

# 5/ Mathematical Microworlds and Intelligent Computer-Assisted Instruction

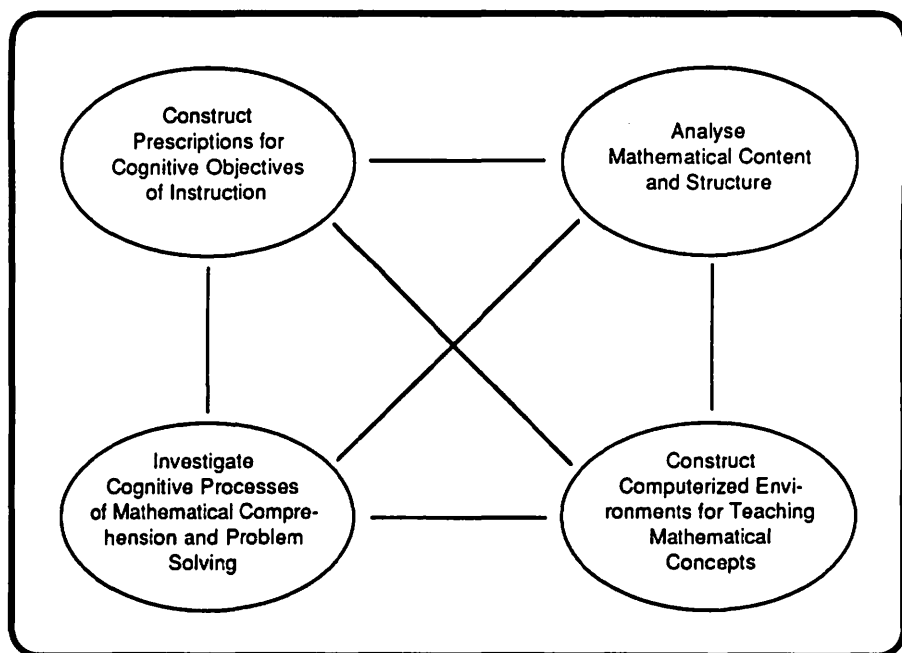
PATRICK W. THOMPSON

**M**y work with microworlds has been to design computerized environments that allow two foci: conceptual development and mathematical problem solving. The theory behind that approach has been elaborated elsewhere (Thompson, 1985a). In this chapter I will only briefly touch upon theoretical motivations, devoting the majority of the discussion to what I mean by a mathematical microworld, how one works, and to issues of designing microworld environments.

The discussion in this chapter is framed by the research program in which I have been engaged over the past four years. Figure 5.1 outlines the principal components of that program. It shows that remarks given here about the design of software to be used in mathematics teaching and learning are not given in isolation. Rather, they are informed by results, conceptions, and metaphors from investigations of cognitive processes of mathematical comprehension and problem solving, prescriptions for cognitive objectives of instruction, and analyses of mathematical content (Dreyfus & Thompson, 1985; Thompson, 1985a). Each of these, in turn, is informed by knowledge gained through research and development of software for teaching and learning mathematics (Thompson, 1985b, 1985c; Thompson & Dreyfus, in press). I do not mean to say that one must accept the research program outlined in Fig. 5.1 to design mathematical microworlds. Rather, I mean only to say that issues of design are at heart theoretical, and as an aid to communication it helps to make explicit one's theoretical perspective from the outset.

---

Revised version of a paper presented at the Invitational Workshop on Intelligent Computer-Assisted Instruction and Personal Computers, San Diego, CA, February 1985. The author wishes to thank Matthew Lewis for his helpful and insightful reactions to a draft of this chapter.



**Figure 5.1** The four components of a research program on understanding and improving mathematics teaching and learning.

Before characterizing mathematical microworlds, it will be instructive to discuss the educational problem they address. It is this: far too many students do not understand that mathematical symbol systems are fundamentally representational; they are contrived by individuals or communities to provide economic ways to think about complex ideas. Instead, students typically perceive that mathematics is concerned primarily with making marks on paper, and that the aim of mathematics instruction is to teach them how to make correct marks at correct positions on the paper. The goal that mathematical microworlds serve is to provide students with opportunities to create mental models that reflect the structure and composition of the formal systems our culture has deemed important to learn. Once in place, a mental model serves as a stepping-stone for students in their reconstruction of their qualitative knowledge into a formal system. In short, a mathematical microworld provides a dynamic semantics for a formal system.

I should also make clear an underlying philosophy of mathematical microworlds. It is that they act as objective systems, in the sense of physical systems studied by scientists. A consequence of this philosophy is that microworlds do

not teach; they do not provide instruction. The reason is that when a scientist performs an experiment upon a physical system, the system does not comment on the quality of the experiment. Rather, it merely behaves. A scientist attempts to hypothesize the principles by which a system operates by observing the system's behavior under experimental conditions. The scientist does not get advice from the system.\*

This chapter has five sections. The first discusses general characteristics of microworlds and gives an in-depth example of one—a microworld for isometric transformations of the plane. The second and third sections discuss issues and principles in designing mathematical microworlds. The fourth section discusses shortcomings that have been found and enhancements that make use of artificial intelligence. The fifth section discusses possibilities for intelligent microworlds in schools.

## Mathematical Microworlds

I will use “mathematical microworld” to mean a system composed of objects, relationships among objects, and operations that transform objects and relationships. This characterization is meant to capture the idea of a mathematical system as constructed from primitive terms and propositions, where the full system initially exists only potentially but includes features that allow students to expand that potential.

It is unfortunate that the generic term “microworld” has been used so many different ways. Papert's (1980) use of “microworld” includes a procedure to create spiral polygons and a program to simulate Newton's third principle of thermodynamics (diSessa, 1982). Lawler (1982) uses “microworld” to describe mental constructs that are often called schemata in cognitive psychology; in Lawler (1984), “microworld” describes any program that provides “inherently interesting phenomena.” Rather than attempt to create a new name, I use the qualifier “mathematical” to distinguish the systems described here from what have been called microworlds by others.

In practice, a mathematical microworld incorporates a graphical display that depicts a visualization of the microworld's initial objects. The display in conjunction with operations upon the microworld's objects constitutes a model of the concept or concepts being proposed to the students. In a very real sense, the microworld embodies the structure of the concept. The students' task is to internalize that structure and make it their own.

The last criterion listed—that the system contain operations by which new objects can be made—is essential. This is what makes a mathematical micro-

\* My favorite analogy is one concocted by a graduate student: If you want to become an expert archer, you practice archery. Suppose you shoot an arrow at a target and the arrow is wide to the left. The target doesn't yell “Hey, more to the right.” Rather, you judge the efficacy of your procedures by their results and modify them accordingly.

world mathematical. In a sense, it forces the system to be a mathematical programming language. However, the result of programming is not a "program." The result is a new mathematical object. That is, a mathematical microworld constitutes the core of an intuitive axiom system, where students can define new objects and operations and interactively investigate their properties.

### Examples of Mathematical Microworlds

Fuller, Prusinkiewicz, and Rambally (1985) have developed a microworld called LEGO for creating geometric illustrations. Though their intention was that LEGO be used only by a teacher for demonstrations, it contains all the ingredients listed above for being a mathematical microworld. Each command in LEGO is a function that happens to create a geometric object. New functions (objects) are defined by composing existing ones and giving the composition a name. New operations on objects (such as intersection and union) are also defined by composition. Greenleaf (1984) reports a similar system, called EUCLID. EUCLID not only gives the capability of composing functions, but also includes a programming language. Neither LEGO nor EUCLID runs on microcomputers. LEGO runs on a VAX-730, under Franz LISP, while EUCLID requires specially developed hardware.

