A RAND NOTE

Multiple Representations of Mathematical Reasoning

David McArthur, Christopher Burdorf,
Tor Ormseth, Abby Robyn, Cathleen Stasz

May 1988

40 Years 1948-1988
RAND

The research described in this report was supported by the National Science Foundation under grant No. MDR-8751515, and by The RAND Corporation using its own research funds.

Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

A RAND NOTE

N-2758-NSF/RC

Multiple Representations of Mathematical Reasoning

David McArthur, Christopher Burdorf,
Tor Ormseth, Abby Robyn, Cathleen Stasz

May 1988

Prepared for
The National Science Foundation

RAND

PREFACE

The research reported in this Note was supported jointly by the National Science Foundation, as a part of its Applications of Advanced Technologies Program, and The RAND Corporation, from its own research funds.

This study will also be published in the proceedings of the International Conference on Intelligent Tutoring Systems, held in Montreal in June 1988.

SUMMARY

Multiple alternative representations of mathematical reasoning have interesting applications in the context of an intelligent tutor for basic algebra. This Note describes the notion of multiple representations at a general level. It then describes RAND's intelligent algebra tutor and specific tools it contains that support multiple representations: a boxes-and-weights tool and goal commands. These tools provide students with important learning opportunities.

The Note concludes with comments about continuing research in this area. It is important to continue implementing different representational systems and testing them in real educational contexts. In addition, theoretical work must be directed toward deriving a general classification scheme for representations.

CONTENTS

FIGURES

I. INTRODUCTION

This Note discusses the role of multiple alternative representations of mathematical reasoning within the context of an intelligent tutor for basic algebra. Section II discusses the notion of multiple representations at a general level. In Section III we describe our intelligent algebra tutor and discuss several specific tools that exploit multiple representations to provide students with important learning opportunities. Section IV presents our conclusions and comments about continuing research in this area.

II. MULTIPLE REPRESENTATIONS OF ALGEBRA AND ALGEBRAIC REASONING

DISCUSSION

The traditional notation of mathematics[1] was not deliberately designed by an individual or committee. It has evolved slowly, largely in response to the preferences and needs of the mathematicians and professionals who use it. Perhaps as a consequence of this evolutionary development, the notation has several unusual properties which are often difficult to perceive because they are so deeply ingrained in our culture. The most distinctive feature of the notation is the complexity of its structure and of the deductive machinery used to manipulate it. This complexity is not an accident. Bertrand Russell has astutely pointed out that the notation is deliberately arcane, to ensure that the results it yields cannot be influenced by our intuition. In addition, the traditional representation is appropriately terse for the expert; however, this terseness is achieved at the expense of syntactic ambiguity.

Neither teachers nor students have played a role in shaping the traditional language of algebra. As a consequence, the notation alone causes many students to find mathematics a difficult subject. Unfortunately, teaching students basic algebra is almost always synonymous with teaching them to effectively manipulate expressions in the traditional notation. Indeed, few people realize that mathematics and the traditional notation are not identical. The traditional notation is only one possible representation, just as FORTRAN is only one of many representation systems for algorithms.

Since the traditional representation is not optimal for learning, it makes sense to investigate alternative representations that may be more appropriate. In computer science, students do not begin with specialized languages, such as APL, or highly general languages, such as ADA. Instead, they start with simple languages, such as PASCAL, or languages deliberately designed for learning, e.g., LOGO. Once students learn the basic concepts, they progress to more powerful languages or languages that are more effective for specific tasks.

The idea of representing algebra in a different notation is not new. Over the past 50 years, several such languages have been proposed, but none has managed to attain any real

---

[1]In this Note, we use the phrases "traditional notation" and "traditional representation" to refer to the way mathematics is expressed in mathematics textbooks, and, more generally, to describe the idea that doing mathematics is equivalent to manipulating expressions that are in this syntax.

popularity. (See Lesh (1987) for a discussion of some of the reasons for these failures.) We propose several new twists to the idea of changing the mathematical notation. First, we are not suggesting replacing the traditional language; rather, we propose to augment it with other notations. More than casting doubt on the standard representation, we are calling into question the whole idea of learning algebra through a single notational vehicle. Just as in computer science, we should seek multiple representations, each with its own particular strengths and weaknesses. In teaching, we should use a particular representation for the tasks suited to it and go on to others as the pedagogical need arises. Adopting a view of mathematics teaching that is not wedded to a particular representation focuses learning on abstract problem-solving skills, rather than on notion-dependent features. Multiple representations have frequently been suggested as a way of helping students abstract the "meaning" from mathematical manipulations (Lesh, 1987; Linn, 1986).

