the pre-test, close to 60% of the 10th grade students answered that Program 1 was more efficient. 18.6% thought it was more efficient because it contained fewer variables, and 26.9% thought it was more efficient because it was shorter (see Fig. 6). Both these explanations follow the intuitive rule 'more of A, more of B'. We were disappointed to discover that at the end of the 10th grade, the percentage of students who answered that Program 1 was more efficient even increased slightly (56.7%). 12.5% of the students explained that Program 1 was more efficient because it contained fewer variables, and 30% said that it was more efficient because it was shorter. At the end of the 11th grade the results were better, but still relatively disappointing: only 39.2% of the students gave the wrong answer, and 35% explained that Program 1 was more efficient because it contained fewer variables, while only 1.4% explained that it was more efficient because it was shorter.

A small percentage of students responded that since the two programs perform the same task, they are **equally** efficient, following the second intuitive rule 'same A, same B'. In the 10th grade on the pre-test, 10.3% of the students gave this answer; 5% of the students gave it at the end of 10th grade; and 10.5% of the students at the end of the 11th grade.

Apart from revealing misconceptions, we also found by means of McNemar's test that no significant learning took place in the 10th grade between the pre-test and the post-test. Only a few

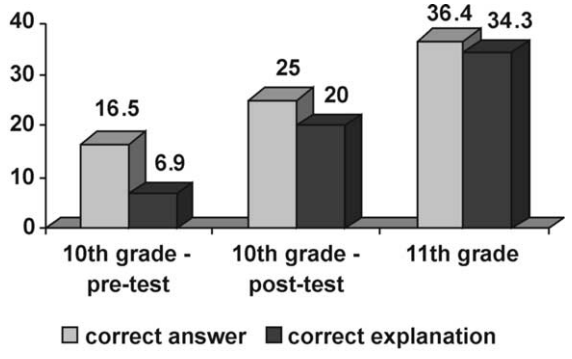

Fig. 5. Comparing the efficiency of two programs, where the number of variables in the more efficient one, Program 2, is larger than the number of variables in Program 1: correct answer–correct explanation (in percentages).

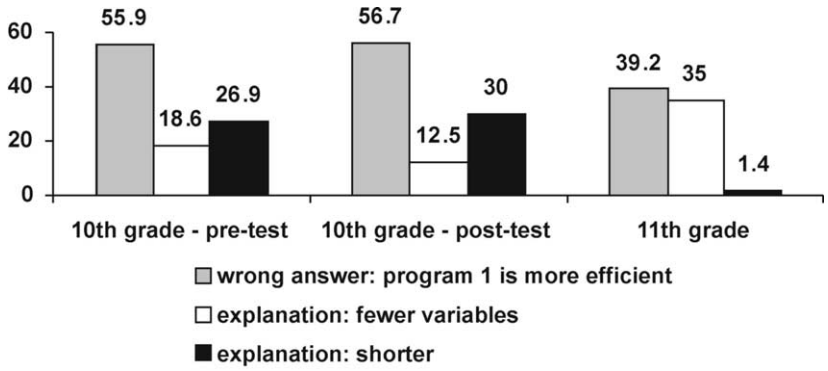

Fig. 6. Comparing the efficiency of two programs, where the number of variables in the more efficient program, Program 2, is greater than the number of variables in Program 1: wrong answer–wrong explanation (in percentages).

students changed their intuitive perception of the efficiency of algorithms. The 11th graders exhibited the same misconceptions, but to a lesser extent.

5. Discussion

The major importance of our study was in uncovering misconceptions regarding the efficiency of algorithms. Since this is a fundamental notion in the theory and practice of computer science, it is important to introduce the topic at a relatively early stage of every computer science program of study. In our research, we examined the implementation of a new high school computer science curriculum and confirmed our assumptions regarding the difficulties in teaching and conceiving of the notion of algorithm efficiency in high school.