Rocky's Boots, by the Learning Company, is almost a mathematical microworld for Boolean algebra. One constructs "selection mechanisms" (logic circuits) that cause Rocky the Raccoon to kick objects that possess certain attributes. One feature it lacks is that it does not contain a formalism that allows a student to represent the effect of one's mechanisms. The need to include a formalism in a mathematical microworld is addressed in a later section on design issues.

Other examples of mathematical microworlds are the Geometric Supposer by Judah Schwartz, published by Sunburst, and a Marble Bag microworld being developed by Wally Fuerzeig (1986) under the sponsorship of the National Science Foundation.

MOTIONS is a microworld for isometric transformations of the plane (mappings of the plane that preserve distances between points) that I developed several years ago to run on Apple II's and Commodore 64's. It is designed to be used at a number of grade levels, ranging from junior high school to university teacher education courses. At the teacher education level, it typically is used for a total of 10 weeks (sometimes spanning two courses).

MOTIONS was designed with a set of cognitive goals in mind. These, briefly, are that students understand motion geometry as a *mathematical system*, and that they develop concepts of multivariate mappings, invariances under mappings, and of composition as an operation upon mappings. Thompson (1985a) gives a detailed treatment of MOTION's cognitive goals.

The MOTIONS microworld shows a pennant within a Cartesian coordinate system. The pennant has three properties: a position, a heading, and an orienta-

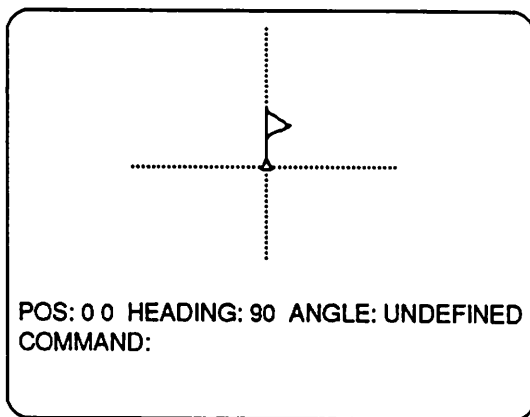
tion. Each property is defined by convention. The pennant's *position* is the position of the bottom tip of its staff. The pennant's *heading* is the direction of the ray emanating from its position and passing through its staff, as measured from the right horizontal. Its *orientation* is the direction one would turn (right or left) were one traversing the pennant from the tip of its base.

The pennant's position and heading are given in a status line, along with the measure of the angle formed by the horizontal  $x$ -axis, the origin, and the pennant's position (Fig. 5.2). The pennant's orientation is not listed since it is visually apparent.

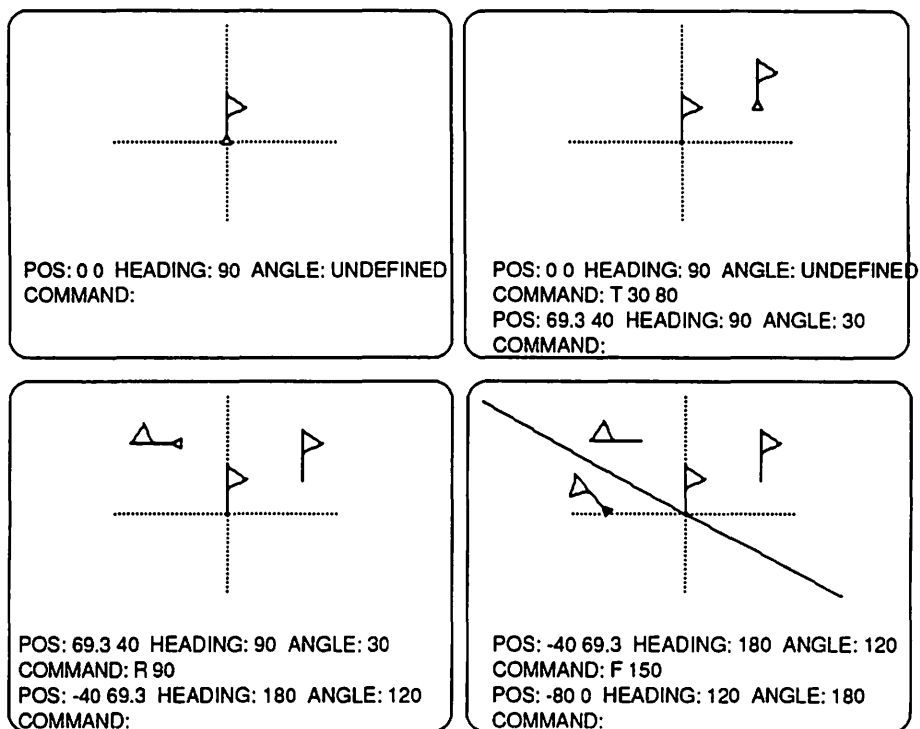
The operations allowed by MOTIONS are translations, rotations, and reflections (flips). The format for executing each is given below.

| COMMAND                   | MEANING   |
|---------------------------|---|
| T <i>heading distance</i> | Translate the plane in a direction of <i>heading</i> through a distance of <i>distance</i> units.         |
| R <i>degrees</i>          | Rotate the plane about the origin through an angle of measure <i>degrees</i> .                            |
| F <i>heading</i>          | Flip (reflect) the plane through the line passing through the origin in the direction of <i>heading</i> . |

Figure 5.3 shows a succession of displays as commands are entered. Each command operates upon the last-drawn pennant. Hence, effects upon the



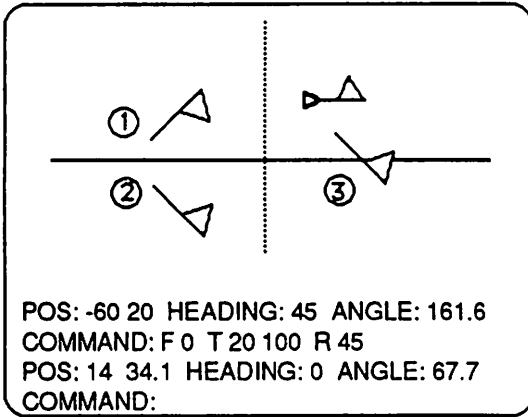
**Figure 5.2** The initial display of MOTIONS. POS gives the pennant's position. HEADING gives its direction as measured counter-clockwise in degrees from the right-horizontal. ANGLE gives the measure in degrees of the angle formed by the pennant's position, the origin, and the positive  $x$ -axis.



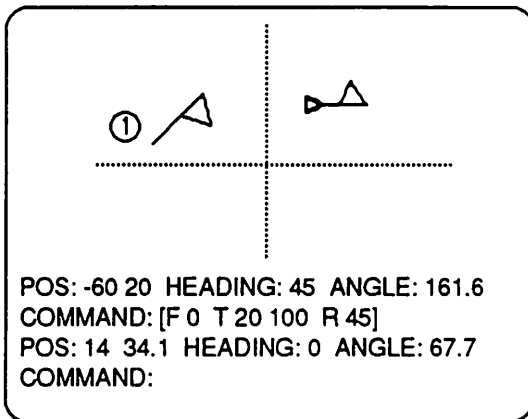
**Figure 5.3** Successive displays generated by entering commands to MOTIONS. From top-left to right-bottom: (1) The initial display. (2) T 30 80 moved the plane containing the current pennant 80 units in a heading of  $30^\circ$ . (3) R 90 rotated the plane containing the current pennant  $90^\circ$  about the origin. (4) F 150 flipped (reflected) the plane through the line passing through the origin at a heading of  $150^\circ$ .

plane are cumulative. The end state of a pennant under one command is the beginning state of the pennant for the next.