Our approach also differs from previous approaches in that we are interested in exploring novel representations that facilitate the process of mathematical reasoning, not just the comprehension of mathematical expressions. The most common objection to mathematical notation is that it is syntactically ambiguous. While this objection is valid, we do not regard it as the most important pedagogical difficulty with the traditional language. An equally significant limitation is that visible solutions to problems contain little information about the reasoning process that generated them. The visible steps in a solution correlate little with the reasoning steps underlying them. The important consequence of this disparity is that *teaching the notation is of little help in teaching how to solve problems in the notation.* We are interested in exploring representations that facilitate dynamic problem solving, as well as static comprehension.

Finally, we have introduced the innovation of using computers as a basis for designing new representations of algebra. One reason for maintaining the traditional notation for algebra has been the constraints of the pencil-and-paper medium. Less terse or nonlinear representations have been impractically awkward to use, however natural they might be for learning. With the advent of computers, these constraints are no longer valid. Using computers as a medium for doing mathematics makes a wide variety of representational possibilities feasible. For example, highly graphical representations can be considered without concern for efficiency. On computers, it can be easier to draw pictures than to write characters.

FEATURES OF A DESIRABLE NOTATION FOR ALGEBRA PROBLEM SOLVING

To help constrain the search for alternative representations, we have sought representations that complement the standard language of mathematics. We are interested in finding notations that are strong where the traditional notation is weak, from the student's point of view. The alternative notations need not be able to accomplish easily the things that the traditional representation already facilitates.

A representation for mathematics that complements the traditional notation might exemplify several of the properties listed below. These properties refer not only to the representation *per se*, but also to the reasoning processes used to manipulate expressions in order to solve a problem. We emphasize that we have not attempted a complete analysis of the strengths and weaknesses of the traditional notation for mathematics; such an analysis must await the development of a more thorough theory of symbols for mathematics (Kaput, 1987). The following discussion is only a first attempt in this direction.

An effective notation for algebra problem solving should be:

- *Unambiguous and syntactically regular.* The traditional notation is ambiguous, in that precedence is not always noted explicitly. As a result, the student must have special knowledge of the operations to disambiguate expressions. For example, the student must know that multiplication has a higher precedence than addition to interpret $2x + 6$ correctly. Similarly, the traditional representation is irregular, in that operations are not treated systematically. In most cases, operators appear in infix position; however, in other cases (e.g., exponentiation), spatial position denotes the operation. An example of a regular and unambiguous notation is the prefix representation of Lisp, in which, for example, (+ (* 3 (^ x 2)) (* 6 x) 4) denotes $3x^2 + 6x + 4$.
- *Concrete.* The traditional notation for algebra is abstract, in that the semantic interpretation of expressions is impossible to discern without knowledge of the special conventions for mapping syntax to semantics. It is possible to devise representations in which expressions, especially the notion of variables, map naturally onto students' existing cognitive structures; this can give the expressions an immediate and reasonable semantic interpretation.

The above two properties apply to the notation *per se*. The following properties pertain to reasoning in the notation:

- *History preserving*. As noted above, solutions in the traditional notation contain little information about the reasoning process that generated them. We would like to design alternative representations that reflect and preserve (reify) the reasoning process that generated them.
- *Connected to physical intuition*. This attribute, like concreteness, refers to the ability to root the representation in prior knowledge or models of systems. However, unlike concreteness, the connection to physical intuition refers to the naturalness of inferencing that the representation affords. It should be possible to map the syntax of the representation onto familiar objects that impart a semantic interpretation. It should also be possible to map inferences in the formalism onto intuitively comprehensible processes.

The representations we have investigated do not necessarily exemplify all of these properties. Since we are considering multiple representations, each one need only possess a few desirable properties. However, the collective notations should include at least one that excels for each of the representational goals. In the following section, we briefly describe some novel representations for algebraic equations and their solution. These notations have been developed as part of an intelligent tutoring system for basic algebra.

III. OVERVIEW OF AN INTELLIGENT TUTOR FOR BASIC ALGEBRA

We have built and tested our intelligent algebra tutor in a series of versions. The first version runs on the Symbolics 3600 Lisp machine and on Sun Microsystems workstations (McArthur, 1986, 1987). It has been tested at The RAND Corporation, with local high school students, and it currently runs on six Sun 3/50 workstations at Santa Monica (California) High School, where it will be more extensively tested and developed over the next several years.

The student sees the tutor as a collection of windows and menus, as shown in Fig. 1. The menus on the left allow the tutor and student to converse about reasoning and problem solving. To the right of the menus, at the bottom of the screen, is the WorkSpace, where the student creates each new line in his or her solution. New lines or reasoning steps can be created either by typing expressions or by writing them on an electronic tablet which recognizes the strokes as characters and passes them to the tutor, where they are interpreted. To the right of the Workspace is the CommentSpace, where the tutor provides textual feedback to the student.

