In P. Romero, J. Good, E. Acosta Chaparro & S. Bryant (Eds). Proc. PPIG 18 Pages 266 - 274

Why Don't They Do What We Want Them to Do?

Shmuel Schwarz, Mordechai Ben-Ari

Department of Science Teaching, Weizmann Institute of Science, Rehovot 76100 Israel
moti.ben-ari@weizmann.ac.il
shmuels@weizmann.ac.il

Abstract. This paper describes an investigation into the factors affecting a student's decision whether to construct a state transition diagram in order to verify the correctness of a concurrent program, or whether to verbally verify the program. We conjectured that the advantages of the visual formal tool would cause it to be adopted as a routine part of the students' practice, but in fact the verbal description was the dominant method of their practice. This paper describes the reasoning that the students used in choosing a proof method. Psychological factors such as personal commitment and evaluation of effort turned out to be more important than the appropriateness of the tool for achieving the goal.

1. Introduction

A problem that beginning students of programming have is the lack of an effective model of a computer, that is, a mental representation of the algorithmic processes that occur during the execution of a program (Ben-Ari, 1998). This is particularly problematic in concurrent programs, because the interleaving semantics of concurrent programs significantly changes the concept of program flow (Ben-Ari & Ben-David Kolikant, 1999). In sequential programming there is one output for every input, while for concurrent programs a single input or initial state can give rise to many scenarios.

Verification of correctness is also much more difficult because sequential programs are functional in the sense that a correctness specification describes the output as a function of the input, making it possible to test a program for representative input values; for a concurrent program global assertions must be satisfied in *all* scenarios. The concept of interleaving is very difficult for students, making it hard for them to mentally simulate the execution of a concurrent program. Even if they could, programs cannot be tested in the usual sense because of the astronomical number of scenarios.

Today the leading technique for verifying concurrent programs is *model checking* (Cleaveland & Smolka, 1996). Model checking is based on the fact that most concurrent algorithms have finite state, so that it is theoretically possible to create the entire state diagram for a program. Since all possible executions are represented by paths in the diagram, correctness specifications can be verified by examining the diagram.

During the past decade, enormous progress has been made in turning this concept into practical tools by automating the construction and search of the state diagram using sophisticated algorithms (Holzmann, 2004). However, the state diagram itself is never explicitly constructed and certainly is never displayed because of its size.

Although model checking is a formal technique that is too complex to show to beginning students, nevertheless, we wish to introduce the concept of verifying correctness by constructing and analyzing a visual representation of the state diagram of the execution of a concurrent program. Visualization tools can improve learning if they are used correctly (Ben-Bassat Levy et al., 2003). However, tools will not be used at all if they are not in the repertoire of tools that the students recognize to be part of the practice of their community, and this will not come about unless the use of the tools is taught (Petre, 1995). We are interested in examining the connection between the course contents, the practice of the community of students, and the factors that influence their decision to use or not to use a tool.

This paper describes research that we performed in order to investigate the factors, psychological and other, that influence the students' willingness to engage with these advanced concepts of verification.

2. Concurrent programming and state diagrams

Here is an example of a simple two-process concurrent program in which a semaphore coordinates the execution of two operations K and G:

```
S: semaphore := 0;
```

process P1:	process P2:

signal(S)
K;
wait(S);
signal(S);
	wait(S);
	G;

A state of the execution consists of a triple: the location pointers of each of the two processes, and the value of the semaphore. The initial state is (signal(S), wait(S), 0), and from this initial state, all possible reachable states can be computed by considering all possible interleavings of the statements of the two processes. Figure 1 shows this state diagram, where X indicates that a process has reached its final statement and cannot continue executing, and forbidden indicates that a process cannot execute a statement because it was blocked when executing a wait statement on a semaphore whose value is 0.