One composes transformations in several ways. The most intuitive is simply to enter them all on the same line. Each command is executed as if entered separately, and each intermediate pennant is drawn (Fig. 5.4). A second way, which more closely represents the mathematical idea of composition, is to enter the commands surrounded by brackets. When a set of commands is entered in brackets, the *composition* is performed (Fig. 5.5). A third way to compose transformations is to define a new transformation that is made of a



**Figure 5.4** Entering commands on one line causes each command to be executed in turn. F 0 executed while the pennant was in state 1, causing it to take on state 2. T 20 100 executed while the pennant was in state 2, causing it to take on state 3. R 45 caused the pennant to change from state 3 to its ending state.

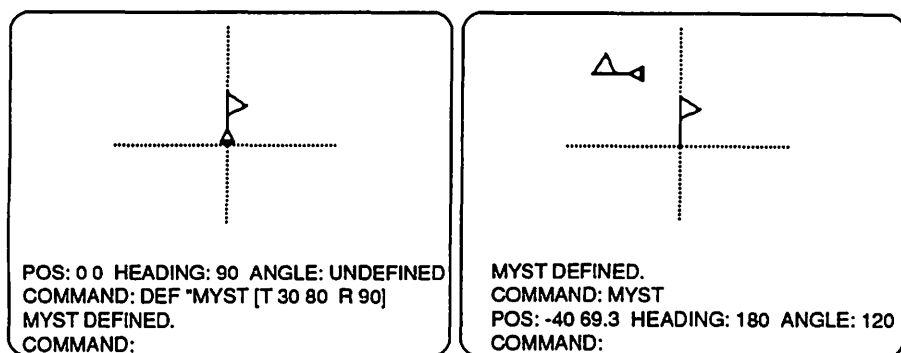


**Figure 5.5** The display after the composition [F 0 T 20 100 R 45] is entered. The pennant began in state 1. The composition caused the pennant to take on its final state. Compare the initial and final states here to those in Fig. 5.4.

composition of existing ones. To define a new transformation, one enters DEF (for DEFine), the new transformation's name, and the composition that defines it (Fig. 5.6). If one wishes a new transformation to be defined so as to take variable arguments, then the composition defining it is preceded by the list of variable names used in the composition. Variable names are preceded by colons in the defining part of the composition (Fig. 5.7). The list of variable names preceding a composition has the purpose of distinguishing names that stand for arguments from names that are automatically supplied by the microworld (e.g., XCOR and YCOR for the  $x$ - and  $y$ -coordinates of the pennant's position, and HEADING for its heading).

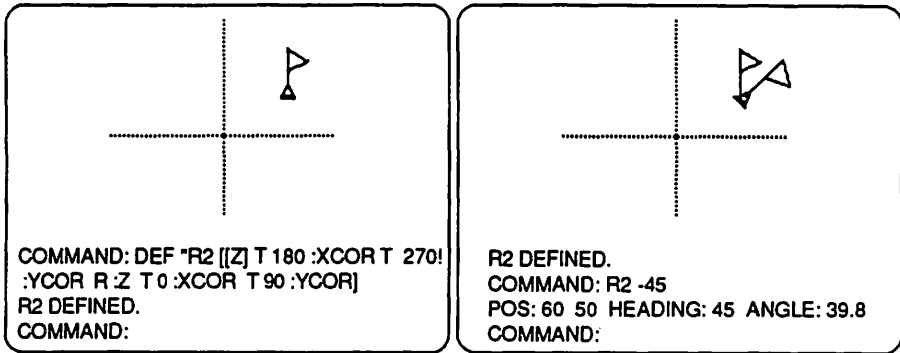
The last feature of MOTIONS to be discussed is not a feature of the program. Rather, it is the problem sets that accompany the program. Without problems to structure students' investigations, they would be limited for the most part to undirected exploration of the program per se, and would most likely avoid coming to grips with the intricacies of the subject matter itself.

The problems are divided into three groups, each group emphasizing different levels of abstraction and generalization. The first group assumes that students will consider the pennant to be the object operated upon and focuses their attention on invariances under various transformations (heading and orientation are invariant under translation, while position varies; orientation and center of rotation are invariant under rotation, while position and heading vary; line of reflection is invariant under reflection, while position, heading, and



**Figure 5.6** Defining a composition by name. The name MYST means to do the composition of T 30 80 and R 90. The display on the right shows the effect of entering MYST. The composition is, in fact, a rotation about the point  $(-14.65, 54.65)$  through an angle of  $90^\circ$ .





**Figure 5.7** Defining a motion that takes variable arguments. Variable names in the defining composition are preceded by a colon to distinguish them from names of operations. The name Z is in a list to distinguish it from names supplied by MOTIONS (XCOR and YCOR). The effect of R2 is to rotate the pennant :Z degrees about its current position, as shown in the display on the right. The exclamation mark (!) indicates that the user's command is continued on the next screen line.

orientation vary). The first group also emphasizes that students should become skilled at visualizing the effects of the transformations.

The second group of problems focuses on transformations of the plane as multivariate mappings. Students are asked to develop generalizations about the effect upon each of the three properties of the pennant (position, heading, and orientation) under the various transformations. An example of this type of problem:

While doing his homework, John entered F 70 while the pennant was at position (30,20) with orientation RIGHT. By how much did the pennant's initial heading change?\*

The third group of problems emphasizes operations *upon* transformations. The primary operation of concern is composition. The problems address the intricacies of composition in a number of ways. One is negation, as in:

\* Initial heading is unstated, so students must determine the general relationship between initial and resulting heading of a pennant under a reflection.

Mary entered T 30 80 R 70. Tell her *one* transformation that will take the pennant back to its original state.

and:

Frank entered F 70 F 90 when he intended to enter F 70 R 90. What *one* transformation could Frank use to put the pennant in the state it would have attained had he entered his command correctly?

Another approach to composition is seen in problems that ask students to determine the *net effect* of entering two or more transformations at once. "Net effect" is defined as the single transformation that one could enter to achieve the same result as that of the combined transformations. Still another approach asks students to form "addition tables" of transformations that are closed under composition. This takes them into the realm of algebras of transformations.

### Differences between Mathematical Microworlds and CAI

There are five primary differences between mathematical microworlds and traditional CAI. The first centers around the idea of information transmittal. In the past, most CAI has been designed with the idea that some body of information is to be transmitted from the program to the student, or that the program will guide the student to a point of "discovery." A CAI program that is tutorial in nature poses questions, gives examples and illustrations, and generally "talks" to students. The idea of information transmittal has no relevance to the design of a microworld, except to the extent that its design makes it easier or harder for students to *infer* information.

The second difference between tutorial CAI and mathematical microworlds is that the focus of a mathematical microworld is on the construction of meaning and relationships, while CAI tends to focus on facts and skills. Kearsley (1985) pointed out that my comments here do not apply to simulations, and I agree. However, I would not want to classify simulations and mathematical microworlds together. A simulation simulates *something*. But what does a mathematical microworld simulate? At most, it "simulates" an axiom system, in the sense that it is an instantiation of a formal system and embodies its structure. I would think a microworld to be more of a model of a formal system than a simulation of a prototype (cf. Kleene, 1952; Kneebone, 1963).\*

A microworld's focus upon relationships leads directly to an emphasis on transformations, at least within mathematics. Two mathematical objects are related when there is some way to transform one or more attributes of one into attributes of the other.

