A PVS-Based Approach for Teaching Constructing Correct Iterations

Michel Lévy and Laurent Trilling

Laboratoire IMAG-LSR
B.P. 72, 38041 St Martin d'Hères, France
Michel.Levy@imag.fr, Laurent.Trilling@imag.fr

Just claiming the importance of formal methods is not enough, it is necessary to teach programming using formal methods. Also, we have to convince students to use them in their programming. To fill this goal, two points seem necessary: a no-fault approach combined with (apparently) affordable proofs and the use of automatic provers.

More than twenty years ago, David Gries and others said that the goal should be to forbid the construction of incorrect programs by teaching constructions of correct programs using a no-fault approach. This point of view appears to us to be both simple and challenging for students, because teaching correct program construction means teaching methodologies based on a process with well-defined steps which decomposes into sub-tasks, each of which is human in scope. Human scope means the sub-tasks are considered obvious or easy to prove by humans; for example, easy to prove sub-tasks preferably do not require inductive proof.

Formal pen and paper teaching of program construction methodologies using formal methods is not enough, since proofs by hand sometimes contain overlooked errors and, by not facing this reality, students do not develop the conviction to use these methods systematically. What is needed here are computer systems to check proof automatically. Using such systems challenges students to write correct proofs, and, in turn, motivates students to employ formal methods in their programming.

Our first objective relates to designing a system called CIA-PVS (for Constructions d'Itérations Assistées par PVS). This system is used in teaching a long known and well known methodology for constructing simple programs, i.e. loops. CIA-PVS is based on a well known proof-checker, PVS (for Prototype Verification System), which was developed at SRI (Stanford Research Institute). What is expected from the CIA-PVS system is that it reacts quasi-automatically to prove the lemmas necessary for the construction of programs which are traditional exercises such as the Dutch National Flag and the dichotomic research in an ordered list. What should be noted here is the simplicity of the lemmas to be proved. The real difficulty in constructing the program should not be the proof of these lemmas but the formalisation of the problem as the definition of the formulas expressing the result of the program, the invariants and the termination function of the iteration.

Our second objective relates to evaluating CIA-PVS for teaching programming via a methodology employing formal methods. In particular, the evaluation

J. Wing, J. Woodcock, J. Davies (Eds.): FM'99, Vol. II, LNCS 1709, pp. 1859–1860, 1999.

will be based on two criteria: automatic proof power and modelisation. Modelisation refers to the capacity to model easily the formal methods methodology so as to reduce as much as possible the gap between the formal teaching of the methodology and the concrete use of it in programming.

Our work began by constructing a PVS theory, called CIA-PVS, which proves the methodology itself. We need to prove it because, even if a methodology is, like this one, very well-known and appears to everybody correct, it is still possible that an error will arise as we attempt to formalise it precisely. Moreover, the use of this theory reduces the proving task of students, as desired, because the proof of the well-foundedness of the methodology is done once and for all. The use of subtypes provided by PVS to construct CIA-PVS has been very useful for reaching this goal. First experimentation on simple yet not trivial problems is encouraging. Once CIA-PVS is proved, power is clearly impressive in many cases and that is clearly positive. The remaining sensitive points are (1) some proofs may become easier or more difficult depending the chosen modelisation and (2) some proofs require a significant know-how level in PVS. The challenge for teaching remains both to define a starting knowledge of PVS to be taught to students and to extend CIA-PVS to deal with more sophisticated exchanges with students.