Two of the misconceptions we encountered were that a longer program is less efficient, or, in other words, needs more execution time, and that the more variables a program includes, the greater the execution time. One possible explanation for these misconceptions is that the students adopt the intuitive rule 'more of A, more of B' which, as mentioned before, has been found to be one of the common thinking patterns underlying misconceptions in science and mathematics (Stavy & Tirosh, 1996a,1996b; 2000).

The other two misconceptions we encountered were that two programs containing the same statements (but in a different order) are equally efficient, and two programs that perform the same task are equally efficient. These two misconceptions seem to be in line with the intuitive rule 'same A, same B', also one of the patterns discussed by Stavy and Tirosh (1996a, 1996b, 2000).

One way to prevent or correct such misconceptions may be indirect intervention, which has been found to be effective in helping students to better understand complex terminology. Indeed, in a separate study, interviews with students were conducted to determine whether indirect intervention had an effect even in the limited time of an interview, and we found a significant effect on outstanding students (unpublished report). We recommend integrating such intervention into the written materials, or at least drawing teachers' attention to the need for it and recommending that they implement it in their teaching.

We recommend including in the materials algorithms that solve specific problems and then asking the students to design more efficient algorithms that perform the same task that the given algorithm performed, as suggested by Gal-Ezer, Vilner, and Zur (2003). In addition, we recommend including exercises similar to those used here: presenting two solutions to a problem, one next to the other. In this way, in order to analyze the efficiency of an algorithm, the student will concentrate, for example, on the number of elementary actions rather than the length of the algorithm.

Another explanation for the misconception 'the more variables, the more execution time' may be the fact that, in addition to following intuitive rules, most students had no knowledge of computers and how they operate. They had not learned anything related to computer organization before studying *CS Fundamentals 1* and *2*. This can be solved by developing a short unit on computer organization to be taught before the two *Fundamentals* units, and then reexamining students' perceptions of the notion of efficiency. In addition, animation can be incorporated into the existing materials (Kann, Lindeman, & Heller, 1997), to help to transform abstract concepts into more concrete examples.

The fact, mentioned above, that students in the 10th grade who are not at all science-oriented study this material because it is mandatory, together with the finding that no significant learning took place in the 10th grade, leads us to an important conclusion: though we believe that it is important to introduce the notion of efficiency relatively early in the course of CS studies, it may not be useful to do so in the 10th grade, at least not to the extent that it is currently taught. Not all 10th graders have enough mathematical background or sufficient algorithmic thinking to deal with this concept. This conclusion may yield radical rewriting of the materials; however, this could only follow a more focused investigation of this issue. More on the differences between the 10th and 11th grades regarding the perception of the concept of efficiency, can be found in Gal-Ezer and Zur (2002).

Finally, one point to which we have not related at all in this paper, though it was among our findings, should be mentioned: differences among teachers. We found that there were significant differences in the performance of students of different teachers. There is an acute shortage of good teachers, among other reasons, because of the high salaries offered in industry (Gal-Ezer, 1995; Poirot, Luehrmann, Norris, Taylor, & Taylor, 1985). This phenomenon needs to be examined carefully in an effort to determine (cautiously) what can be done. The current high-tech crisis may provide a partial, though unintended, solution to this problem.

In general, further research into the conception of theoretical computer science notions by high school students is highly recommended.

Acknowledgements

We thank Yael Alberton for her advice on the statistical analysis. The first listed author would like to thank the Weizmann Institute of Science for hosting her on a sabbatical leave, during which parts of this paper were written.

References

ACM Curriculum Committee on Computer Science. (1968). Curriculum 68: Recommendations for academic programs in computer science. *Comm. Assoc. Comput. Mach.*, *11*(3), 151–197.

Du Boulay, B. (1986). Some difficulties of learning to program. *Journal of Educational Computing Research*, *2*(1), 57–73.