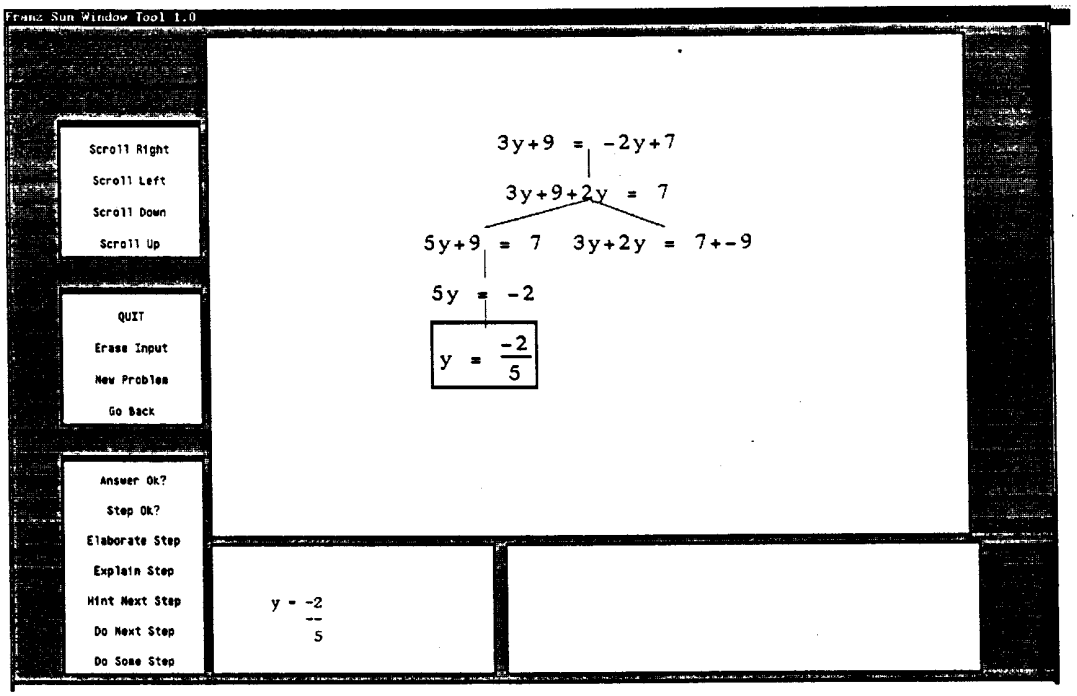

Fig. 1—The basic interface for the algebra tutor

The large window is the DisplaySpace, where the student's reasoning is recorded and queried. Problem solving is represented here as a *reasoning tree*. Many of the menu items are used to manipulate the "nodes" in this tree. For example, the student can test one of his own steps by selecting *Step OK?* and then pointing between two nodes. The tutor will respond, telling the student that the step is reasonable, that it is logically invalid, or that it is valid but may be strategically unreasonable.

In a simple way, the reasoning tree is itself a subtly different representation of algebra problem solutions. Students typically write solutions as linear sequences of steps. When they think they have made a mistake, they usually have to abandon the current sequence and start a new one. They thus lose the history of their reasoning. The reasoning tree, however, preserves the reasoning history. Each branch in the tree represents an alternative solution, or line of attack. Hence a tree representation allows easy comparison of different solutions, both the student's and the tutor's. Moreover, menu items such as *Go Back* permit the history to be exploited effectively. The student selects this item when he or she wants to return to a previous solution path. The tutor then permits the new current expression (i.e., the one that it boxed) to be changed by pointing to any other expression in the tree. That expression then becomes the current one.

Over the past year, we have extended the basic tutoring environment shown in Fig. 1, using it as the foundation for a series of divergent intelligent tutoring tools. The following sections briefly describe two such tools, each of which demonstrates the value of multiple representations of mathematical reasoning.

EXAMPLE 1: THE BOXES-AND-WEIGHTS NOTATION FOR LINEAR EQUATIONS

The Boxes-and-Weights Tool

Figure 2 shows an example of the boxes-and-weights tool, in which students are given a problem to solve, i.e., $4(x + 1) + 2(x + 1) = -6$, but instead of solving it by typing in successive algebraic transformations, they manipulate an analogous boxes-and-weights representation. The boxes-and-weights representation is shown in the upper left window, and corresponding equational representations are shown in the upper right window.

The primitive objects in the concrete model are weights, boxes, and a balance scale. A positive unit weight is a mass of single-unit size (according to some arbitrary linear scale). All unit weights are identical. Each positive unit weight is designated by a plus sign. A positive unit weight denotes the numeral 1 or +1. A negative unit weight is a mass of single-unit size whose weight is the inverse of a positive unit weight on our arbitrary linear scale.

