On Learning Quantification

Ed Dubinsky

Departments of Mathematics and Curriculum & Instruction

Purdue University, West Lafayette, IN 47907, USA

This paper reports on a study of students learning the concept of universal and exstential quantification in undergraduate mathematics courses in which the instruction was based on previous research into what it means to understand this concept. Following a theoretical analysis of the epistemology of quantification obtained in previous work, instruction is designed in which students construct mathematical concepts on a computer, using the mathematical programming language ISETL. The intention is that, as a result of making the computer constructions, students will tend to make effective mathemtical constructions in their minds.

The 36 students in the study were sophomores and juniors with joint majors in mathematics and computer science who were taking a course in discrete mathematics. The two groups were at two different universities and the same instruction, with minor variations, was given to both groups.

Instruments used for assessing student learning consisted of written questions on which students were to work individually to provide written responses. The questions were given in class without warning.

Results suggest that when the pedagogical approach described in this paper is used, students can develop some understanding of quantification and the ability to work with it, even when the particular problems they are given are difficult.

INTRODUCTION

The study reported in this paper is part of a comprensive, on-going program of research in learning and post secondary mathematics curriculum development. In this program a very specific paradigm is used in which research points to pedagogical strategies and their implementation leads to data that is then analyzed in ways that informs, and is informed by, the research.

The paradigm used in this program is a repeated traversal of the activities indicated in Figure 1.

INSERT FIGURE 1 HERE

Working in this paradigm for a particular topic in mathematics begins with an analysis of what it means to learn and understand the concept or concepts in the topic. The theoretical analyses are performed in the context of a general theory of learning, based on the ideas of Piaget which are applied to concepts in post-secondary mathematics.

The product of the theoretical analysis is a set of assertions about mental constructions an individual might make in order to construct her or his understanding of the concept. Instruction is designed using a number of pedagogical strategies that relate to the general theory and are focused on getting the students to make the asserted mental constructions. This instruction is then implemented and the students are observed while they are in the process of trying to learn the concept.

Up to this point, the interaction of the boxes in Figure 1 is uni-directional. That is, the theoretical analysis drives the instruction which thus is the source of the data coming from observations. But now, the interaction between the data and the theory goes in both directions. The theory suggests what questions might be asked in the analysis of the data: is such and such a mental construction apparently being made? On the other hand, the analysis has a profound effect on the theory. If students are learning without seeming to make the asserted mental constructions, or if they are doing so, but not learning, then the theory must be revised. As the revisions build up, the question of keeping or replacing the entire theory becomes relevant. The whole process is then repeated using the revised theory.

Assuming that the theory remains reasonably robust, repeated traversal of this circle is expected to lead to successive improvement in student learning.

This paradigm has been applied, over the last 10 years to a number of topics and courses leading to research results, textbook, and new curricula. The areas in which there has been the most work include: functions (Ayres, Davis, Dubinsky, & Lewin, 1988; Breidenbach, Dubinsky, Hawks & Nichols, 1992); mathematical induction (Dubinsky, 1986, 1989; Dubinsky & Lewin, 1986), calculus (Dubinsky & Schwingendorf, 1991) and abstract algebra (Dubinsky, E., J. Dautermann, U. Leron, & R. Zazkis, 1994; Dubinsky & Leron, 1994; Leron & Dubinsky, 1995).

This paper is a continuation of work in this paradigm on predicate calculus or (universal and existential) quantification begun in Dubinsky, Elterman, & Gong (1988) where we presented a theoretical analysis of the concept of quantification and how it might be learned. This represents the first step in a first iteration of our paradigm.

Our purpose in the present paper is to describe the next steps of this first iteration: instructional treatments based on the analysis and some results regarding student learning.

The instructional strategy consisted mainly of having students make various constructions on the computer using the programming language ISETL (see Baxter, Dubinsky, & Levin, 1988; Dautermann, 1992; Schwartz, Dewar, Dubinsky, & Schonberg, 1986), followed by classroom discussion of mathematics concepts corresponding to these computer tasks. There was also a certain amount of paper and pencil work for the students to do, both in and out of class. In general there appeared to be a reasonable amount of learning of this very difficult, but critical concept in mathematics. This is consistent with similar results we have had in helping students learn the concept of function (Ayres, Davis, Dubinsky, & Lewin, 1988; Breidenbach, Dubinsky, Hawks & Nichols, 1992) and mathematical induction (Dubinsky, 1986, 1989).

The observations were written instruments consisting of questions aimed at indicating students' understanding of quantification and their ability to reason about it.

The results reported in this paper were obtained several years ago. Since then, although our method has been and is being used in a modest number of schools throughout the world, and the users are apparently satisfied (since they continue to use the method), there has been no attempt to make a systematic study of the effect of this instructional treatment on the development of student understanding of predicate calculus since the present work. As far as we can tell, these two papers are the only reports on attempts to make improvements in students' learning quantification at the collegiate level.

In the remainder of this paper we give a brief outline of our general theory and recall how it was used in Dubinsky, Elterman & Gong (1988) to construct a genetic decomposition of quantification, which is what we a call a set of assertions about the mental constructions involved in learning a particular mathematical concept. Next, in the following section, we give a brief description of the programming language ISETL, its relation to our analysis of what the learner must do to construct the concept of quantification and how it is used in our instructional treatments.

In order to get some idea of the extent to which the students were coming to understand quantification, they were asked to write the answers to various questions, either in class or as take-home assignments. The questions are given in the section INSTRUMENTS below along with a description of how they were treated and their relation to the problems that were used as illustrations in class. The results are tabulated and discussed in the section on RESULTS.

The paper ends with a summary and interpretation of our results.