Eylon, B., & Lynn, M. (1988). Learning and instruction: An examination of four research perspectives in science education. *Review of Educational Research*, *58*, 251–301.

Fischbein, E. (1987). *Intuition in science and mathematics: an educational approach*. Dordrecht, Netherlands: Reidel.

Gal-Ezer, J. (1995). Computer science teachers' certification program. *Computers and Education*, *25*(3), 163–168.

Gal-Ezer, J., Beeri, C., Harel, D., & Yehudai, A. (1995). A high school program in computer science. *Computer*, *28*(10), 73–80.

Gal-Ezer, J., & Harel, D. (1999). Curriculum and course syllabi for a high-school CS program. *Computer Science Education*, *9*(2), 114–147.

Gal-Ezer, J., Vilner T., & Zur, E. (2003, June). *Teaching efficiency at early stages: a different approach*. To be presented at ITiCSE, Thessaloniki, Greece.

Gal-Ezer, J., & Zur, E. (2002). *The concept of "algorithm efficiency" in the high school CS curriculum*. Paper presented at FIE 2002, Boston, MA.

Harel, D. (1992). *Algorithmics: the spirit of computing* (2nd ed.). Reading, MA: Addison Wesley.

Hart, K. (1981). *Children's understanding of mathematics*. London: John Murray.

IEEE Computer Society/ACM Task Force. (2001). *Computing curricula 2001*. Available: http://www.computer.org/education/cc2001/final.

Kann, C., Lindeman, R. W., & Heller, R. (1997). Integrating algorithm animation into a learning environment. *Computers and Education*, *4*, 223–228.

Linn, R. L. (1989). *Educational measurement* (3rd ed.). New York: Macmillan.

McNemar, Q. (1955). *Psychological statistics* (2nd ed.). New York: Wiley.

Merritt, S. M., Buen, C. J., East, P., Grantham, D., Rice, C., & Proulx, V. K. (1993). ACM model: High school computer science curriculum. *CACM*, *36*(5), 87–90.

Osborne, R., & Freyberg, P. (1985). *Learning in science: the implications of children's science*. Auckland, NZ: Heinemann.

Perkins, D. N., & Simmons, R. (1988). Patterns of misunderstanding: an integrative model for science, math, and programming. *Review of Education Research*, *58*, 303–326.

Poirot, J. L., Luehrmann, A., Norris, C., Taylor, H., & Taylor, R. (1985). Proposed curriculum for programs leading to teacher certification in computer science. *ACM SIGCSE Bulletin*, *17*, 1.

Putman, R. T., Sleeman, D., Baxter, J. A., & Kuspa, L. K. (1986). *A summary of misconceptions of high school BASIC programmers*. Occasional Report No. 10, Stanford and the Schools Project.

Saj-Nicole, A. J., & Soloway, E. (1985). But my program runs! Discourse rules for novice programmers. *Journal of Educational Computing Research*, *2*(1), 95–125.

Stavy, R., & Tirosh, D. (1996a). Intuitive rules in science and mathematics: the case of 'more of A–more of B'. *International Journal of Science Education*, *18*(6), 653–667.

Stavy, R., & Tirosh, D. (1996b). Intuitive rules in science and mathematics: the case of 'everything can be divided by two'. *International Journal of Science Education*, *18*(6), 669–683.

Stavy, R., & Tirosh, D. (2000). *How students (mis-)understand science and mathematics: intuitive rules*. New York: Teachers College Press.

Stemler, L. (1989). Effects of instruction on the misconception about programming in BASIC. *Journal of Research on Computing in Education*, 26–33.

Tucker, A. B., Lidtke, D. K., Mulder, M. C., Rogers, J. B., Spafford, E. H., & Turner, A. J. (1991). Computing curricula 1991: a summary of the ACM/IEEE-CS Joint Curriculum Task Force Report. *Comm. Assoc. Comput. Mach.*, *34*(6), 69–84.