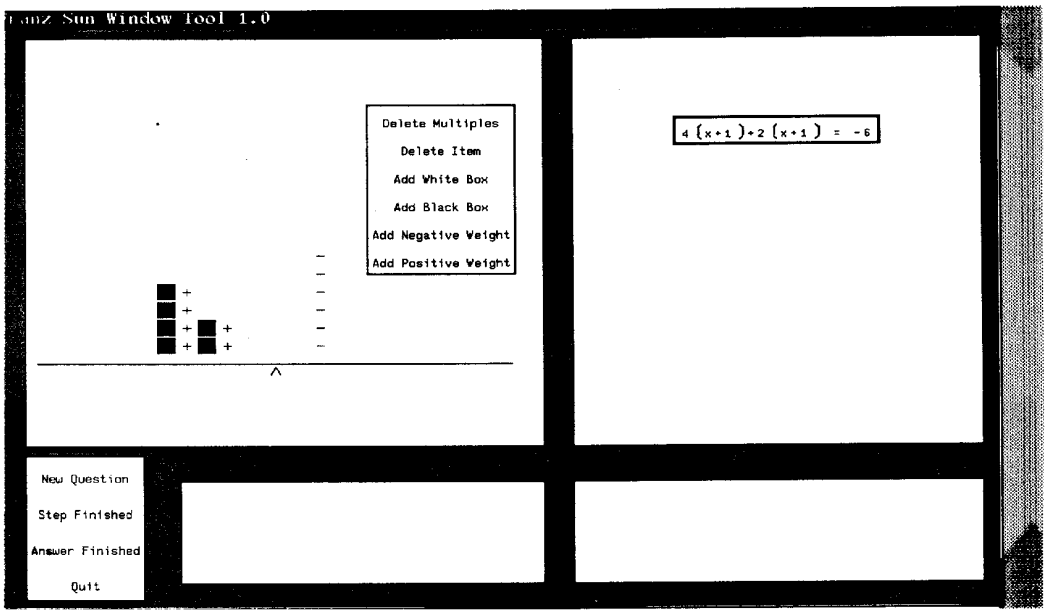

Fig. 2—Configuration of the tutor using the boxes-and-weights representation

Intuitively, a negative weight is "lighter than air" and combined with a positive unit weight would yield a compound with exactly 0 weight. Each negative unit weight is designated by a minus sign. A negative unit weight denotes the numeral -1.

A black box is a weightless container that holds a fixed number of weights, K. For any given problem, a black box contains either all positive weights or all negative weights. We refer to this as the valence of the black box. A white box is a weightless container that must have the same number of unit weights as a black box. For any given problem, a white box must have a valence opposite to that of a black box. If a black box contains only positive unit weights, a white box must contain negative unit weights. A black box denotes a variable (e.g., x).

When a student is given a problem like that in Fig. 2, the initial configuration of boxes and weights is "balanced," indicating a true algebraic statement. The goal of the student is to deductively determine the number of unit weights in a black box and the valence of the black box. Students must arrive at their answer by transforming the given problem configuration into a final configuration in which one black box is on one side of the balance, and only unit weights are on the other side.

Configurations are transformed by using a mouse and menu that enable the student to add or delete boxes or weights from either side. When the student thinks he has made a balance-preserving transformation, he clicks the menu item *Step Finished.* The tutor underlying the interface will then (1) translate the configuration of boxes and weights to an equational representation and display the equation on the window on the right; (2) determine whether the manipulation represents a valid deduction; (3) provide textual feedback about the validity of the step (in the lower right window); and (4) graphically unbalance the configuration if the step is invalid. Figures 3, 4, and 5, show several of these activities in the course of solving the problem displayed in Fig. 2.

The student is able to play several different roles in this learning environment. In the version shown in the figures, the student manipulates the concrete boxes-and-weights representation and observes its effect on the corresponding equations. In another version, the student effects the equation transformations, and boxes-and-weights manipulations are done automatically. In a third version, the student merely watches as the tutor solves the problem in both representations.