It is important to emphasize that the data presented in this paper are not offered as formal statistical evidence. Our suggestion is that the reader consider them as global and vague indications of the possible extent to which learning did or did not take place. We are taking advantage in this study of what we think is a universal experience with quantification. It is our impression that it is a very difficult concept for undergraduate mathematics students and one would not expect them to perform very well with the kinds of questions in the section on INSTRUMENTS. Therefore, if the results can be construed as satisfactory performance in the context of an ordinary classroom situation in which the time spent on the topic of quantification was not excessive and the questions required more than imitation of behavior, first illustrated in class or the text and then practiced in homework, then something worthwhile may have taken place and one might be justified in pursuing such an approach. It is on this basis that we attempt to evaluate and interpret the results. In addition to conclusions about what has taken place through the present iteration, we also make some observations of phenomena that are not addressed by the present version of our analysis of quantification and which provide material for extensions of this analysis that can be studied in future iterations.

THE GENETIC DECOMPOSITION OF QUANTIFICATION

The first step in our paradigm is to attempt to understand something about how a person might learn a particular concept. This is based on a general theory of learning mathematics, clinical observations of students in the process of learning the concept in question, and the researcher's understanding of mathematics involved in the concept. This epistemological analysis leads to a certain kind of description of how the concept might be learned. We call this description a *genetic decomposition* of the concept.

General theoretical perspective

Before reviewing the genetic decomposition of quantification that was obtained in Dubinsky, Elterman and Gong (1988) we describe the briefly the theoretical perspective on which this analysis is based.

We begin with a very general statement about the nature of mathematical knowledge and how it can develop in an individual.

> An individual's mathematical knowledge is her or his tendency to respond to mathematical problem situations by reflecting on them in a social context and constructing or reconstructing mathematical actions, processes and objects and organizing these in schemas to use in dealing with the situations.

There are many important issues raised by this statement that could be discussed, but for the purposes of this paper it suffices to consider the actions, processes and objects which are the mental constructions an individual makes in order to deal with a mathematical problem situation. These constructions are illustrated in Figure 2.

INSERT FIGURE 2 HERE

The constructions are the actions, processes and object whereas the mechanisms for making these constructions, which Piaget has named *reflective abstractions*, are the interiorizations, coordinations, reversals, encapsulations and de-encapsulations.

Action. A transformation is considered to be an action when it is a reaction to stimuli which the subject perceives as external. The action tends to control the individual.

Consider, for example, the proposition,

Every member of my family is unemployed.

If an individual can only think of this statement in terms of the reality of certain specific members of her or his family and whether or not they have jobs, then he or she is understanding this statement as an action. In this case, there would be no indication that the individual can think of all members of a perhaps extended family and imagine whether the statement is true or not, because the individual can only focus on a specific individual who may or may not be employed.

Process. When an individual reflects on an action scheme, it can become perceived as a part of the individual and he or she can establish control over it. It can be considered without needing to be performed explicitly.

In the above example, it is possible for an individual to think about all members of her or his family, or even someone else's family, to imagine checking the employment status of each individual, and to decide on the truth or falsity of the statement by summarizing all of the data obtained.

Object. 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 our example, and individual might think about applying negation to this statement and so imagine it being true, say for one family, but false for another.

In this way, the statement becomes an object that has a different status depending on the family in question. For each family, however, there is a definite proposition which ay be true or false.

Application to quantification

We will refer to Figure 3 in describing the genetic decomposition of quantification that was developed in Dubinsky, Elterman & Gong (1988).

INSERT FIGURE 3 HERE

The construction begins with simple declarations that may be true or false. These are the basic mental objects that form the building blocks for all that follows. They are made more complex in two ways. First, several such objects are collected in a set and, making use of the learner's schema for function (the existence of which is a prerequisite for understanding quantification), variables are introduced to obtain proposition valued functions, interpreted as processes. This means that the individual is able to think about iterating through the functions domain, checking the truth or falsity of the proposition for each value of the variable. Second, two or more declarations are coordinated by linking them with the standard logical connectors of either conjunction or disjunction. Combining mental actions that are performed on the objects and must be interiorized to form mental processes.

Once these two processes have been constructed, the learner is ready to move on to the construction of single-level quantifications, which consist of a single quantification, universal or existential, applied to a proposition valued function of one variable. In the digra, this indicated by the joining of the two large branches at a node in the middle of the page. The transition is achieved by coordinating the two processes previously constructed. That is, the propositions that are obtained for the various values of the variable are all connected by conjunctions (for universal quantification) or disjunctions (for existential quantification) resulting in a single proposition. Thus the learner interiorizes a process of iterating a variable through its domain to obtain a set of propositions to which a quantifier is applied so as to obtain a single proposition.

Next comes two-level quantifications, displayed below the middle ofthe diagram, in which two (usually different type) quantifiers are applied in succession to a proposition valued function of two variables. The process which we just described for constructing single-level quantifications must be encapsulated so that the result becomes a proposition which is a mental object. Note that the effect of the process is to eliminate a variable. If the original proposition valued function had two variables, then the resulting object actually depends on the value of the other variable and the schema for single-level quantifications can again be applied to this proposition valued function. Thus, when analyzing a statement which requires a two-level quantification over two variables, the subject begins by parsing it into two quantifications. There is an inner quantification over one of the variables in a proposition valued function of two variables. There is also an outer quantification over the other variable. What we have described is a coordination of these two quantifications to obtain a third which will be two-level quantification. In order to proceed to higher-level quantifications this new process must again be encapsulated to obtain a single proposition.

Given a statement which is a three-level quantification, the subject can group the two inner quantifications and apply the two-level schema to again obtain a proposition which depends on the outermost variable. This proposition valued function is then quantified as before to obtain a single proposition. The entire procedure can now be repeated indefinitely to obtain quantifications of any level.