\* My remarks here apply only to *mathematical* microworlds. Microworlds as described by Papert (1980) and by Lawler (1984) typically do have a "real world" sense to them, and are more like simulations.

The third difference between CAI and mathematical microworlds is in the layered approach to structure in microworlds. To learn mathematics, students must construct mathematical objects, construct relationships among them, and then take those relationships as new objects to be operated upon. Mathematical microworlds are designed to facilitate that process, whereas mathematical structure in CAI has typically been ignored.

The fourth difference is the assumed role of the teacher. My impression of CAI, both traditional and intelligent, is that designers envision their programs as being temporary substitutes for the teacher—that a teacher would send students off to use the program with the confidence that it will do a good job in his or her stead. My vision of a teacher using a microworld is as a tool to provoke classroom discussions. A teacher might have one computer with a large-screen monitor at the front of the classroom and use it to challenge the class to think about what will happen when a particular command is entered and about what is generally true under various circumstances. That is, a microworld's design is predicated upon the assumption that a teacher will be integrally involved in the students' explorations. A teacher will certainly have students work individually with microworlds, but will do so only with the aim that they investigate some particular feature of the mathematics embodied within it.

Fifth, in traditional CAI, the program and the curriculum are inextricably intertwined. In a mathematical microworld there is a strong distinction between the curriculum and the environment upon which students act. The software presents the environment and metaphor and incorporates the mathematical structure within a model. Instructions, explanations, and questions for students are given in printed materials. This allows teachers to adapt the curriculum to the purposes of instruction as well as to the levels of the students. Continued use of a model under these circumstances is apt to contribute to the integration, in students' minds, of the various levels at which the concept can be examined.\*

## Issues of Design

In this section I will recount the dialectic between designing MOTIONS and assessing its impact on preservice elementary teachers who were using it. Throughout the design and evaluation processes, the aim was to uncover fundamental features of their cognitions that were retarding their understanding of fundamental concepts, and to modify the microworld to address those difficulties directly. After discussing the evolution of MOTIONS, I will summarize principles of designing mathematical microworlds that have evolved from the implementation of it and five others.†

\* For example, one microworld (for teaching and learning integers) has been used in grade 1 to teach addition, in grade 6 to teach negation, and in college in teaching commutative groups.

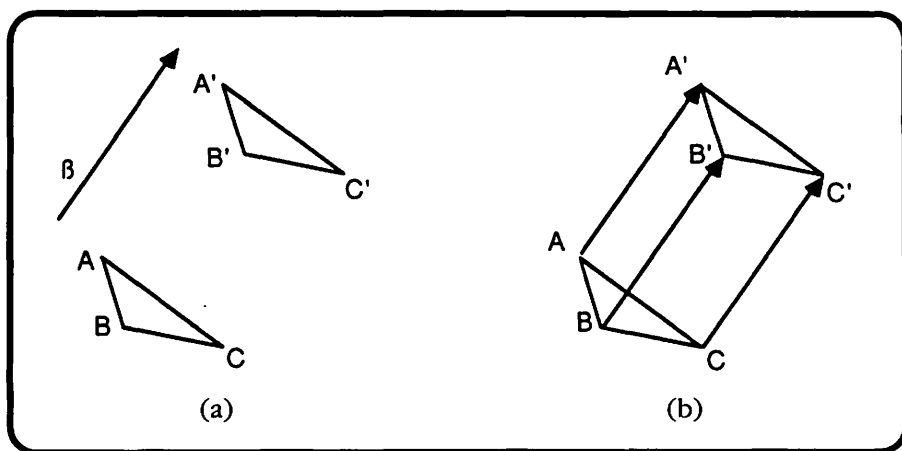
† The five others cover integers and introductory algebra, probability, number theory, equivalence relations, and elementary arithmetic.

## The Pedagogical Problem

The program was first conceived as an alternative to teaching transformation geometry through straightedge and compass constructions. In that approach, the aim was to have students develop procedures for various constructions (e.g., copying an angle, bisecting a line segment, etc.) and apply those procedures to problems of mapping points in the plane under isometric (length-preserving) transformations. That approach was singularly unsuccessful, for two reasons. First, students would become absorbed in a construction's details and lose sight of the mathematics that they were supposed to learn. Second, they considered each application of a procedure to be a unique transformation, even if it was applied as a subprocedure. For example: they were asked to find the image of triangle  $ABC$  under the translation defined by directed line segment  $\beta$  (Fig. 5.8a). They would construct the images of  $A$ ,  $B$ , and  $C$  (by copying angles and segments), connect them, and be done. However, in their thinking they performed three translations: one that took  $A$  to  $A'$ , one that took  $B$  to  $B'$ , and one that took  $C$  to  $C'$  (Fig. 5.8b). They did not understand that the entire set of points in the plane was mapped by the translation and that we merely located the images of three of them. This might not appear to be a serious problem. However, when we began to cover compositions of transformations they were bewildered by my asking them to find the image of a triangle, for example, under the composition of two translations when in their view the problem involved nine translations: six to do two translations of a triangle and three to link the original points with their final images. Moreover, they could not understand how one could equate the entire process of translating a triangle with a single translation, since (in their view) any translation moves only a single point.

## The Design of the Interface

The initial idea behind MOTIONS was to remove the drudgery of straightedge and compass constructions so that students could focus upon the mathematics of the transformations. The first issue to address was the design of the screen and the manner in which students would interact with the program. The design of the screen was relatively easy, as transformation geometry is commonly taught analytically, where the transformations are represented by matrices and points of the plane are represented as vectors. Thus, the screen needed to show a Cartesian coordinate system. The decision to have a flag as an object was made for two reasons: first, it is much easier to understand a transformation's effect upon the plane by observing preimage-image correspondences between sets of points than by observing the effect upon individual points. Second, whatever figure was to be displayed needed to be such that changes in orientation are apparent.



**Figure 5.8** (a) Translate the plane in the direction and distance defined by the directed line segment  $\beta$ . (b) A student's conception of the translation depicted in (a): translate A to A' (one translation); translate B to B' (two translations); translate C to C' (three translations).

The decision about how to have students interact with the program was also relatively easy to make. Since the aim was to teach transformation geometry as a mathematical system, and since the objects in that system are functions, the program needed to allow students to type common representations of those functions in order to “see” their meaning.

**Translations** To do a translation they would enter T, along with some measure of the distance and direction in which to translate. An alternative would have been to define T as taking two arguments that denote horizontal and vertical displacement. The two definitions of translation are equivalent. Since the former definition is easier to visualize (as a vector connecting initial and final position), I settled on the former and included activities and problems that focused on having students uncover the equivalence of the two representations.

**Rotations** To do a rotation they would enter R, along with the measure of the angle about which to rotate. Since the center of a rotation could also vary, it would have been sensible to have a rotation's center be variable. However, I chose to let students investigate how one *could* do an arbitrary rotation by composing existing transformations (as in Fig. 5.7) instead of providing them with a command that would trivialize the problem.

**Reflections** To do a reflection (flip) they would enter F (since R is already used to denote rotation) along with some defining characteristic(s) of the line through which the flip was to be performed. Again, I chose conservatively. The default class of lines were those through the origin, and their defining characteristic was their heading (direction). This was decided in anticipation of providing students with the problem of how to define an arbitrary flip.