In order to prove that the program is not correct it is sufficient to demonstrate one state or scenario that violates a correctness specification, while to show its correctness, we have to show that all states and scenarios fulfill the correctness specification. Con-

sider the correctness specification: *absence of deadlock*. It is easy to see that the program is not correct, because there is a deadlocked state in the diagram (state 11). However, the diagram is not really needed in this case, because we can reason verbally: First execute process `P2`; it will block immediately because the value of semaphore `S` is 0. Now execute process `P1`; the `signal` statement will unblock process

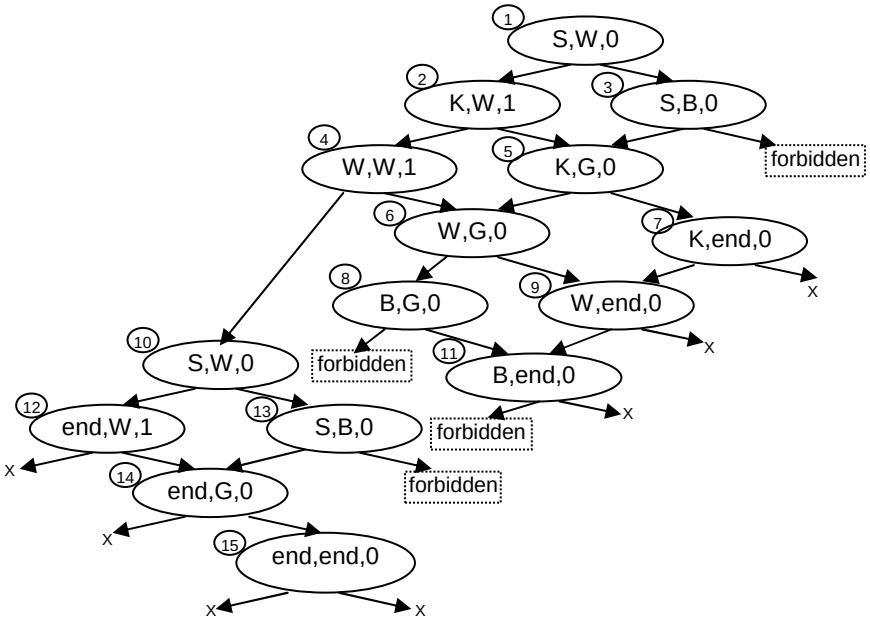

Figure 1: State diagram for the concurrent program

`P2`, while the value of the semaphore will remain 0. `P1` now continues execution and it will block at the `wait` statement. The system is now in a deadlock, because there is no process that can unblock the blocked process. It is more difficult to verify the correctness specification that `G` is executed in any scenario. With the diagram this is relatively easy to check by examining all the paths, while is very difficult to prove this verbally.

3. Methodology

The research was carried out in several classes of final year high school students who studied an elective theoretical course in concurrent and distributed computation. The population included 60 students in five classes from five different schools; each class was taught by a different teacher. We asked: (1) Would instruction in the use of state diagrams lead to its adoption within the practice of the students? (2) What considerations enter into a student's decision to use a state diagram?

To answer these questions, the students were given up to three worksheets containing problem-solving activities in concurrent programming in which state diagrams could be used. Each worksheet consisted of a concurrent program followed by six to eight claims, and the students were asked to prove or disprove each claim. The worksheets were analyzed in order to find out the number of occasions on which state diagrams were used, and to compare the correctness of solutions in which diagrams were used with those in which verbal (written) proofs were given. The students were allowed to work as individuals or in pairs (and in one occasion as a triple); altogether there were 31 groups. We encouraged students to work in pairs, because the group dynamics ensues encourages verbalization and thus facilitates understanding the students' thinking.

Follow-up interviews were used to deepen our knowledge about the decision-making processes that the students used. We encouraged the student to express his thoughts aloud, and from time to time asked him to explain or rephrase what he said. Each interview included three parts: (1) The student was posed a problem that he had solved in the past and asked to reflect on his work. (2) The student was asked to build a state diagram; this enabled us to ask questions about the building process itself, including technical questions as well as ones about their feelings. (3) The student was asked to prove or disprove a correctness claim; we let her decide whether to build a state diagram or not. From this part we hoped to understand her reasoning concerning state diagrams and thus to validate the information gathered from the first two parts. The interviews were analyzed into episodes that were used to create grounded categories that enabled us to infer the decision-making processes of the students.

