Writing Programs to Learn Mathematics

Ed Dubinsky, Georgia St. Univ., USA

There are a number of different ways of using computers to help students learn mathematics and each has its advantages and disadvantages. In my research and curriculum development, I focus on students performing specific computer activities that involve the construction on the computer of mathematical objects and processes, often by writing programs. The computer tasks for each particular mathematical topic are closely related to a certain theory of learning in the sense that they are designed to cause students to make mental constructions that (1) are proposed by the theory in connection with the topic and (2) are likely, according to research results, to help the students learn the mathematics involved.

After considering various ways of using computers, I will give a very brief sketch of the theory and provide some examples of computer activities based on it in relation to topics in discrete mathematics, calculus and abstract algebra. Finally, I will point to research results.

1 Ways of using computers

With today's powerful computers, complex pictures in two and three dimensions—both static and dynamic and closely connected with formulas and numerical data—can illustrate for students many profound mathematical concepts and even provide concrete representations of abstract mathematical structures. There is no question that this can have a salutory effect on learning, but there are caveats. The literature is beginning to be sprinkled with examples of functions, for instance, whose graph can not be produced fully on a screen and cannot be understood without a lot of analytic knowledge. Perhaps more important, however, is what an individual understands by looking at a picture. There is now considerable evidence that interpretation of a visual experience depends more on a viewer's understanding, with all its potential misconceptions, than on what is "really" in the picture. See, for example, Piaget (1969), (1977, p. 681).

Another way in which students can use technology is to run sophisticated algorithms to solve problems. With the touch of a few buttons, a student can integrate the most complex expressions, produce a Taylor polynomial with error estimate, or calculate the quotient groups of a fairly large group. Although there can be little doubt of the value of this for learning how mathematics can be applied, that is not quite the same as understanding the mathematics. A chauffeur may not need to know very much about how an internal combustion engine works in order to perform her or his driving duties, but a mathematics student really does need to have some feeling for why computing rates of change and calculating areas are inverse operations before we are willing to say that he or she understands much about calculus. Indeed, if one questions this need, then I would ask if one might not be questioning mathematics itself.

I believe that a more effective way to use computers is to have students construct the mathematics — on the computer — and then use what they have constructed to produce pictures and apply algorithms to solve problems. Of course, such a claim of effectiveness must be justified and I think this needs to be done through research. One way of studying the effectiveness of *any* pedagogical approach is to relate it to a theory of learning. We can use theory both to suggest the questions we ask in our research and to tell us how well we have answered them.

2 APOS Theory

The theory I work with is a constructivist theory according to which learning mathematics consists, *inter alia*, of dealing with problem situations by constructing mental actions, processes and objects (Figure 1) and organizing them in coherent schemas to use in dealing with the problems. I include here some very brief comments about APOS Theory and refer the reader to Asiala et al (1996) for more information.

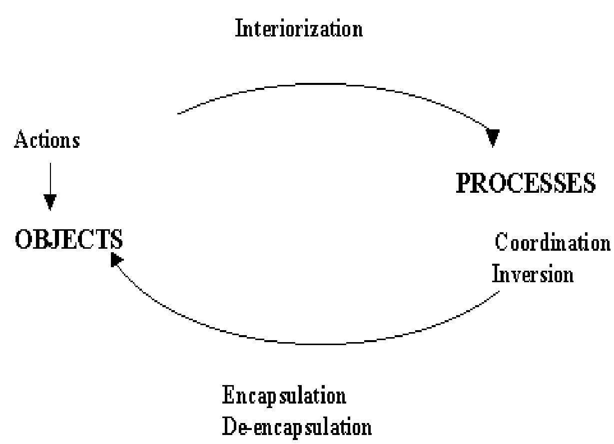

Figure 1

A transformation is considered to be an action when it is a reaction to stimuli which the subject perceives as external. For example, an action conception of function requires an explicit formula for each function. With such a conception, an individual is not able to do much more than replace a variable in a formula by a number or another formula and calculate.

When an individual reflects on performing an action, it can be interiorized to a process that the individual perceives as taking place in her or his mind. Thus with a process conception of function, an individual imagines taking one or more objects and applying some sort of transformation to obtain new objects. The transformation does not actually have to take place, but rather can be only a potential operation. With a process conception of function, an individual can combine two or more functions for example by adding, multiplying, dividing and substracting them, or by forming compositions of functions. At this point, he or she can imagine running the process in reverse and so begin to think about inverses of functions. It also becomes possible to think about the composition of functions. In a sense, it can be said that in passing from an action to a process conception of a mathematical concept, an individual moves from being controlled by the transformations through external means such as formulas or recipes, to exerting this control her or himself by imagining internal transformations.