Getting students to make mental constructions

The next phase in our paradigm is to design instructional activities that will induce the students to make the desired constructions. This is a much more delicate operation than is perhaps generally assumed. The point is that if students are mentally active, if they are listening to lectures, doing homework, studying for tests, then they will be making mental constructions. It is by no means certain that they will make useful constructions - that is, that they will construct schemas that will help them do mathematics. Perhaps more likely, they will construct ineffective procedures which some have called “bugs”. Examples of this include the observations of students’ (and teachers') errors in solving the Student/Professor problem (Clement, Lochhead, & Soloway, 1980), the "repairs" of Brown and vanLehn (Brown & vanLehn, 1980; vanLehn, 1980) misconceptions in Physics (DiSessa, 1985), the almost universal difficulty with Calculus (Douglas, 1986), and the trouble students have with implications (see the section, CONCLUSIONS for some indications of this). The literature reports not only on the initial errors that students make, but on the persistence of misunderstandings and on the lack of success that education seems to have in eliminating them. One possible explanation could be that the student's total experience leads her or him to construct an incorrect schema which is then reinforced by doing a large number of examples and misinterpreting explanations and illustrations.

We have found that computer experiences can be an effective way of not only helping students to construct reasonable schemas, but also to get them to reconstruct erroneous or incomplete conceptions. The basic principle is that anytime you construct something on a computer then, whether you are aware of it or not, you construct something in your head. By studying the connections carefully, we have found it possible to induce a considerable amount of learning. Specifically, in our theory, the main construction activities are interiorization, encapsulation, coordination, generalization, and reversal. These have counterparts in computer activities which we will describe in the next section. These activities can be used to get students to perform corresponding mental constructions.

Summary of mental constructions

Before going on to the instructional treatments and the description of the programming language ISETL that was used, let us summarize the most important mental constructions that, according to our analysis, the learner must make in order to acquire the concept of quantification. These are the activities that the teacher must try to facilitate. The design of computer work in our instructional treatments was driven by the goal of finding tasks that would tend to lead students to make these particular mental constructions.

1. Coordinate two or more declarations by linking with logical connectors.
2. Generalize the schema of function to include proposition valued functions.

3. Interiorize the action of iterating over the domain of a variable in a proposition valued function, checking the truth or falsity of the proposition for each value of the variable.
4. Coordinate linking declarations with iterating over a proposition valued function and apply a quantifier.
5. Encapsulate the process for a single-level quantification to obtain a single proposition.
6. Coordinate two instantiations of the single-level quantification to obtain a single proposition.
7. Encapsulate the process for a two-level quantification to obtain a single proposition.
8. Coordinate three or more instantiations of the single-level quantification schemas to obtain higher level quantifications.

In the next section we will try to explain how computer activities with ISETL can relate to each of these mental constructions.

USING ISETL

ISETL is an interactive programming language which runs on microcomputers and mainframes. It is based on the programming language SETL developed by Jacob Schwartz and his group at the Courant Institute. Full descriptions of these languages are found in Schwartz, Dewar, Dubinsky & Schonberg (1986) for SETL and Dautermann (1992) for ISETL. Baxter, Dubinsky, & Levin (1988) is a text for a discrete mathematics course using ISETL. Chapters 2 and 5 contain the complete treatment of propositional calculus and quantification on which this paper is based. (This book refers to an early version of ISETL, but an update of the text is in preparation.) We will give here only a very brief description of ISETL, concentrating on how it relates to the mental constructions listed at the end of the previous section.

ISETL is an interactive, interpreted language which means that, once it is invoked, expressions are typed on the terminal and, when a carriage return is entered, the system performs the activity indicated by the expression. This can be an evaluation, in which case a result is printed on the screen, an internal action such as an assignment to a variable, or some communication with the external system, such as writing to a file.

The syntax for simple constructions such as naming variables, assignments, if then clauses, loops, etc., is standard and very similar to Pascal. There are, however, almost none of the usual programming requirements such as type declarations, sizing, etc. As a result, the basic syntax of the language is very easy to learn and our experience with students suggests that relatively little time need be spent on teaching the language directly. Almost all of the students' ability to program in ISETL comes from working with mathematical constructions.

The syntax for operations and for the construction of complex objects is almost identical to standard mathematical notation. As a result, the student is often under the impression that he or she is studying the syntax of ISETL when, in fact, important mathematical ideas are being learned. This tends to make some things relatively painless.

ISETL supports standard types of values which can be entered directly, assigned to a variable, or obtained as the result of a calculation. These include integers, reals (actually floating-point approximations to real numbers), character strings, and Boolean. The latter is the set of two values, $\{true, false\}$ and this permits the representation of simple declarations. The standard logical operations are supported and so it is possible to enter an expression such as

```
((P and Q) impl (R and P)) or (not Q);
```

The semicolon at the end is the command to actually evaluate the expression and so, if Boolean values for `P`, `Q` and `R` were previously assigned, then either true of false will appear on the screen.

The main use of this feature is to ask students to translate logical expressions given in English to ISETL, evaluate the expressions, write programs to construct truth tables, and so on. These activities relate to item 1 in the list of mental constructions at the end of the section on genetic decompositions above.

Procedures in ISETL are called `funcs`. Students are asked to construct `funcs` that represent proposition valued functions. For example, if the following code is entered,

```
P := func(n);
        return n mod 223 = 0;
     end;
```

then a proposition valued function has been constructed to represent the statement,

A number is divisible by 223.

This relates to item 2 in the list at the end of the section on genetic decompositions. The idea, in item 3, of iterating over the domain and checking the value is suggested by asking the students to evaluate `P(892)`, `P(100)` and many other values. It is not difficult, when the students have actually done this to get them to think about what the computer is doing when it makes those evaluations. We call this thinking ISETL.

The coordination of items 1 and 3, which is item 4 in the list of mental constructions, involves the idea of a set which is a fundamental mathematical concept and another type of value in ISETL. Sets in ISETL must be finite. Following are some examples of simple ways to construct sets in ISETL,

```
{3, 7, 9};
{-2, 8.65, "a", {6, true, 87}};
{7 .  .  26};
{900, 902 .  .  1100};
```

Notice the mixing of value types and the nesting in the second set. The third set is the set of all integers from 7 to 26 and the fourth is the set of all even integers from 900 to 1100. (There is also a more powerful method of constructing sets in ISETL that is very close to set formation in mathematics. The syntax is

```
{expr :  x in S | P(x)}
```

but this will not be of concern in this paper).