Results

Since some of the groups worked on more than one worksheet, we obtained 44 worksheets for analysis. In 14 of them state diagram was used, while 30 used verbal proof. Solutions to problems were classified into four levels of correctness: (1) a correct solution; (2) a solution that was generally correct—a mistake in proving or disproving one or two claims out of six or eight; (3) a solution that was partially correct—mistakes in at least three claims; (4) an incorrect solution. Out of the 31 groups only 13 groups chose to build a state diagram for one or more problems, so we cannot conclude that the construction of diagrams is an integral part of their practice. Verbal proof remained the dominant practice with state diagrams taking only a secondary role. Figure 2 shows the correctness levels of these answers for each method of proof. There is a clear indication that the use of state diagrams is more effective for verifying correctness.

There is a basic assumption in education that when a student is presented with a problem to be solved, she will (attempt to) choose a method that leads to a clear and complete solution. The results of the analysis of the interviews show that this is not so, and that the decision-making process is far more complex.

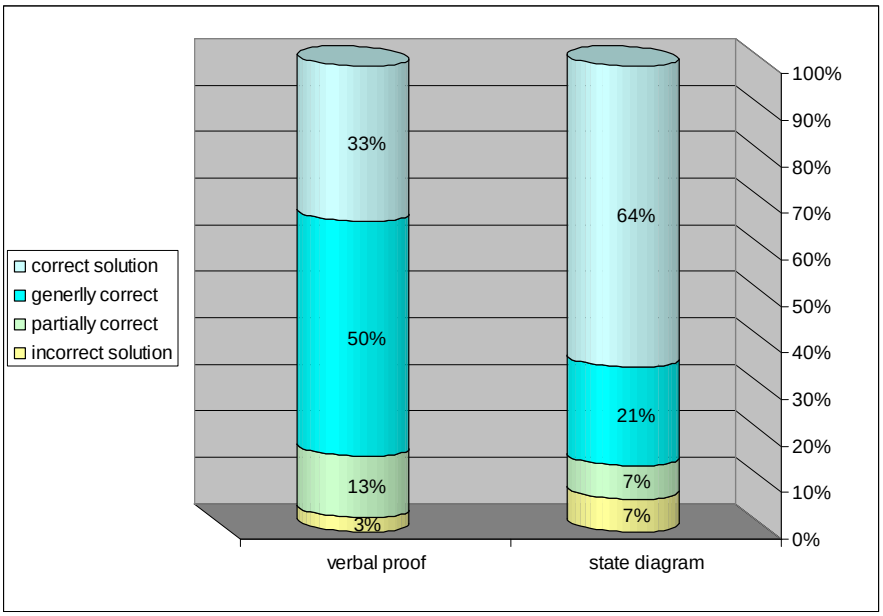

Figure 2 – Correctness levels for verbal proof and state diagrams

Because we are interested in the cognitive aspects of the decision-making process, we will present the results as a hypothetical internal dialogue of a student, composed of a composite of episodes that were identified among the students interviewed. It is important to note that the decision-making process given here was—in one form or another—characteristic of all the students who were interviewed. In the dialog, fictitious names are given, followed by the number of the grounded category (see the Appendix).