When an individual reflects on operations applied to a particular process, becomes aware of the process as a totality, realizes that transformations (whether they be actions or processes) can act on it, and is able to actually construct such transformations, then he or she is thinking of this process as an object. In the case of function, an object conception allows an individual to construct operations on functions, form sets of functions and posit axioms to form, for example, a vector space of functions.

All of the actions, processes and objects that an individual has constructed relative to a particular mathematical concept can be collected together as a schema. A schema (which can include previously constructed schemas) is a totality that is coherent in the sense that the individual has an (explicit or implicit) understanding of when a phenomenon is within the aegis of the schema and when it is not. In the case of functions, this coherence consists in knowing what is and what is not a funcion, being able to distinguish a function from a relation, or finding a function in a situation.

A basic hypothesis of APOS theory is that many difficult mathematical concepts (e.g., the chain rule, Lagrange's theorem, construction of quotient groups) become more accessible to students once they have constructed appropriate actions, processes, objects and schemas. The way in which I use computers is to have students write programs that lead them to make these specific constructions on the computer and this can result in their making appropriate constructions in their minds. Reflection, in the classroom, on these computer tasks can help them construct appropriate understandings of the mathematics involved.

3 Examples of computer activities for mental constructions

Using an appropriate programming language, one can find tasks for students that correspond directly to each of the mental constructions of actions, processes, or objects for a mathematical concept. Consider, for example, a chemical experiment in which the compressibility of a substance varies with time and is given by three different expressions:

$t^3 + 1$ in the first second,

$2\sqrt{t}$ in the next three seconds

$\frac{t}{2} + 2$ thereafter.

An action conception of such a function is implemented simply by writing these three expressions and evaluating them for various values of t — with explicit instructions on which expression to use.

Students who have done this could then be asked to write a computer program that implements the function. Following is one example of such a program, followed by commands to evaluate it at two points.

```
c := func(t);
        if 0 <= t and t < 1 then return t**3 + 1;
        elseif t < 4 then return 2*sqrt(t);
        else return t/2 +2;
     end; end;

c(0.5); c(2.3);
```

When students write such a program, use it in the form of commands such as `c(0.5)`, `c(2.3)`, and discuss their experiences, they tend to develop a process conception of such a function.

One can go further and, after students have practiced with difference quotients for specific functions and specific differences, ask them to write a program that will accept a function and return a function whose value at x is the difference quotient determined by the points x and $x+0.0001$. Following is a program which does this job and, although it seems short and simple, it is difficult for students to write because of the novelty of using functions as inputs and outputs. The difficulty is not so much in the syntax, but in the strangeness of treating functions as objects.

```
ad := func(f);
         return func(x);
                   return (f(x + 0.0001) - f(x))/0.0001;
      end; end;

ad(c);
ad(c)(2.7);

plot([c,ad(c)]);
```

After writing such a program, students can reflect on what happens when they give commands such as `ad(c)` or `ad(c)(2.7)`. Moreover, they can plot the functions `c` and `ad(c)`, noting that the first has a corner at $x = 1$ but not at $x = 4$ and the second appears to be continuous except where the first appears to have a corner. All of this activity can result in students developing object conceptions of functions.

4 Results

We have implemented this way of using computers in courses in pre-calculus, calculus, discrete mathematics and abstract algebra. Research studies suggest that it is a useful strategy for helping students learn difficult abstract concepts. Descriptions of methodologies and summaries of results are given in Weller et al (in review). There is not sufficient space here for details, but perhaps the reader will be interested in the final conclusion of that report.

> Our results seem to point to the success of this theoretically-based approach as a valid tool by which students learn advanced mathematical concepts.

5 References

Asiala, A., Brown, A., DeVries, D.J., Dubinsky, E., Mathews, D., and Thomas, K., 1996. A framework for research and curriculum development in undergraduate mathematics education. *Research in Collegiate Mathematics Education II*, CBMS 6, pp. 1-32.

Piaget, J., 1969. The Mechanisms of Perception. (G. Nott Seagrim, trans.) New York:Basic Books.

Piaget, J., 1977. Mental Images. In H.E. Gruber and J.J. Voneche (eds), The Essential Piaget. New York:Basic Books.

Weller, K., Clark, J., Dubinsky, E., Loch, S., McDonald, M., and Merkovsky, R., in review. An examination of student performance data in recent RUMEC studies.