### Instructional Design

It became apparent early on that the preservice teachers who were using MOTIONS would need guidance in their investigations if they were to cover anything resembling a curriculum. I decided to provide that guidance through the form of a handout containing nothing but an introduction to the program (how to start it, the commands, etc.) and a list of questions. The students did not like that approach. Their major complaints were (1) that they would work on a question and not know whether or not they had answered it “correctly” and (2) that they did not understand what they were supposed to learn. I rephrase their complaints this way: their first complaint arose because they did not understand that the aim of a question was that they “play” with a problem—varying parameters, varying conditions, and so on. It was not that the questions were ill-defined. They were well-defined. Rather, the questions were unlike those they were used to. The second complaint, about not knowing what they were supposed to learn, again arose because of the novelty of the approach. In their previous experiences in math courses they were told what they were supposed to learn—how to perform some procedure. They had little conception of solving a problem with the aim of forming a generalization from their solution methods.

### Intricacies of the Content and Students' Difficulties with MOTIONS

A number of small matters (in my initially naive view) about the design of MOTIONS became significant obstacles to students' understanding of the content. Aside from making the program more effective, addressing these issues opened a host of research questions on mathematical thinking. I became aware of them largely as a result of my practice of listening to students talk among themselves while working on MOTIONS problems in the computer laboratory.

**Reference Systems** A pennant exists within three reference systems—Cartesian, polar, and clock (orientation). It is important to understand that a transformation affects the pennant's properties relative to all three. Students had a horrible time coordinating the three, and had particular difficulty with the idea of a system of headings that “moved” with the pennant. For example, when thinking of translating the plane when a pennant was off the origin many would imagine the heading from the origin instead of from the pennant's position, and

would become hopelessly confused about how such a thing could be accomplished.

My awareness of this difficulty resulted in the addition of DOT and LINE commands so that students could experiment with position and heading independently of each other. DOT takes two inputs (an  $x$ - and a  $y$ -coordinate) and places a dot at that position. LINE takes three inputs (two coordinates and a heading) and draws a line through the given point with the given heading. Then I added activities to the handout that focused on the independence of position and heading.

**Command Formats** When first writing MOTIONS, it seemed natural to structure the command for performing a translation so that the user specified first a distance and second a direction. The image I relate to this format is that you first delineate a set of possible image points (the circle of radius *distance* from the pennant's position) and then restrict those to a single point—the intersection of the circle and the line having direction *heading* and passing through the pennant's position. However, students could not conjure up this image. They insisted that they needed to know the direction first and the distance second in order to imagine a translation. Apparently, when the distance was given first they would wait for the direction, switch the arguments, and then consider what translation that command represented. They found switching arguments bothersome and distracting. After changing the command format for translations I have yet to hear a similar complaint.

**Specifying an Initial State** Occasionally, one wishes to have a pennant start from a particular initial state. It might be that the pennant is too close to the screen's edge to see the result of the next command, or that one wishes to perform two motions from the same starting state. The command originally included in MOTIONS to put the pennant in a given state was GOTO. GOTO took a position, a heading, and an orientation as inputs, and placed a pennant in that state.

The problem with GOTO was that students thought of it as a transformation—on a par with T, R, and F. Apparently, the common-sense meaning of *go to* made them think of the pennant going from “here” to “there.” It appeared to them as a state-change operator.\* I changed GOTO to START.AT, and the problem disappeared.

**Comparing Transformations** It is essential that students understand that a transformation is applied to the entire plane, and that we merely “highlight” its

\* The reason that GOTO cannot be a state-change operator is that it acts only upon the state of a pennant, and not upon the entire plane. GOTO 20 30 40 RIGHT (go to the state of position (20,30), heading 40, and orientation RIGHT) would affect the plane one way were a pennant at the origin and a different way were a pennant at position (50,70).

effect on a particular subset of it. For example, T 30 80 has the same effect regardless of the initial state of the pennant. However, students tended to think that T 30 80 applied to the plane with the pennant in two different states resulted in two different motions, since they “looked” different. The solution to this problem was not another feature added to the program. Rather, it was a new set of activities that focused upon the independence of state and effect. Or, put another way, the activities focused upon invariant relationships between initial and final states under classes of transformations.

**The Concept of Net Effect** The original treatment of composition consisted of having students enter two or more commands on a line. Each motion was performed separately; the composition was the motion that related the pennant’s initial and final states (see Fig. 5.4).

The approach was logically sound. All one must do is ignore intermediate states of the pennant. However, few students could ignore the intermediate pennants. The presence of intermediate pennants “dominated” their conception of a command-set’s net result. An analogy would be someone’s absolute insistence that a trip from San Diego to New York via Denver and Chicago cannot be equated with a trip from San Diego directly to New York. Students felt that the two are different transformations since they involve different itineraries. This conception of equivalence is acceptable as common sense, but it completely blocks a student’s access to the mathematical idea of composition.\*

The solution to the problem of representing composition within a micro-world was to allow the student to compose transformations in two ways. The first, as already described, is by entering a sequence of motions and having MOTIONS carry them out individually. The second is to allow students to group a sequence of motions within brackets—make a unit of the sequence, so to speak. Whenever a sequence of commands is grouped in brackets, only the composition is performed (see Fig. 5.5). Thus a student can examine a composition at two levels: intuitively, as the consecutive execution of individual commands where the pennant’s end state under one motion is the beginning state for the next, and more formally, where only the initial and final states of the pennant are shown. The idea of composition did not become “easy” as a result of this change, but it did become accessible.

**Composition as a Mathematical Operation** One of the most difficult ideas for students is the idea that some object can be represented in a number of different ways. In the context of MOTIONS, this difficulty first appeared in problems requiring that students understand that a translation can be both that

\* An example of a student who understands that they are the same in terms of their composition is one who would say that a direct flight from New York to San Diego negates either trip, so they must be equivalent in their net effect.



translation and a composition of others. For example,  $T\ 0\ 5$  is also represented by the composition  $[T\ 45\ 3\ T\ -45\ 4]$ . Students fall victim to the same misconception about composing functions that young children have about computing sums, which is that a sum is the answer you get from adding. It is not a number satisfying a logical relationship with two other numbers. The parallel with MOTIONS is in problems such as "Mary entered two translations. The first was  $T\ 45\ 3$ . The net effect was  $T\ 0\ 5$ . What was the second motion?" Unless students understand that  $T\ 45\ 3$  bears a logical relationship to  $T\ 0\ 5$  they will not think to "subtract" the effect of  $T\ 45\ 3$  from  $T\ 0\ 5$  (by entering  $[T\ 0\ 5\ T\ 45\ -3]$ ) to see what translation remains.\* For this difficulty, I did not put anything into the microworld. Rather, I forced the issue by having students work problems that emphasize logical relationships among compositions.

**Defining New Transformations** It was surprising to find that students did not think of compositions denoted by the use of brackets (e.g.,  $[R\ 70\ F\ 10]$ ) as being on a par with the primitive motions denoted by  $T$ ,  $R$ , and  $F$ . Students thought of them much as many LISP students think of lambda-expressions: as "temporary" functions having no ontological status. In the thinking of students of transformation geometry, if a motion does not have a name, then it is not a bona fide transformation and cannot be equated to one that does have a name. To counter this misconception I added a feature whereby they could name a composition, and moreover, could specify variable parameters (whence the DEF command; see Figs. 5.6 and 5.7).

## Principles of Designing Microworlds

Several design principles have emerged from the continuing use and evaluation of MOTIONS and from the implementation of several other microworlds. These are discussed below.