> I have to prove the correctness of a list of assertions. Should I choose to use a state diagram or a verbal proof? Never before have I been given a choice of methods!
>
> During the year I rarely used a state diagram in my work, except for exercises about the diagrams themselves (Nick, 3.2.1). Even so, I have seen and built diagrams, and they are an excellent visual tool for verifying programs; you simply see all the solutions at once (Renee, 1.1). But that experience was a bit problematical; it took me so long to build the diagram and it exhausted me so much, and just for one simple correctness specification. And during the exam with all the pressure, I had to erase everything and copy it from the beginning (Nick, 3.2.1).
>
> Was the diagram really necessary? Before, I have successfully used verbal proofs and I didn't really need the diagram; there was never an occasion

> where an exercise was of a level where a diagram was really needed and really important; most of the exercises were sufficiently simple that I could solve them in my head (Nick, 3.2.1, 2.2).
>
> My first gut feeling is that I'll build a state diagram anyway because its advantages are so clear; finally, it is ready, let's go, let's take a marker and indicate the answers. Otherwise you could mess around for half a page—why yes, why no—and in the end you haven't proved anything for sure (Renee, 1.2, 3.3.3).
>
> Wait minute ... this is an anonymous worksheet and the exercise is hard, so I'll pass on it, because no one will know that it is I; if the exercise is hard, I won't do it (Otto, 3.1.1).
>
> OK, I still have a bit of self respect, so I'll try anyway to answer the question; so we're back to the beginning: is it worth building a state diagram? The diagram is a powerful tool, but an expensive one to use; the more questions there are in a single exercise, the more it is worth my while to build it. If there was only one question, I wouldn't build a diagram, but if there are four more questions, I would invest the five minutes it takes and then each question would only take a minute or two (Abby, 3.2.1).
>
> OK, as always, when I have a dilemma I just decide whatever I feel like at that moment; it really is so much more fun to say “go right,” “go left,” draw a circle here, a circle there, and an X here. I don't know but there is something attractive about it, something more emotional as a student ... I don't know if they all feel this way (Nick, 3.2.1).

4. Discussion

Verbal proof was found to be the dominant practice that was adopted by the community of students, with the use of the state diagram having secondary importance. Students examined many considerations when deciding if to use a diagram; the primary consideration was weighing the expense of constructing the diagram against its value in verifying correctness. Another important consideration was the commitment of the student to solving the problem. Other considerations were: the difficulty of verbal proof, the student's experience in constructing diagrams, preconceptions and previous experience.

The thought process described in the previous section demonstrates that it is difficult to discover the “real reason” why a student would decide to use or not use a state diagram to verify correctness. The “real” (professional) reasons for using this tool are masked by a plethora of extraneous (personal) reasons that a student might give.

The students' use of a cost-benefit analysis is somewhat surprising because we are used to seeing things from a purely scientific or pedagogical viewpoint. We wish to believe that the student is committed first of all to the correctness of her answer with no extraneous considerations, while it seems that the students are willing to risk an incorrect solution if it lowers the cost of obtaining the solution.

The cost-benefit analysis that our students employed could be framed within Blackwell's (2002) model of measuring attention investment in *attention units*. The students had to decide whether they prefer "paying for" a somewhat more expensive but much more accurate formal tool, or whether to be satisfied with a cheaper but less accurate verbal description. Different students seem to calculate their attention-unit costs differently, and in extreme cases their analysis is biased by extraneous factors:

> It is as if I have something psychological against state diagrams. I just see them and already I … [grimaces] … I prefer not to do it. I have a psychological barrier against the state diagram. It's like in mathematics with that ln(x).

Finally, there was no risk factor for the students in our experiment, so this certainly affected their calculations.

An additional consideration is that a student will use a tool if she finds it somehow attractive. This type of consideration is not usually noticed, and it implies that hidden dimensions of reasoning might be the real motivation for certain behaviors.

We believe that not only does the relative importance of these factors differ among students, but they also change over time for an individual student. The hypothetical decision process that was presented demonstrates the wide variety of considerations taken into account by the students; it indicates why state diagrams were not always and why verbal proof remained dominant in practice. Although the students were repeatedly taught that verbal proof might not be a reliable means of verification (because it is difficult to ensure that all scenarios are covered), there are other considerations that outweigh that of reliability, which was paramount in the intentions of the course developers.