At this point the student works with lists of propositions. For example, there could be 5 numbers $b_1, b_2, \ldots b_5$ and a situation is created where the student would form the conjunction of each of the propositions,

$$b_i > 2, \quad i = 1, 2, \ldots, 5$$

In ISETL this would look like

```
(b1 > 2) and (b2 > 2) and (b3 > 2) and (b4 > 2) and (b5 > 2);
```

There is an ISETL construction that represents this with a combination of a proposition valued function and a universal quantification. One first constructs the function,

```
F := func(b);
          return b > 2;
     end;
```

and the set,

```
B := {b1, b2, b3, b4, b5};
```

Then one can enter

```
forall b in B | F(b);
```

and, the result, true or false, depending on the values of the b's will be printed on the screen.

As another example, the student might be asked to express in formal language a statement such as

Out of all of the integers between 900 and 1100, one is divisible by 223.

Using the `func P` defined above and assuming that S has been assigned the value `{900, 902  1100}`, then taking ISETL to be the formal language, the student can write

```
exists n is S | P(n);
```

A major assumption of our approach is that coordinating the two processes on the computer to construct such expressions that can be manipulated in various ways, and reflecting on what has been done, will lead the student to construct a process corresponding to a single-level quantification, and interiorize it in her or his mind.

The encapsulations required for items 5 and 7 seem to be very difficult for students. Sfard (1991) considers the overall difficulty of encapsulation and suggests that this mental construction may be possible for only a fraction of students. It is here that certain special features of ISETL are particularly helpful. For example, if the above constructions have been made, one can perform the assignment

```
Q := forall b in B | F(b);
```

The variable `Q` can then be used in any expression combining Boolean values. Our idea is that performing such an action will make it more likely that a student will reflect on this single-level quantification, see it as a totality and even convert it mentally into an object which is the requirement of item 5 in our list. A similar procedure can be followed for two-level (and higher) quantifications, once they have been constructed as processes.

To satisfy item 6 by coordinating two single-level quantifications it is necessary to exploit the fact that a process such as a `func` in ISETL is a piece of data and can be made an element of a set, assigned to a variable, or returned as the value of another `func`. Suppose, for example, the student is asked to analyze the following statement and express it in formal language.

> There is a number in the set S which is dominated by every number in the set B.

The first step would be to replace the function `F` above by the following function of two variables,

```
G := func (b,n);
         return b > n;
     end;
```

Then the fairly complicated coordination can be implemented in two steps. First one "quantifies out" the variable b in G to obtain the following proposition valued function of one variable,

```
G1 := func(n);
          return forall b in B | G(b,n);
      end;
```

and then the coordination is achieved by entering,

```
exists n in S | G1(n);
```

Actually one can go even farther with this delicate point. It is possible in ISETL to automate all or part of the process. For example, one can construct a `func` that will accept as input a set and a proposition valued function of two variables. It will then return a proposition valued function of one variable obtained by "quantifying out" the other variable. The whole idea of "returning a function", that is, to work a problem whose solution is not a number but a function is very new and profound for many students. Thus this activity helps with three important mathematical concepts: quantification, finding a function as the solution to a problem, and replacing a function of two variables by a function of one variable whose values are functions of one variable.

INSTRUCTIONAL TREATMENTS

Two classes were involved in these investigations. The first, which we will call Class1, was a course entitled Introduction to Finite Mathematics given at Clarkson University during Fall 1986. There were 19 students, mostly sophomores, and they were following a joint major in mathematics and computer science. Understanding quantification was a major goal in this course.

The second course, which we will call Class2, took place the following semester. The course title was Introduction to Analysis and the 17 students were mainly Juniors, again with the joint mathematics and computer science major. The purpose of this course was to provide for the students a transition between the skill-oriented Calculus courses that they took in their first two years and the more theoretical Advanced Calculus course which they were required to take in the third or fourth year. Quantification per se is not usually a topic in this second course and relatively little time was spent with it. This particular semester, the course was done as an experiment which was the beginning of an investigation of applying our general approach to analysis. Clearly, quantification is an integral part of most arguments in analysis and so it seemed reasonable to spend at least a minimum amount of time discussing it explicitly. Class1 met twice a week for 75 minutes each. One meeting, on Tuesday, was in a lecture situation in which no computers were present and the other, on Thursday, was in a computer laboratory where the students were required to perform certain tasks that were given to them in written form. In the lab, students worked individually, each with her or his own terminal. Assistants were available to answer questions, give hints, help with syntax, and so on. Each week, the students were required to turn in a homework assignment on Tuesday. Actually, very little lecturing was done on Tuesdays. Much of the time was spent in discussion of the homework or the previous lab. Occasionally it was necessary to make some explanations that the students would need in the coming Lab session. Another classroom activity consisted of students working paper and pencil problems that were set by the instructor. The main purpose of the Lab sessions was to make sure that every student had at least attempted to perform certain computer tasks before coming to class. The idea was to present the students with disequilibrating problems so that they could make useful mental constructions in order to get past the difficulty.

During class, the goal was to get the students to reflect on what they had been doing in Lab and for the homework which had just been turned in. On occasion, the instructor tried to lead students (e.g., by verbal suggestion, pointing out possibilities, asking questions) towards making certain coordinations of processes that the students may have interiorized. The instructor also emphasized the encapsulation of processes into mental objects again by asking leading questions and by pointing it out when it occured. The in-class assignments were another way of presenting disequilibrating situations. They were collected and used to help determine what happened during the following week. They were also used, along with the homework, to evaluate the learning that may have been taking place. All of this work will be described and discussed in the remaining sections of the paper.