### Orientation toward Functions

One of the first requirements in designing a mathematical microworld is to describe the system one aims at modeling. In mathematics, a system *is* a system because it is closed under some set of transformations and operations, or more generally, functions. Idiosyncracies of any system are a result of the functions that define it, and are independent of the objects upon which the functions are evaluated.† Thus, since mathematical microworlds are meant to focus upon

\* The translation  $T\ 45\ -3$  is the inverse of  $T\ 45\ 3$ .

† Changing the objects produces a system that is isomorphic to the original system (*iso*—"same"; *morphic*—"structure").

mathematical systems, they automatically have a bias toward being function oriented. This means that a user commands a microworld to do something, and the "something" it does has an effect on the state of the system.

### Command Formalism

Since the aim of a mathematical microworld is to provide students an entry into a mathematical system, and since mathematical microworlds are by nature function oriented, they are also command oriented. Two issues must be addressed in deciding upon the formalism students will use to interact with the microworld: the presentation of commands and the format in which students will use them.

**Presentation** One could present a list of commands in a menu and have students choose from the list. This is disadvantageous for a number of reasons. First, it is distracting, since the menu must disturb the display, and it is the current state of the display that will be affected by the student's next command. Second, choosing from a written list is antithetical to what one does when *doing* mathematics. I have found that menus detract from students establishing a correspondence between a formalism and its semantics. It is much more effective simply to give students a written glossary of commands as part of the printed material that contains problems and activities. Third, when students enter individual commands, their attention is focused on *that* command and its effect on the system's state. Thus, they have a much more explicit set of experiences from which to generalize (e.g.,  $F\ x$  does *this*) than when choosing commands and parameters from a menu.

**Command Format** The command format must be concise, and must at least resemble conventional mathematical formalism. In MOTIONS,  $T\ x\ y$  corresponds to the mathematically conventional  $T_{x,y}$ ;  $R\ x$  corresponds to  $R_x$ . In a microworld for probability,  $P[A\ B\ C]$  corresponds to the conventional  $P[A\ \text{and}\ B\ \text{and}\ C]$ ;  $P[[A\ B]\ [C\ D]]$  corresponds to  $P[(A\ \text{and}\ B)\ \text{or}\ (C\ \text{and}\ D)]$ .\* A microworld for integers and algebra uses prefix notation instead of infix, but represents functions and composition as one would in tenth-grade algebra [e.g., as either  $f(g(x))$  or  $(f \circ g)(x)$ ]. In short, one is somewhat constrained in choosing a command format by the conventions already established in mathematics.

**Graphics Display** Decisions about the design of the graphics display are important, but are not among the most crucial decisions one makes in designing a mathematical microworld. The only strong constraint is that there is a clear

\* This microworld focuses upon representing the sample spaces of experiments as trees.  $[A\ B\ C]$  represents the path in the experiment tree containing those outcomes. Thus, it would be nonsensical to represent  $P[[A\ B]\ \text{and}\ [C\ D]]$  if  $[A\ B]$  and  $[C\ D]$  are of different paths. *A priori*, the probability is 0.

correspondence between the *change* in the display effected by a command and the mathematical meaning of the command.

**The Problem Sets** The creation of problems for students is not part of designing a mathematical microworld per se, but the problems exert a strong influence upon a microworld's design. This point was illustrated in the section on the evolution of MOTIONS. In order to pose problems and activities that focused on the independence of position and direction, the microworld needed to include DOT and LINE commands. Also, to pose problems that focused upon composition of transformations on several levels of conceptual complexity, the microworld needed to allow several ways to represent composition. That is, the designer of a mathematical microworld must have an idea of the kinds of problems one wishes to pose so that the microworld allows the problems to be posed meaningfully.

A consensus is building that the problems one commonly asks students to solve has a dramatic influence on students' cognitive structures in scientific domains (Heller & Hungate, 1985; Larkin, 1981a, 1981b; Reif & Heller, 1982; Thompson, 1985a). As such, careful consideration must be given to the content and organization of the problems one asks students to solve in the context of a mathematical microworld. One crucial feature of the problem sets is that they address the subject matter at several levels of abstractness, and the microworld needs to allow a student to use it at any of those levels at any time.

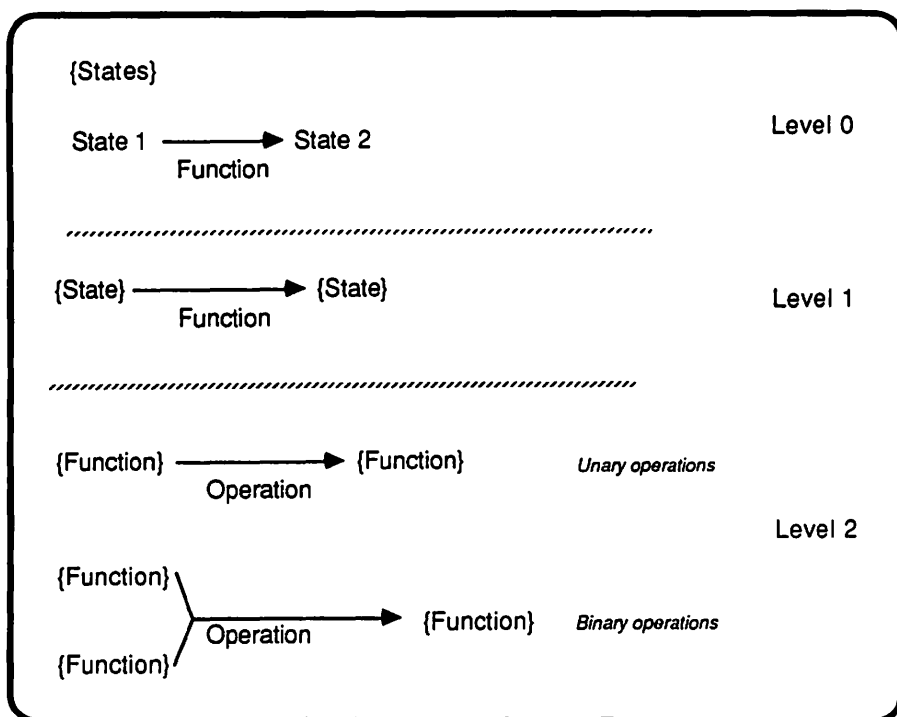
"Levels of abstractness" in the problems means that initial objects are related by functions, and classes of functions are related by operations. This is reflected in Fig. 5.9. Entry-level problems focus on states, as depicted in the graphics display, with functions serving to connect states (Level 0). Problems for Level 0 can be of two kinds. The first kind is problems that focus students upon states and their defining characteristics (e.g., the state of a pennant in MOTIONS); the second kind provide any two of

[*State 1, State 2, function*]

and asks the students to provide the third. Problems at Level 1 focus on functions, with states serving as a background for describing a function's general effect. Level 2 problems focus on unary and binary operations upon functions, such as negation (inverse under some binary operation), composition, and (generalized) arithmetic operations. The process of dividing problems into levels is discussed in Dreyfus and Thompson (1985), and Thompson (1985a).