We found that state diagrams were not adopted as a primary element of the practice for their intended purpose—verifying concurrent programs. Nevertheless, there are three reasons why state diagrams should continue to form part of the syllabus: (1) Some students do use state diagrams despite the time and effort needed to build them, because they accept that it gives more certainty in proofs of verification; (2) There are some situations in which students using verbal proofs simply do not succeed and they are forced to take a more formal approach; (3) It is likely that many students use state diagrams as an aid to the formation of mental models, even if they do not build diagrams (cf. Thomas et al., 2004).

5. Conclusion

We conclude that it is essential to investigate the range of considerations that a student takes into account when trying to solve a problem. This will enable the teacher to increase the likelihood that a student will actually invest the time and effort to solve non-trivial problems.

Since we allowed the students to use either method of solution—without expressing a preference on our part—they had to engage in a decision-making process. Tallman & Gray (1990) distinguish between the act of deciding and choosing a preferred method to arrive at a decision: Deciding involves unknown situations with the associated uncertainty and risk taking, whereas choice involves known situations without uncertainty. They present a model for predicting the behavior of a subject which they call satisfaction balance. Our research has shown that there is a need for such models in order to enable researchers to recognize all the considerations that might influence the students' decision making.

6. References

1. Ben-Ari, M.: Constructivism in computer science education. Journal of Computers in Mathematics and Science Teaching 20(1), (2001) 45–73.
2. Ben-Ari, M., & Ben-David Kolikant, Y.: Thinking parallel: The process of learning concurrency. SIGCSE Bulletin 31(3), (1999) 13–16.
3. Ben-Bassat Levy, R., Ben-Ari, M., & Uronen, P. A.: The Jeliot 2000 program animation system. Computers & Education 40(1), (2003) 15–21.
4. Blackwell, A. F.: First steps in programming: A rationale for attention investment models. In Proceedings of the IEEE Symposia on Human-Centric Computing Languages and Environments, pp. 2–10.
5. Cleaveland, R., & Smolka, S. A.: Strategic directions in concurrency research. ACM Computing Surveys 28(4). (1996) 607–625.
6. Holzmann, G. J.: The Spin Model Checker: Primer and Reference Manual. Addison-Wesley, Boston (2004).
7. Petre, M.: Why looking isn't always seeing: Readership skills and graphical programming. Communication of the ACM, 38(6) (1995) 33–44.
8. Tallman, I., & Gray, L. N.: Choices, decisions, and problem-solving. Annual Review of Sociology 16 (1990) 405–433.
9. Thomas, L.A., Ratcliffe, M.B., & Thomasson, B.J.: Scaffolding with object diagrams in first year programming classes: Some unexpected results. SIGCSE Bulletin 36(1), (2004) 250–254.

Appendix – Table of Categories

There are three first-level categories based upon the research questions: what techniques do the students use and why? The next two levels contain the categories that resulted from the analysis; for example, 3.2.1 means that the student's self-confidence based upon her experience in building state diagrams was a factor in her decision. The criteria for assigning an episode to a category are listed in the last column.

First level	Second level	Third level	Criterion
1. Using the state diagram	1.1 visualization		Uses phrases from the visual domain: states in the diagram "show" states of the program
	1.2 proof		Uses properties of states and/or transition in the diagram to prove a claim
2. Using verbal description	2.2 imaging		Uses phrases like "I am doing it in my head" that indicate a mental image of the interleaved execution
	2.2 proof		Gives a verbal scenario based on lines of the program
3. Considerations for building a state diagram	3.1 commitment to the answer	3.1.1 formality	Sees the claim as a formal requirement, for example, on an exam
		3.1.2 required	Discerns between diagrams that are (implicitly) required or just options
	3.2 self-confidence	3.2.1 experience	Takes into account her experience in building state diagrams
		3.2.2 emotions	Expresses emotions about state diagrams
		3.2.3 size and complexity	Analyzes the expected size and complexity of the state diagram when assessing his confidence to successfully build a diagram
	3.3 cost-benefit analysis	3.3.1 time	Explicitly refers to the time needed in order to build the state diagram
		3.3.2 number of claims	Explicitly refers to the number of claims in the exercise