The work on quantification in Class1 was not done in isolation. It took place over a six-week period, but at the same time that other topics, in particular, sets and functions occupied a major portion of the course. Spread out through this period, a total of three lectures, four assignments, and three labs were devoted to quantification.

Roughly speaking, one can say that a little less than a month or about one-fourth of the course was concerned with quantification and its applications. The material that was used can be found in Baxter, Dubinsky, & Levin (1988) which was being written in conjunction with giving the course.

The treatment in Class2 was different in several respects. The class met twice weekly in a lecture situation and there were no labs. There was, as before, a weekly assignment. The students were left entirely on their own to learn ISETL from the material in Baxter, Dubinsky, & Levin (1988) (which existed in manuscript form by this time) as required to do homework assignments. The only lecture that was concerned with quantification per se concentrated on two- level quantifications. This reappeared in a lecture two weeks later (after the evaluations were made) as the key element in the definition of the limit of a sequence.

Class2 had one assignment on quantification in which they were required to read 26 pages of the appropriate Chapter in Baxter, Dubinsky, & Levin (1988) and do 17 problems on two-level quantifications. These problems represent the only actual work that the students did in what may be considered a minimal classroom treatment of quantification. They also form the only source of evaluation of learning in this class.

INSTRUMENTS

We consider Class1 and Class2 separately.

The Instruments for looking at what learning may have taken place in Class1 consisted of three occasions on which the students were given (in class and without warning) a set of questions on quantification to which they were expected to write the answers. We shall refer to these as SetA, SetB, and SetC and they are listed in Figures 4, 5, and 6. In addition there were two assignments to be handed in. They will be classed AssignA and AssignB. The questions are listed in Figures 7 and 8.

In most cases, after each of the in-class problems were completed, there was a discussion designed to give the students feedback on how they had done. The students understood that their performance on this work would be considered in determining their final grades. The in-class work was definitely individual. The students were asked to do individual work on the assignments, but there was no control on this.

AssignA was due on the same day that the students did SetC in class and AssignB was due one week later. There was no discussion after SetC and so it may be considered that AssignA, SetC and Assign B all took place (in that order) after the instructional treatment was completed.

It may be useful to give some details of the instruction that took place before SetA, between SetA and SetB, and between SetB and SetC so as to indicate what relations there may have been between the questions on which the students were evaluated compared to questions they had worked on previously. The majority of the students' study of quantification took place before the day on which they did the questions in SetA. There was preliminary work with declarations linked by connectors and/or depending on variables. Students were asked to translate back and forth between English and formal language (ISETL) and to write programs that would generate all possible truth values of the statement for each possible set of values for the variables. All of this was done in laboratory and in homework assignments. None of the problems considered up to this point had any relation with quantification nor any connection with any part of the statements subsequently used in the evaluation.

At the end of the preliminary work, the students were asked to work with (translate and negate) the following statement.

> `Every one of` $B_1, B_2, B_3, B_4, B_5, B_6$ `which is greater than 10 has its value less than the value of one of` B_1, B_2, B_3`.`

The students were expected to use conjunctions and disjunctions explicitly. It was considered that such an exercise would provide a transition to quantification. This idea follows a suggestion of D. Griese (Griese, 1981.)

After the preliminary period, there was a single lecture during which there was explanation and discussion of propositions and the meaning (in mathematics) of the negation of a proposition. The students were introduced briefly to examples of statements involving single-level and two-level quantifications, but little time was spent on them and there was no analysis. Again, the statements had no relation to the examples used later in the evaluation, except for the following which has some similarity with Question 1 in SetC (Figure 6).

For every city in New York, there is a city in Vermont such that if you were born within 100 miles of the city in Vermont, then you have visited the city in New York.

Next, there was a laboratory session in which the students did exercises analyzing ISETL character strings that expressed propositions, wrote code to generate truth tables for propositions, constructed sets of propositions on the computer, and constructed ISETL `func`s to represent proposition valued functions.

At this point the students were asked to complete an assignment that repeated and summarized the material covered up to this point and to read, on their own, 26 pages of text on quantification presented from the point of view discussed in this paper and in Dubinsky, Elterman, & Gong (1988).

There was a three week break between the end of the preliminary period and the day on which the students were given (without warning) the problems in SetA to do in class. During this interval, the course covered sets and their construction in ISETL and in mathematics. The only connection with logic during this period occurred when the students used propositions to construct subsets. This had nothing to do with quantification.

INSERT FIGURE 4 HERE

The students were given the four questions in SetA separately and they were not permitted to refer to their notes or to the textbook. After they finished each question, their papers were collected and there was a discussion of the question and its answer. The discussion did not take place after the last question because of lack of time.

Before the students were given the questions on SetB the following week, they were asked to review the text material on quantification and were given an assignment in which they could apply quantification to greatly simplify the solution to a rather tedious problem they had worked on previously. This problem concerned a chemical factory and various choices of the proportions of ingredients in a certain compound. The choices had to satisfy a long list of conditions which were fairly complex when expressed in first order propositional calculus. The students had to write long programs with many decision points to solve this problem. Now, with quantification, one can write an ISETL program to solve the problem in which the code is almost identical to the statement of the conditions and takes only 4 or 5 lines.

The questions in SetB were given, again without warning, one week after the questions in SetA.

INSERT FIGURE 5 HERE

There was another period of one week between SetB and SetC. During this time students had one laboratory session in which they constructed the ISETL `func`s referred to at the end of the section above on ISETL. These `func`s perform the appropriate quantifications on given sets and proposition valued functions. The essential point is that they can be composed (in the sense of composition of functions) and this construction on the computer corresponds to the mental construction of coordinating two single level quantifications to obtain a two-level quantification.

INSERT FIGURE 6 HERE

AssignA described in Figure 7 was due on the same day as the students did the questions on SetC so that it may be considered that this was a high point in the students' practice with quantification.