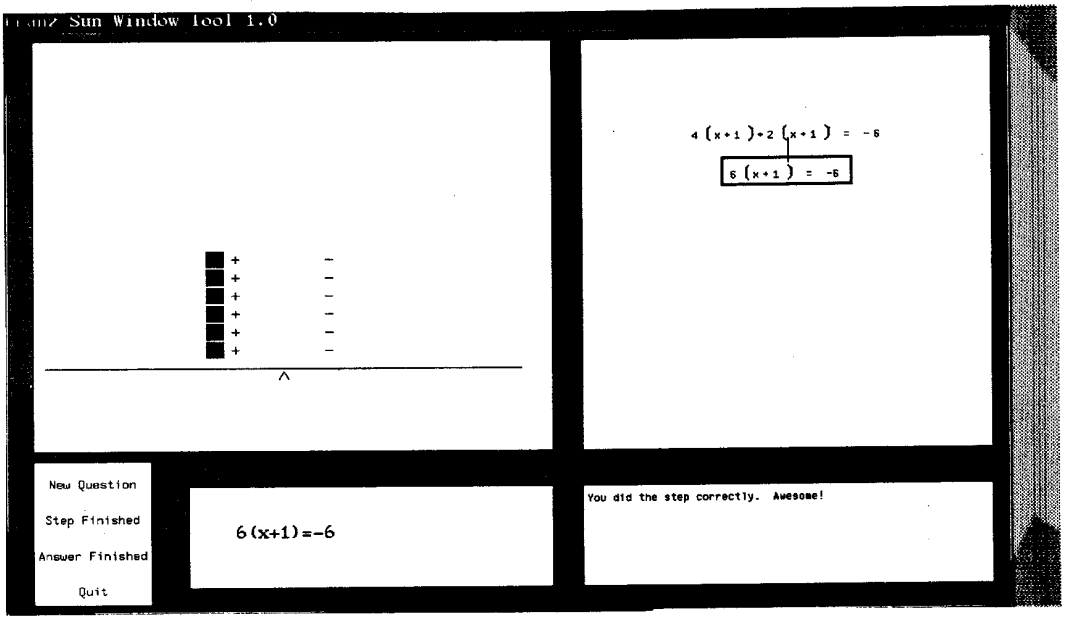

Fig. 3—Response to a correct student step

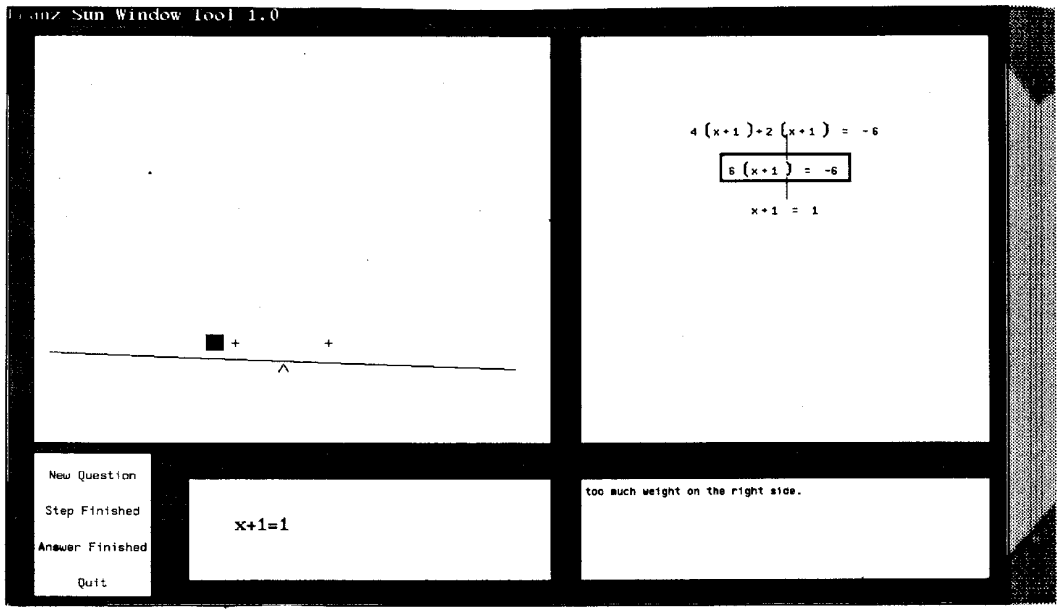

Fig. 4—Response to an incorrect student step

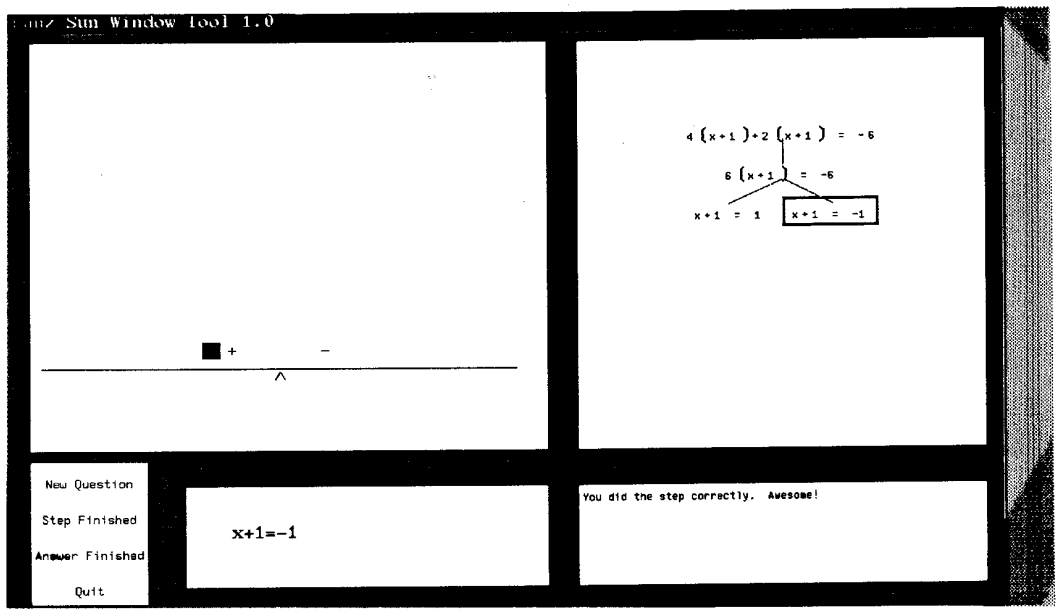

Fig. 5—Branching of alternative solution lines

Implications of the Boxes-and-Weights Tool for Learning through Multiple Representations

The boxes-and-weights notation is somewhat limited in its expressive capacity. It is applicable to only a well-defined subset of basic algebra equation solving and cannot be thought of as a possible replacement for the traditional representation for algebra. However, viewed as one of several representational systems in a heterogeneous package, it has several potential pedagogical advantages. Most important, it is a concrete representation. Symbols and combinations of symbols have natural interpretations in the form of physical objects and forces with which the student is already familiar. The student can use his or her existing intuition abouts weights and balances to guide the interpretation of a given configuration. In addition, the same physical intuition can guide the student in the process of problem solution. The student can assess the validity of a step through his or her understanding of how balance changes with the the addition, deletion, or rearrangement of weights. For example, it should be obvious to the student that any manipulation that simply rearranges the objects on a single side of the balance will not cause it to become unbalanced. Corresponding transformations of equations (e.g., the distributive law of multiplication over addition) are not at all obvious to many students. Thus, the representation is *dynamic* in the sense of Lesh (1987) and Dienes (1960). It is not merely another way to state static algebraic equations; it is an alternative way to reason mathematically, however simplified. Students can not only translate structures of one representation into structures in the other, they can *transform* structures in either representation to arrive at solutions, and this may help them understand reasoning in the other representation.

EXAMPLE 2: COMMANDS FOR SOLVING ALGEBRAIC EQUATIONS

The Commands Tool

Figure 6 shows a second tutoring tool that exploits alternative representations of algebra and algebraic reasoning. In this version, students solve typical problems, but instead of manipulating equations or boxes, they issue commands, which the tutor uses as instructions to effect the actual algebraic symbol manipulation. For example, in Fig. 6, the student has issued commands to add $+2x$ to both sides, simplify, subtract 8 from both sides, and divide both sides by 5. Commands are recorded in the CommandSpace at the right. Commands issued by the tutor (e.g., in response to a succession of *Help Next Step* requests) are shown in inverse video. As in other versions of the tutor, the student is free to explore alternative lines of reasoning. When the student moves from one line to another, the tutor