## **Microworlds Are Sometimes Not Enough**

Mathematical microworlds can effect significant improvements in students' mathematical understandings and abilities (Thompson, 1985c; Thompson & Dreyfus, in press). Most students benefit from the use of mathematical micro-



**Figure 5.9** A framework for organizing problems. Level 0 problems focus on states and functions as relationships among states. Level 1 problems focus on the general effect of classes of functions on states. Level 2 problems focus on unary and binary operations on classes of functions.

worlds by exploring them somewhat scientifically, such as by controlling variables while manipulating others, and by using visual feedback to “debug” faulty understandings of the subject matter whenever they lead to unexpected results. However, a significant minority of students using microworlds are effectively blocked from taking advantage of their semantically rich environments. These are students who fail to understand the initial concepts depicted by a micro-world, who do not know how to explore ideas, and/or who do not know how to generalize from examples. For these students, microworlds are not enough. They need assistance in coming to terms with the *raison d’être* of microworlds. The following is a discussion of a current research project at Illinois State

University that emphasizes concepts of exploration and generalization in micro-world environments for students who have difficulty with them.\*

We are immediately in a conundrum when redesigning mathematical microworlds to address misconceptions. The nonjudgmental, passive nature of microworlds is a strong positive feature that serves the goal of long-term mathematical development. But to assist students who have misconceptions, those misconceptions must be diagnosed and remediated. The ability to diagnose and remediate means that a microworld must pose problems, judge responses, and give appropriate feedback—features that are antithetical to the design of mathematical microworlds. The resolution to this conundrum was this: rather than have a tutor decide when to intervene and how to structure the lesson, we allow *students* to decide when to invoke the tutor. This allows us, and students, to maintain a strong separation between a microworld and its tutor.

A second design issue was how students would interact with a microworld so as to allow it to make inferences of *qualitative* misconceptions, as distinct from misconceptions about formal rules and their application (such as those investigated by Matz, 1982, and by Sleeman, 1982). Our solution is to allow students to have qualitative interactions with a microworld, such as by pointing to screen positions or sketching predictions on the screen. These latter features demand input facilities equivalent to a mouse.†

A third design issue was how to represent qualitative knowledge of mathematical content to allow automatized inferences about a student's concepts. Current tests suggest that semantic networks, augmented by bug catalogs, provide at least a viable solution to this problem.‡

### What Does an Intelligent Microworld Look Like?

The major extensions to MOTIONS are designed to address misconceptions of basic concepts and misconceptions of exploration and generalization. To address misconceptions of basic concepts, we incorporate two new commands: a WHY? command, which will cause the computer to explain why it did what it did in response to the previous command, and a PREDICT command, which allows a student to predict the outcome of a command and have the computer comment on the prediction. WHY? and PREDICT are discussed in the next section.

\* As of this writing, only pieces of the programs described herein are actually running. However, since the major issues discussed concern design, I will speak as if the programs are complete.

† Intelligent microworlds are being implemented for the Apple Macintosh™.

‡ I should point out that the link between mathematical microworlds and ICAI has not arisen because of a penchant for finding applications of artificial intelligence in education. Rather, the nature of the pedagogical problem and the design issues noted above have led to a natural marriage.

We plan to address misconceptions of exploration and generalization by incorporating another three commands: EXPLORE, GENERALIZE, and CHALLENGE-ME. The commands EXPLORE and GENERALIZE will work together to emphasize that exploring, hypothesizing, generalizing, and then exploring again is a natural and powerful method for understanding mathematics. CHALLENGE-ME will provide a context for a microworld to interact with students about problem-solving strategies. EXPLORE, GENERALIZE, and CHALLENGE-ME are discussed in a succeeding section.

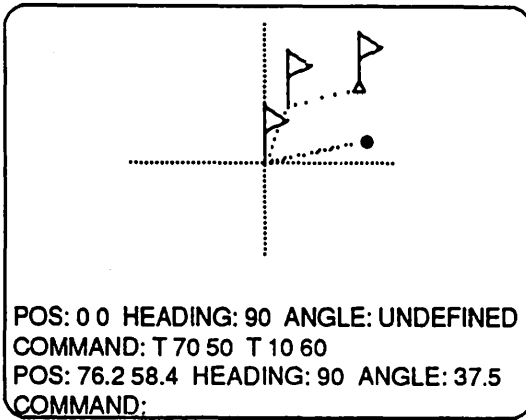
### WHY? and PREDICT

The WHY? command provides students with explanations. Whenever a student enters a command in the context of attempting to solve a problem and the command leads to an unanticipated result, often the source of the error is a misconception of some basic concept. If the student cannot understand why the microworld acted differently than expected he or she can enter WHY? and have the microworld explain its actions.

The level of an explanation is a function of the microworld's current model of the student in regard to the concepts involved in the explanation. A student model with high values on the component concepts receives explanations in terms of theorems (supposedly) already known to that student. A student model with neutral values on the component concepts leads to an explanation in terms of basic definitions and relationships. A student model with low, negative values on the component concepts leads to detailed explanations in terms of basic definitions and relationships accompanied by graphical illustrations.

The more detailed explanations are available to students with high and neutral values in their respective student models by their entering WHY? again, until the microworld has given the most basic explanation of which it is capable. Afterward, the student is referred to the instructor.

The inclusion of a PREDICT command serves two purposes. First, estimating the effect of an operation helps students understand mathematical definitions, relationships, and principles. Second, by having a student register his or her prediction of a command, the microworld then has information from which to infer misconceptions. For example, a very common misconception in transformation geometry is that the direction in which a pennant will move is always determined from the origin (intersection of  $x$ - and  $y$ -axes) when the plane is moved by a translation. In fact, the direction a pennant is moved is determined from its position. Students with this misconception cannot understand the idea of composition of translations, for the pennant never ends up where they expect it (see Fig. 5.10). This misconception is easily uncovered once a student predicts the end state of a pennant prior to performing two translations.



**Figure 5.10** The difference between the actual result of entering T 70 50 T 10 60 and a student's estimate of the result (marked as ●). This student thought that all directions are measured with lines passing through the origin, not realizing that directions can be measured relative to any position in the plane.

### EXPLORE, GENERALIZE, and CHALLENGE-ME

We address misconceptions of exploring and generalizing in two ways. The first is to incorporate an EXPLORE command. When a student enters EXPLORE, the microworld begins to record the student's commands in the anticipation that at some time he or she will enter GENERALIZE. In response to GENERALIZE, the microworld computes a number of generalizations from the student's commands, some, none, or all of which may be valid; nevertheless, they fit the data. The student then chooses one for further exploration and continues to enter commands. The microworld estimates the relevancy of the commands to the chosen generalization, and communicates its estimations when the student enters the command COMMENTS?. The student can also enter YOUR OPINION?, which causes the microworld to give what it considers to be a viable set of commands to test the chosen generalization. The student can enter WHY? to have the microworld explain its choice of commands.

A second way that we address misconceptions of exploring and generalizing is to incorporate a CHALLENGE-ME option. The student can select the type of problems on which to be challenged by referring to a section in the booklet that accompanies the microworld. For example, in the MOTIONS microworld, CHALLENGE-ME 3.4 means that the student wishes to be challenged by a problem dealing with the composition of rotations and reflections (chapter 3,

section 4). The microworld's knowledge of problems is stored as schemata that describe the types of displays, transformations, problems that can be posed, and legitimate plans for solving those problems. It judges students' problem-solving strategies by attempting to fit the commands they enter into one of its lists of legitimate plans.

The GENERALIZE and CHALLENGE-ME commands provide occasions for a microworld to comment on specific problem-solving strategies. These are the only two commands that provide the microworld with definite information about the problem a student is solving. In the case of CHALLENGE-ME, we can constrain the types of problems to those we think are particularly illustrative of important concepts and methods. With prior knowledge of the types of problems upon which a student will work, we can anticipate specific misconceptions, errors, and strategies.