INSERT FIGURE 7 HERE

AssignB was due one week after ASsignA. In the interim, the class had moved on to other topics.

INSERT FIGURE 8 HERE

There was just one set of problems in Class2 and they are given in Figure 9.

INSERT FIGURE 9 HERE

RESULTS

The problems in SetA, SetB, and SetC were graded by the author. AssignA and the Class2 problems were graded by a student assistant and regraded by the author. AssignB was graded by the student assistant. In all cases, the grading consisted of assigning the result to one of three categories: correct, partially correct, and incorrect. The middle category was the most difficult to determine. In general a problem was given this designation if the response indicated that the student had some non-trivial understanding of the main issues involved in solving the problem, but not enough that one could consider the response to be essentially correct.

The results are presented in six tables, one for each problem set. In each table there are five columns. The first column designates the problem as listed in the previous section. The next three columns give the number of students whose responses were placed in the correct, partially correct and incorrect categories respectively. The last column computes an "average" score per student on the problem, based on 100 points for a correct response, 50 points for partially correct and 0 points for incorrect. The number of students is given in the caption.

INSERT TABLES 1,2,3,4,5,6 HERE

There are two striking features of the performance of Class1. The first that, in this author's opinion, they did very well. In 11 of the 21 problems, the average was better than 70%. In some cases, such as Problems 1a and 2 on SetA which are only single-level quantifications and Problem 1 on SetC which is a very straightforward two-level quantification,, it can be argued that these represent only the very beginnings in working with quantification. Also, Problem 2b on SetA was done immediately after the answer to the very similar Problem 1b was discussed, and Problem 1 on AssignB was due one week after the identical Problem 3 on SetC was done in class (although there was no discussion of its solution). A further point to remember is that, although the students were proctored while working on the in-class problems and the time was controlled (indeed, lack of time may partially explain the results on SetC, Problem 3) they were only asked to do individual work on AssignA and AssignB. There was no control on this and they had a week to work on the problems. Moreover, the atmosphere of the course was such that, up to this point, students had been strongly encouraged to cooperate and work together. It is entirely possible that some of the work on the assignments represented group efforts.

On the other hand, Problems 3a and 4b of SetA were quite new to the students and their performance on them as well as on Problem 3 of SetB was quite strong. The results on the latter two indicate that they were able to take apart a complicated statement of this kind and analyze it to some extent. Problems 1 and 3 of AssignA also seem to be difficult and the performance here in encouraging. In particular, Problem 1 of AssignA is very typical of reasoning often required in doing mathematics. Finally, regarding the repetition of the "flying fish" on SetC and AssignB, at least there was significant improvement the second time around and this held to the extent of being able to reason about the statement in Problem 2 of AssignB.

The second striking feature concerns the kind of errors that the students made. Of the seven problems on which the students did poorly (less than 60%), five of them involved an implication which was not part of the quantification. It is clear from the papers that difficulty with implications contributed to the low score. For example, on Problem 4a of SetA, all but one of the students were quite correct in negating the quantification part. Their errors were entirely the result of greater or lesser difficulties they had with negating an implication.

The situation with Class2, where much less time was devoted to quantification and almost all of it was independent student work, is much different and the student performance is somewhat more erratic. They did quite well on only 7 of the 16 problems they were given and very poorly on 6 of them. Although the problems were not the same, it is interesting to note that the worst score on a problem by Class1 was 47% while in Class 2 there were 3 problems on which the students did less than 30%.

The Class2 students did quite well on Problems 1, 2, and 3 which as translations to English are no doubt fairly easy. Their worst performance was on Problems 4, 5, and 6 which required translation into formal language (ISETL). It is possible to argue that this result is largely due to lack of practice with the syntax. Problems, 7, 8, and 9 require some real understanding of quantification and the overall performance on these three is not discouraging. Problems 10, 11, and 12 are not so easy and the overall performance is marginal. It should be noted, however, that the wording of problem 10 includes a "trap" intended to keep the students from applying textual translation and formal rules in order to negate. The results on the last four problems are very encouraging and would be even more so, if they had not done so poorly on Problem 16. It is possible to argue that the trouble here had something to do with working with evaluating a function at all points in an interval, and, in light of the previous three, perhaps was not entirely due to lack of understanding of quantification. It should be noted that although these last four problems required only a true, false, or can't tell answer, many of the students included reasons on their papers. Not only were these generally correct (when the answer was correct) but their variety suggested that the papers may have been largely the result of individual efforts.

CONCLUSIONS

It seems reasonable to conclude from these results that in going through the approach we have described, students can develop some understanding of quantification and the ability to work with it, even when the particular problems are difficult. Certainly the performance of Class1 was more than minimally acceptable. Even in Class2, although the overall performance was much weaker, some learning seems to have taken place. The difference between the two classes might be entirely due to the relatively small amount of time that Class2 spent on quantification. It does appear that even small increases in that time could lead to improvement in learning. More experience is required to determine optimal levels of effort (presumably somewhere between that of Class1 and Class2). Another point that requires further study is to determine whether students who go through this approach not only come to perform well, but also seem to construct the schemas that constitute our genetic decomposition of quantification as developed in Dubinsky, Elterman, & Gong (1988) and summarized here in the section on genetic decompositions above. This would further support our general theory.

If we are right in suggesting that this approach can be an effective and practical way of introducing students to the concept of quantification, then it should be used in courses that integrate this study with other areas of mathematics - where quantification is needed. We have made a start on this here and elsewhere with analysis (note that some of our examples concern the definition of continuity). Another direction would be to see if students have an easier time understanding concepts in subjects such as Abstract Algebra and Graph Theory where quantification is an essential part of the language used in definitions and statements of results.