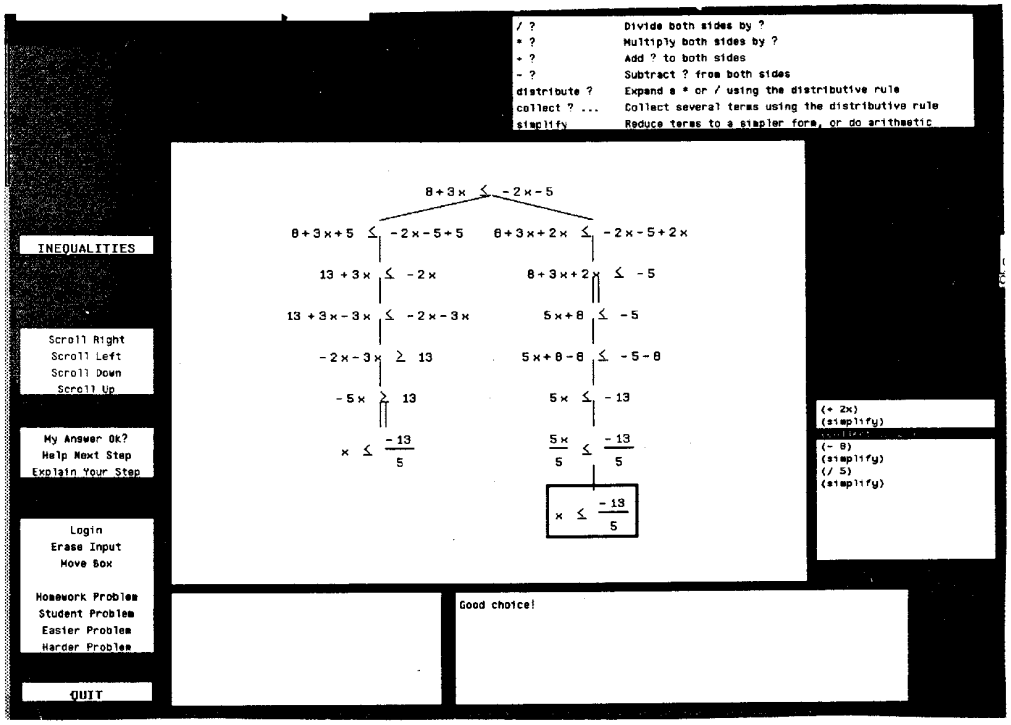

Fig. 6—Commands tool with lower-level commands available to student

simply restores the commands that generated the new line to the CommandSpace window. In Fig. 6, for example, the commands in the CommandSpace are those that generated the current (right) path in the DisplaySpace. If the student used *Move Box* to change the current focus back to the left path, the CommandSpace would be updated to show the commands that generated that line.

We have implemented several variants of the commands tool, each offering students different levels of commands or imposing different tasks on them. Figure 7, for example, shows a variant with a different command set. Here, students choose from a much higher-level set of options. The commands in this version do not describe the desired symbolic transformation of the current expression, but rather mention the *strategic goal* to be achieved by the next step. The tutor takes the responsibility of interpreting this goal in terms of a lower-level symbolic transformation. Permitting students to focus on very high-level decisionmaking while ignoring the details of symbol manipulation opens up some exciting learning possibilities. For example, we have seen students speed through the whole Algebra I curriculum for linear equations in a matter of hours, rapidly acquiring a general feeling for the kinds of skills they need, while leaving the practice of detailed symbol-manipulation skills until later.