## **Intelligent Microworlds in Schools**

Mathematics education in the nation's schools is ripe for intelligent microworlds. Many states report a critical shortage in qualified mathematics teachers while at the same time they are raising minimum mathematics requirements for graduation. A short-term measure adopted by many states is the practice of allowing teachers to "cross over" from nonscientific disciplines to become mathematics teachers—without requiring further training in mathematics. Shortages and crossovers are occurring at the same time that government and professional reports point out the necessity for an increased emphasis on problem solving in the K–12 mathematics curriculum.

Clearly, schools need assistance in their mathematics classrooms. Microworlds can provide one form of assistance, in that teachers need not be burdened with the task of creating models through which the students can explore and understand mathematical concepts. Moreover, pedagogically sound and mathematically valid intelligent microworlds can put a cognitive scientist, a mathematician, and a diagnostician within calling distance of any student using them. More than a few teachers could benefit from such assistance.

Even with today's critical problems, prospects for the introduction of intelligent microworlds, or ICAI in general, into the nation's mathematics classrooms are not good. By and large, schools are still purchasing 64K 8-bit computers, and there is no reason to believe that they will significantly change their purchasing habits at any time soon (Mazer Corp., 1983; Strategic, Inc., 1984).

An even greater obstacle than the lack of suitable ICAI hardware is the problem of overcoming teachers' conceptions of how one should learn mathematics. Too often, their conception is that one learns mathematics by memorizing fixed, inflexible algorithms for answering stereotypical problems. I have comments written by classroom teachers who have participated in micro-



world-based demonstration lessons wherein they state that though they found microworlds to be interesting, what they witnessed was irrelevant to the topics as they teach them. Microworlds "do not teach rules well enough."

Another obstacle to large-scale implementation of ICAI is the small number of computers available to any one teacher. Presentations on how to use one computer with 30 students are generally among the best attended at mathematics education conferences.

The present structure of the mathematics curriculum and the various bureaucratic systems for changing it present another obstacle to realizing mathematics ICAI in the schools. Large-scale implementation of ICAI will require restructuring the curriculum, if only to allow time for students to interact with computers. Such changes are likely to be within the purview of textbook adoption committees, and innovations in general have not fared well at that level. We would do well to recall the pitfalls encountered by large-scale curriculum projects of the 1960s and 1970s (Conference Board of the Mathematical Sciences, 1976; Vogeli, 1967).

Finally, the greatest obstacle to realizing ICAI in the schools is the need for teachers to rethink their management of time and resources and the need for them to rethink their role in students' learning processes. Subject matter, classroom management, and student-teacher interactions all change when a microworld is integrated with instruction. The subject matter must be rethought so as to be presented within the model embodied in the microworld. Also, when using a computer as a teaching tool teachers are no longer in complete control of the situation; they are no longer the center of attention in the classroom. Their role becomes more that of a choreographer than a manager. They ensure that all the people and props are in the right place at the right time, provide the motivation for the "piece," and then hope for the best.

The potential benefits of intelligent mathematical microworlds, and ICAI in general, are great enough to warrant our best efforts to realize their application to classroom teaching and learning. But we should not think that their value will be apparent to potential consumers. Nor should we think that their value will be easily demonstrated.

## References

- Collis, K. F., & Biggs, J. B. (1982). *Evaluating the quality of learning: The SOLO taxonomy (Structure of Learned Outcomes)*. New York: Academic Press.
- Conference Board of the Mathematical Sciences, National Advisory Committee on Mathematical Education. (1976). *Overview and analysis of school mathematics: Grades K-12*. Washington, DC: CBMS.
- diSessa, A. (1982). Unlearning Aristotelian physics: A study of knowledge-based learning. *Cognitive Science*, 6, 37-76.

- Dreyfus, T., & Thompson, P. W. (1985). Microworlds and Van Hiele levels. *Proceedings of the Tenth Annual Conference of the International Group for the Psychology of Mathematics Education*. Utrecht, The Netherlands: IGPME.
- Fuerzeig, W. (1986). Algebra slaves and agents. Manuscript to appear in *Artificial Intelligence*, Special Issue, Winter 1986.
- Fuller, N., Prusinkiewicz, P., & Rambally, G. (1985). L.E.G.O.—An interactive computer graphics system for teaching geometry. *Proceedings of the World Conference on Computers in Education*, Norfolk, VA.
- Greenleaf, N. (1984). EUCLID: A graphics language for plane geometry. In V. Hansen (Ed.), *Computers in mathematics education*. Yearbook of the National Council of Teachers of Mathematics. Reston, VA: NCTM.
- Heller, J. I., & Hungate, H. N. (1985). Implications for mathematics instruction of research on scientific problem solving. In E. A. Silver (Ed.), *Teaching and learning mathematical problem solving: Multiple research perspectives*. Hillsdale, NJ: Lawrence Erlbaum.
- Kearsley, G. (1985). Personal communication.
- Kleene, S. (1952). *Introduction to metamathematics*. New York: D. Van Nostrand.
- Kneebone, G. (1963). *Mathematical logic and the foundations of mathematics*. New York: D. Van Nostrand.
- Larkin, J. H. (1981a). Cognition of learning physics. *American Journal of Physics*, 49, 534–541.
- Larkin, J. H. (1981b). Enriching Formal Knowledge: A model for learning to solve textbook physics problems. In J. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum.
- Lawler, W. (1982). The progressive construction of mind. *Cognitive Science*, 5, 1–30.
- Lawler, W. (1984). Designing computer-based microworlds. In M. Yazdani (Ed.), *New horizons in educational computing*. London: Ellis Horwood.
- Matz, M. (1982). Toward a process model for high school algebra errors. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press.
- Mazer Corporation (1984). *Trends '83: The school microcomputer market*.
- Papert, S. (1980). *Mindstorms*. New York: Basic Books.
- Piaget, J. (1951). *The psychology of intelligence*. London: Routledge and Kegan Paul.
- Reif, F., & Heller, J. I. (1982). Knowledge structure and problem solving in physics. *Educational Psychologist*, 17, 102–127.
- Rumelhart, D., & Ortony, F. (1977). The representation of knowledge in memory. In J. Anderson, S. Spiro, & W. Montague (Eds.), *Schooling and the acquisition of knowledge*. Hillsdale, NJ: Lawrence Erlbaum.
- Skemp, R. R. (1979). *Intelligence, learning, and action*. London: Wiley.
- Sleeman, D. (1982). Assessing competence in basic algebra. In D. H. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems*. New York: Academic Press.

- Strategic, Inc. (1984). *Educational software: The next boost to the microcomputer market*. San Jose, CA.
- Thompson, P. W. (1985a). Experience, problem solving and learning mathematics: Considerations in developing mathematics curricula. In E. A. Silver (Ed.), *Learning and teaching mathematical problem solving: Multiple research perspectives*. Hillsdale, NJ: Lawrence Erlbaum.
- Thompson, P. W. (1985b). Computers in research on mathematical problem solving. In E. A. Silver (Ed.), *Learning and teaching mathematical problem solving: Multiple research perspectives*. Hillsdale, NJ: Lawrence Erlbaum.
- Thompson, P. W. (1985c). A Piagetian approach to transformation geometry via micro-worlds. *Mathematics Teacher*, 78, 465–471.
- Thompson, P. W., & Dreyfus, T. (In press). Integers and algebra: Parallels in operations of thought. *Journal for Research in Mathematics Education*.
- Vogeli, B. (1967). The rise and fall of the "new math." *Inaugural lecture as Clifford Breuster Upton Professor of Mathematical Education*. New York: Columbia University, Teachers' College.