Finally we mention two issues not directly connected with quantification that arose in our investigations and could bear further study. The first is the continued difficulty that students have with implications. Of course they can be (and are) trained to negate implications in isolation. Here, however, we have seen that when implication is not the direct object of consideration, but rather is placed in the context of another issue (quantification), then there is a sharp drop in students' ability to cope. Indeed, the improvement in performance on the "flying fish" statement - without any classroom discussion - seemed greater than any improvement in dealing with implications.

It may well be that a full study of implication, in the spirit of the study described this paper, may be worth doing.

The second issue concerns the difference between the performance of students in Class2 on Problems 4, 5, and 6 where they did very poorly and Problems 13, 14, and 15 on which they did quite well. The former group suggests that the students were unable to express the statements in formal language, while the latter indicates that, nevertheless, they had quite a good understanding of the statements. (We have already pointed out the inconsistency of this with the result on Problem 16 and suggested that this may have been due to difficulty with the function concept.) One possible conclusion is that understanding the meaning of a statement can develop without a corresponding ability to deal with the statement linguistically. Again, more study is required.

References

Ayres, T. Davis, G., Dubinsky, E., & Lewin, P. (1988). Computer experiences in learning composition of functions, *Journal for Research in Mathematics Education, 19*(3), 246-259.

Baxter, N., Dubinsky, E., & Levin, G. (1988). *Learning Discrete Mathematics with ISETL.* New York: Springer.

Breidenbach, D., Dubinsky, E., Hawks, J., & Nichols, D. (1992). Development of the Process Conception of Function, *Educational Studies in Mathematics 23,* 247-285.

Brown, J. S. & vanLehn, K. (1980). Repair Theory: A Generative Theory of Bugs in Procedural Skills, *Cognitive Science 4,* 379-426.

Clement, J., Lochhead, J., & Soloway E. (1980). Positive effects of computer programming on students' understanding of variables and equations, *Communications of the Association for Computing Machinery* 467-474.

Dautermann, J. (1992). *ISETL: A language for learning mathematics*, St. Paul: West Educational Publishing.

DiSessa, A., Knowledge in Pieces, Address to the Fiftieth Annual Symposium of the Jean Piaget Society, Philadelphia, June, 1985.

Douglas, R. G. (1986). Toward a Lean and Lively Calculus, Report of the Conference/Workshop to develop Curriculum and Teaching Methods, *Mathematical Association of America, Notes and Reports 6.*

Dubinsky, E. (1986). Teaching mathematical induction I. *The Journal of Mathematical Behavior, 5,* 305-317.

Dubinsky, E. (1989) Teaching mathematical induction II, *The Journal of Mathematical Behavior 8,* 285-304.

Dubinsky, E., Dautermann, J. Leron, U. & Zazkis (1994). On learning fundemental concepts of group theory, *Educational Studies in Mathematics, 27,* 267-305.

Dubinsky, E., Elterman, F. & Gong, C. (1988). The student's construction of quantification, *For the Learning of Mathematics, 8*(2), 44-51.

Dubinsky, E. & Leron, U. *Learning Abstract Algebra with ISETL*, New York: Springer-Verlag, 1994.

Dubinsky, E. & Lewin, P. (1986). Reflective abstraction and mathematics education: the genetic decomposition of induction and compactness, *The Journal of Mathematical Behavior, 5*, 55-92.

Dubinsky, E. & Schwingendorf, K. (1991). Constructing Calculus Concepts: Cooperation in a Computer Laboratory, in Carl Leinbach (ed.) *The Laboratory Approach to Teaching Calculus, Mathematical Association of America 20.*

Griese, D. (1981). *The Science of Programming* New York: Springer- Verlag.

Leron, U. & Dubinsky, E. (1995) An Abstract algebra Story, *American Mathematical Monthly, 102*(3) 247-272.

Schwartz, J. T., Dewar, R. B. K., Dubinsky, E. & Schonberg, E., (1986). *Programming with Sets,* New York: Springer.

Sfard, A., (1991). On the dual nature of mathematical conceptions: Reflections on processes and objects as different sides of the same coin, *Educational Studies in Mathematics, 22,* 1-36.

vanLehn, K., (1980). Bugs are not enough: empirical studies of bugs, impasses, and repairs in procedural skills, *The Journal of Mathematical Behavior, 3*(2), 3-71.

1. Consider the following statement

   Every third character in the string is a vowel.

   (a) Negate the statement by translating it into ISETL, performing a formal negation, and translating back.
   (b) What can you tell about the truth or falsity of the original statement if you know that the string is empty?

2. Consider the following statement

   There is a flower in the garden which is a rose.

   (a) Negate the statement by translating it into ISETL, performing a formal negation, and translating back.
   (b) What can you tell about the truth or falsity of the original statement if you know that there are no flowers in the garden?

3. Consider the following statement

   Every word in the sentence has the property that if its fifth letter is a consonant, then it has at least 8 letters.

   (a) Negate the statement by translating it into ISETL, performing a formal negation, and translating back.
   (b) What can you tell about the truth or falsity of the original statement if you know that every word in the sentence is less than 5 letters long?
   (c) What can you tell about the truth or falsity of the original statement if you know that every word in the sentence is less than 8 letters long?
   (d) What can you tell about the truth or falsity of the original statement if you know that every word in the sentence is at least 8 letters long?

4. Consider the following statement

```
forall P > 0 | (exists Q>0 | (forall x in [c-Q,c+Q] |
                (abs(x-c)<Q impl abs(F(x)-F(c))<P)
```

   (a) Negate the statement and express the negation in English
   (b) What do you know about the truth or falsity of the statement if `F` is a constant function?

Consider the following statement:

> There is a year in the 19th Century during which in Potsdam it snowed at least one day in every month.

1. Define appropriate structures and express the statement formally.
2. Express the negation in English.
3. Describe how a person or the computer would determine the truth or falsity of the statement.

1. Negate the following statement.

   For every city in Vermont, there is a city in New York which has the same name.