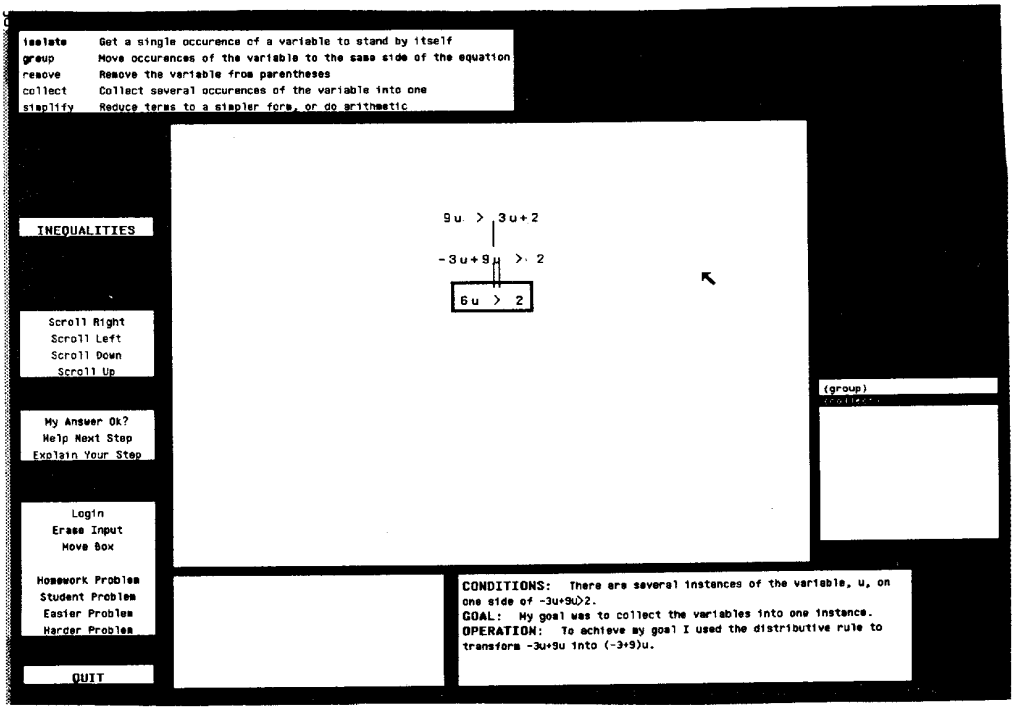

Fig. 7—Commands tool with higher-level commands available to student

Implications of the Commands Tool for Learning through Multiple Representations

The commands described above are a different kind of representation of algebra from the boxes-and-weights tool. Like Dienes blocks in arithmetic, the boxes-and-weights representation is *isomorphic* to the traditional notation. For a subset of algebra, boxes-and-weights structures map one-to-one onto traditional structures.[1] On the other hand, the commands shown in Figs. 6 and 7 do not map onto traditional structures in obvious ways. Essentially, they cannot be regarded as alternative representations for algebraic expressions; rather, they are *representations of algebraic transformations or of reasoning intentions*. This distinction has several implications. First, the commands representation does not stand by itself. While problems in the boxes-and-weights representation can be solved by themselves, without reference to the traditional notation, the commands representation is useful only if it is coupled with the traditional notation (or some isomorph). The commands do not result in transformations within their own notational system, but rather within the system to which they are coupled. Thus, the commands are a *dependent* representation, not an *autonomous* one.

[1]In reality, the mapping is not quite so simple, but one-to-one is a reasonable approximation for purposes of this Note.

The commands tool as a whole can be thought of as a *hybrid* representational system, composed of two distinct but complementary notations: Mathematical reasoning is done in the dependent notation, while the results of reasoning are represented in the coupled notation. As noted earlier, traditional representations for mathematics do not record the history of reasoning that leads to results. The commands representation is a simple first attempt to provide a vocabulary that enables the process of reasoning, not merely the products of reasoning (Brown, 1984), to be explicitly recorded.

IV. RELATED RESEARCH AND CONCLUSIONS

We believe a two-pronged approach must be taken if progress is to be made in developing effective alternative representations for mathematics learning. First, it is important to continue implementing specific different representational systems and to test them in real educational contexts. Several researchers who are developing intelligent tutoring systems have begun to investigate the role of multiple representations of reasoning (Janvier, 1987). One such system, Algebraland (Foss, 1987), is similar to our commands tool in structure, although its intent is to help students learn metacognitive skills through "productive thrashing." Lesh (1987) describes several systems that provide multiple embodiments of arithmetic and algebraic concepts in computer environments. Graphical representations of algebraic notions, not discussed in this Note, play a central role in Lesh's work. The Word Problems Project at Harvard's Educational Technology Center is also examining the power of different representations for amplifying or transforming students' reasoning capabilities (Kaput, 1986).

In addition to these experiments, theoretical work directed toward deriving a general classification scheme for representations must continue. The number of possible alternative representations is so vast that without general organizing principles, it will be impossible to make principled comparisons between apparently different kinds or representations. Moreover, there will be no systematic way of judging what types of representations should be sought most intensively. The study reported in this Note is merely a small first effort at such an analysis. Kaput (1987) has recently begun a more ambitious attempt to develop a theory of the use of symbols in mathematics.

REFERENCES

Brown, J. S. (1984). Process versus product: A perspective on tools for communal and informal electronic learning. In *Education and the Electronic Age*, proceedings of a conference sponsored by the Educational Broadcasting Company.

Dienes, Z. (1960). *Building up mathematics* (4th ed.). London: Hutchinson Educational Ltd.

Foss, C. (1987). Acquisition of error management skills. *Third International Conference on Artificial Intelligence and Education*, p. 27.

Janvier, C. (ed). (1987) *Problems of Representation in the Teaching and Learning of Mathematics*. Hillsdale, NJ: Lawrence Erlbaum.

Kaput, J. (1986). Towards a theory of symbol use in mathematics. In C. Janvier (ed.), *Problems of Representation in the Teaching and Learning of Mathematics*. Hillsdale, NJ: Lawrence Erlbaum.

Kaput, J., C. Luke, J. Poholsky, and A. Sayer (1986). *The Role of Representations in Reasoning with Intensive Quantities: Preliminary Analyses*. Educational Technology Center Technical Report, September.

Lesh, R., M. Behr, and T. Post (1987). Representations and translations among representations in mathematics learning and problem solving. in C. Janvier (ed.), *Problems of Representation in the Teaching and Learning of Mathematics*. Hillsdale, NJ: Lawrence Erlbaum.

Linn, M. (1986). *Annual Report: Computers and Problem Solving.*

McArthur, D. (1986). Developing computer tools to support performing and learning complex cognitive skills. In K. Pedzek, D. Berger, and B. Bankes (eds.), *Applications of Cognitive Psychology: Computing and Education*. Hillsdale, NJ: Lawrence Erlbaum Associates.

McArthur, D., C. Stasz, and J. Hotta (1987). Learning problem-solving skills in algebra. *Journal of Educational Technology Systems*, 15(3), pp. 304-324.