2. Negate the following statement.

   There is a positive number Q such that for every positive number P and for every $x \in [c - Q, c + Q]$, if $|x - c| < Q$ then $|F(z) - F(c)| < P$.

3. Negate the following statement.

   Amongst all the fish flying around the dining hall, there is one for which there is, in every Computer Science class, a Physics major who knows how much the fish weighs.

1. Explain what it would mean for the following statement to be false.

   There exists $\delta > 0$ such that for all x near c, $|x - c| \leq \delta \Longrightarrow |F(x) - F(c)| \leq \epsilon$.

2. Defining whatever sets, maps, funcs, and/or Boolean expressions that are necessary, express the following statement in ISETL.

   For every book in the library, there is a number of days (less than 1000) such that if the book is that number of days overdue, then the fine is $10.

3. Defining whatever sets, maps, funcs, and/or Boolean expressions that are necessary, express the following statement in ISETL.

   There is a book in the library such that for every staff member in the library, the weight of the book is more than 1/10th the weight of the staff member.

1. Consider the following statement.

   > Amongst all the fish flying around the dining hall, there is one for which there is, in every computer science class this semester, a Physics major who knows how much the fish weighs.

   Negate this statement by performing the following three steps.

   (a) Define appropriate sets and translate the statement into ISETL.

   (b) Negate the ISETL syntax and simplify.

   (c) Translate back into English.

2. Answer the following two questions about the statement (not its negation) in Problem 1.

   (a) What can you say about the truth value of the statement if you know that there are no computer science classes this semester?

   (b) What can you say about the truth value of the statement if you know that there are computer science classes this semester, but none of them has a physics major?

1. Translate the following statement into as simple and clear English as you can.

   ```
   forall s in {1..4} | exists t in {1..4} | (s*t) mod 5 = 1);
   ```

2. Translate the following statement into as simple and clear English as you can.

   $$\exists t \in \{1, \ldots, 4\} \ni \forall s \in \{1, \ldots, 4\},\ s \cdot t mod 5 = 1$$

3. Translate the following statement into as simple and clear English as you can.

   ```
   exists M in Mapset | (forall x in domain (M) | [x,x] in M)
   ```

   (Here `Mapset` is a set of smaps.)

4. Translate the following statement into ISETL syntax. If any auxiliary objects are required, make sure to specify them.

   There is a vowel in every word of the text.

5. Same as the previous problem for the following statement.

   The five polynomials have a common zero.

6. Same as the previous problem for the following statement.

   Every positive integer x in the set has a positive even divisor which is less than $x - 5$.

7. Explain why the two statements in Problems 1 and 2 are not equal.

8. Replace the proposition valued function in Problems 1 and 2 by a proposition valued function P such that the two statements will be equal.

9. Give three examples of pairs of sets S, T such that

   $$(\exists x \in S \ni \forall y \in T, P(x, y)) = (\forall y \in T,\ \exists x \in S \ni P(x, y))$$

10. Negate the following statement directly from the English.

    In all the classes that I have taught, there is one in which every student got an “A”.

11. Negate the following statement directly from the English.

    At least one of the six functions has a derivative which never vanishes.

12. Negate the following statement directly from the English.

    For a given $\epsilon > 0$, it is the case that for every $\delta > 0$ there is an x in the domain of the function such that $|x| < \delta$ but $|F(x)| > \epsilon$.

13. What do you know about the truth value of the statement in Problem 6 if you know that every integer in the set is odd?

14. What do you know about the truth value of the statement in Problem 6 if you know that the set is $\{1, 2, 3, 4\}$?

15. What do you know about the truth value of the statement in Problem 12 if you know that the domain of the function is the set of real numbers, $F(x) = 0$ for each element of its domain, and $\epsilon = 0.001$?

16. What do you know about the truth value of the statement in Problem 12 if you know that the domain of the function is the set of real numbers, $F(x) = x^2$ for each element of its domain, and $\epsilon > 0$?

Problem	#Correct	#Partial	#Incorrect	Average
1a	14	1	3	81
1b	7	7	4	58
2a	17	0	1	94
2b	18	0	0	100
3a	14	4	0	89
3b	11	0	7	61
3c	3	14	1	56
3d	10	0	8	56
4a	8	4	6	56
4b	14	4	0	89

Problem	#Correct	#Partial	#Incorrect	Average
1	11	3	4	69
2	10	3	5	64
3	12	3	3	75

Problem	#Correct	#Partial	#Incorrect	Average
1	18	0	1	95
2	6	9	4	55
3	10	2	7	58

Problem	#Correct	#Partial	#Incorrect	Average
1	10	9	0	76
2	5	8	6	47
3	13	6	0	84

Problem	#Correct	#Partial	#Incorrect	Average
1	10	7	2	71
2	14	3	2	82

Problem	#Correct	#Partial	#Incorrect	Average
1	16	0	1	94
2	14	2	1	88
3	10	6	1	76
4	5	8	4	53
5	1	2	14	12
6	3	4	10	29
7	16	1	0	97
8	6	0	11	35
9	7	2	8	47
10	7	0	10	41
11	1	13	3	44
12	9	4	4	65
13	14	1	2	85
14	17	0	0	100
15	14	0	3	82
16	2	1	14	15

Figure 1: A paradigm for research and development

Figure 2: Constructions for Mathematical Knowledge

Figure 3: The genetic decomposition of quantification

Figure 4: SetA. In-class questions

Figure 5: SetB. In-class questions

Figure 6: SetC. In-class questions

Figure 7: AssignA. Questions to be handed in.

Figure 8: AssignB. Questions to be handed in.

Figure 9: Class2 Problems. Questions to be handed in.

Table 1. Results on SetA. 18 students.

Table 2. Results on SetB. 18 students.

Table 3. Results on SetC. 19 students.

Table 4. Results on AssignA. 19 students.

Table 5. Results on AssignB. 19 students.

Table 6. Results on Class2 Problems. 17 students.

Captions for Figures and Tables